LaTeX'e Kısa Bir Giriş

Onur Şahin

Giresun Üniversitesi

İçindekiler I

1. Giriş
   - Dersin Amacı ve LaTeX'in Avantajları
   - LaTeX'in kısa tarihi
   - LaTeX Kaynak Dosyaları
2. LaTeX Kaynak Dosyasının Hazırlanması
   - Kaynak dosyasının yapısı
   - Paketler
3. Dokümanın Hazırlanması
   - Genel Ayarlar
   - Düz Metin Ayarları
   - Ortamlar
4. Matematiksel İfadelerin Yazılması
   - Giriş
   - Temel Matematik Komutları
   - Matematik Ortamları

İçindekiler II

5. Şekil, Kaynakça ve Dizin Ekleme
   - Şekil Ekleme
   - Kaynakça
   - Dizin Ekleme
6. Grafik Çizimi
   - Basit Geometrik Şekillerin Çizimi
   - Fonksiyon Grafiklerinin Çizimi
7. LaTeX'in Özelleştirilmesi
   - Yeni Komut Tanımlama
   - Mevcut Komutu Düzenleme
8. Slayt Hazırlamak
9. Kaynaklar

Dersin Amacı

Bu dersimizde, matematik ve benzeri bilimsel içerikli dokümanları hazırlamakta sıkılıkla kullanılan LaTeX yazılımın ne olduğunu ve bu yazılım kullanılarak dokümanların (tez, kitap, makale vb.) ve sunumların nasıl hazırlanabileceği öğrenilecektir. Bilindiği üzere MS Word, Google Docs, Apple Pages ve benzeri WYSIWYG (what you see is what you get) tarzı yazım programlarında dokümanda karakterler yazılırken belgenin son hali görülebilmektedir. LaTeX'te ise belge, herhangi bir derleme programında kaynak kodu olarak hazırlanarak daha sonra LaTeX yazılımı kullanılarak biçimlendirilerek istenilen halinde görüntülenebilir.

Peki neden LaTeX?

Neden LaTeX?

1. Tüm işletim sistemlerinde çok düşük donanım gereksinimiyle çalışır.
2. Matematiksel formülleri içeren dokümanları oluşturmak için en uygun yazılımdır.
3. Ücretsiz ve açık kaynak kodludur.
4. Kaynak dosyası standart ASCII (American Standart Code for Information Interchange) dosyası olduğundan tüm metin editörleri tarafından işlenebilir.
5. Platformadan bağımsızdır. Windows, Linux, MacOs gibi işletim sistemlerinde sorunsuzca çalışır.
6. Springer, Elsevier, Birkhauser gibi dünyanın en büyük yayınevleri artık dergi ve kitaplar için yazarlardan LaTeX dosyası istemektedir.

Neden LaTeX?

1. Tüm işletim sistemlerinde çok düşük donanım gereksinimiyle çalışır.
2. Matematiksel formülleri içeren dokümanları oluşturmak için en uygun yazılımdır.
3. Ücretsiz ve açık kaynak kodludur.
4. Kaynak dosyası standart ASCII (American Standart Code for Information Interchange) dosyası olduğundan tüm metin editörleri tarafından işlenebilir.
5. Platformadan bağımsızdır. Windows, Linux, MacOs gibi işletim sistemlerinde sorunsuzca çalışır.
6. Springer, Elsevier, Birkhauser gibi dünyanın en büyük yayınevleri artık dergi ve kitaplar için yazarlardan LaTeX dosyası istemektedir.

Neden LaTeX?

1. Tüm işletim sistemlerinde çok düşük donanım gereksinimiyle çalışır.
2. Matematiksel formülleri içeren dokümanları oluşturmak için en uygun yazılımdır.
3. Ücretsiz ve açık kaynak kodludur.
4. Kaynak dosyası standart ASCII (American Standart Code for Information Interchange) dosyası olduğundan tüm metin editörleri tarafından işlenebilir.
5. Platformadan bağımsızdır. Windows, Linux, MacOs gibi işletim sistemlerinde sorunsuzca çalışır.
6. Springer, Elsevier, Birkhauser gibi dünyanın en büyük yayınevleri artık dergi ve kitaplar için yazarlardan LaTeX dosyası istemektedir.

Neden LaTeX?

1. Tüm işletim sistemlerinde çok düşük donanım gereksinimiyle çalışır.
2. Matematiksel formülleri içeren dokümanları oluşturmak için en uygun yazılımdır.
3. Ücretsiz ve açık kaynak kodludur.
4. Kaynak dosyası standart ASCII (American Standart Code for Information Interchange) dosyası olduğundan tüm metin editörleri tarafından işlenebilir.
5. Platformadan bağımsızdır. Windows, Linux, MacOs gibi işletim sistemlerinde sorunsuzca çalışır.
6. Springer, Elsevier, Birkhauser gibi dünyanın en büyük yayınevleri artık dergi ve kitaplar için yazarlardan LaTeX dosyası istemektedir.

Neden LaTeX?

1. Tüm işletim sistemlerinde çok düşük donanım gereksinimiyle çalışır.
2. Matematiksel formülleri içeren dokümanları oluşturmak için en uygun yazılımdır.
3. Ücretsiz ve açık kaynak kodludur.
4. Kaynak dosyası standart ASCII (American Standart Code for Information Interchange) dosyası olduğundan tüm metin editörleri tarafından işlenebilir.
5. Platformadan bağımsızdır. Windows, Linux, MacOs gibi işletim sistemlerinde sorunsuzca çalışır.
6. Springer, Elsevier, Birkhauser gibi dünyanın en büyük yayınevleri artık dergi ve kitaplar için yazarlardan LaTeX dosyası istemektedir.

Neden LaTeX?

1. Tüm işletim sistemlerinde çok düşük donanım gereksinimiyle çalışır.
2. Matematiksel formülleri içeren dokümanları oluşturmak için en uygun yazılımdır.
3. Ücretsiz ve açık kaynak kodludur.
4. Kaynak dosyası standart ASCII (American Standart Code for Information Interchange) dosyası olduğundan tüm metin editörleri tarafından işlenebilir.
5. Platformadan bağımsızdır. Windows, Linux, MacOs gibi işletim sistemlerinde sorunsuzca çalışır.
6. Springer, Elsevier, Birkhauser gibi dünyanın en büyük yayınevleri artık dergi ve kitaplar için yazarlardan LaTeX dosyası istemektedir.

LaTeX ile MS Word'ün karşılaştırması

	LaTeX	MS Word
WYSIWYG	Hayır	Evet
Şekiller	Evet	Evet
Matematik	Evet	Evet
Stil Değiştirme	Evet	Evet
Atıf Verme	Evet	Evet
Kaynak Dizini	Evet	Hayır
Platformadan Bağımsızlık	Evet	Hayır
Ücretsiz	Evet	Hayır

TeX ve LaTeX'in Tarihçesi

TeX

TeX, Donald E. Knuth tarafından yazılmış bir bilgisayar programıdır. Amacı metinleri ve matematik formülleri dizmektir. Knuth 70li yıllarda, kitap basımında kullanılmaya başlayan otomasyon teknolojilerinin özellikle kendi kitaplarının ve makalelerinin baskı kalitesini nasıl düşürmekte olduğunu görünce, bu teknolojinin imkanlarını araştırmak üzere 1977 yılında TEX dizgi programını yazmaya başladı.

TeX ilk olarak 1979'da SAIL (Stanford Artificial Intelligence Language) ile yazılmış daha sonra bu sürüm 1982'de Pascal ile yeniden yazılmıştır. Bugün kullandığımız şekliyle TeX 1982 yılında ortaya çıktı, daha sonra 8-bitlik karakter işleyebilen ve diğer dilleri de destekleyen sürümü 1989 yılında yapıldı. TeX, 3.0 sürümü ile yaygınlaşmaya başlamıştır. Bu sürümden sonra TeX sürümleri π (Pi) sayısına her defasında bir ondalık eklenerek ifade edilir, bugün 3.141592653 sayılı sürümdedir.

LaTeX

LaTeX, Leslie Lamport tarafından geliştirilen TeX sistemi üzerine kurulmuş ve TeX'i kullanmayı kolaylaştıran bir yazılımdır. Diğer bir deyişle LaTeX (Lamport TeX), TeX'in kendisidir.

TeX ve LaTeX kelimelerinin sonundaki X harfi İngilizce'deki "x (iks)" harfi değil Yunanca'daki χ (chi) harfidir. Dolayısıyla bu kelimeler "Teks" ve "Lateks" olarak değil "Tek(Tekh)" ve "LeyTek(LeyTekh)" veya "LaTek(LaTekh)" şeklinde telaffuz edilir.

LaTeX'in kurlumu

LaTeX'in birçok ücretsiz sağlayıcısı mevcuttur. Bunlardan en yaygınları MikTeX, TeXLive ve MacTeX dir. LaTeX' ile ilgili pek çok gerekli kurulum dosyası ve paketi CTAN (Comprehensive TeX Archive Network)'ın resmi sitesi olan "`https://ctan.org`" sitesinden elde edebiliriz.

Biz bu dersimizde LaTeX'i bulunduran ve Windows işletim sistemi üzerinde çalışan MikTeX yazılımını kullanacağız. Bu yazılım "`https://miktex.org`" sitesinden indirilebilir.

LaTeX Kaynak Dosyaları

LaTeX, platformdan bağımsız herhangi bir editör ile kullanılabilir. Örnek olarak Windows için işletim sistemi ile gelen Notepad, Wordpad, MacOs için TexShop ve Linux için Vi metin editörleri verilebilir.

Windows işletim sistemi ile çalışan bazı LaTeX editörler:

TexMaker	`https://www.xm1math.net/texmaker/`
WinEdt	`http://www.winedt.com`
TeXnicCenter	`https://www.texniccenter.org`
TeXworks	`https://www.tug.org/texworks/`
TeXstudio	`https://www.texstudio.org`

Yukarıda listelenen editörlerin dışında onlarca daha editör bulunabilir. Burada vurgulamalıyız ki kullanılacak olan editör ara yüz ve bazı kısa yol kullanımı dışında bir şey fark ettirmeyecektir. Çünkü bu editörlerin kaynak dosyalarına yazılacak LaTeX komutları aynı olacaktır.

TexMaker

Dersimiz boyunca *TexMaker* editörünü kullanacağız.

Not: Yukarıda verilen editörler dışında ayrıca

`https://www.overleaf.com`

internet sitesinden herhangi bir kurulum yapmaksızın LaTeX kaynak dosyası ve bu dosyanın derlenmesi ile dvi veya pdf dosyası oluşturulabilir.

Diğer Dosyalar

Yukarıda listelenen editörler ile hazırlanmış LaTeX kaynak dosyasının (kaynak dosyasının uzantısı ".tex" olacaktır. Örneğin "deneme.tex" gibi) derlenmesi sonucunda kaynak dosyasın bulunduğu klasörde aşağıdaki dosyalar oluşabilir:

.aux	Bir derlemeden diğerine bilgilerin aktarıldığı ve doküman içindeki denklem veya sayfa atıflarının saklandığı dosya
.dvi	Aygıttan bağımsız dosya (device independent). Girdi dosyasını LATEX ile derlemenin başlıca sonucu bu dosyadır. İçeriğini bir DVI ön izleyici programla görebilir
.idx	Index Dizinin saklandığı dosya
.lof	Şekiller dizini (List of figures)
.log	Kaynak dosyası derlerken ekranda görülen tüm mesajların saklandığı dosya yani bir nevi kayıt defteri
.lot	Tablolar dizini (List of tables)

Kaynak dosyasının yapısı

Dokümanın kodlarını LaTeX formatında kaydettiğimiz dosyaya LaTeX kaynak dosyası denir. Daha önce de bahsettiğimiz gibi kaynak dosyalar herhangi bir metin editöründe oluşturulabilir.

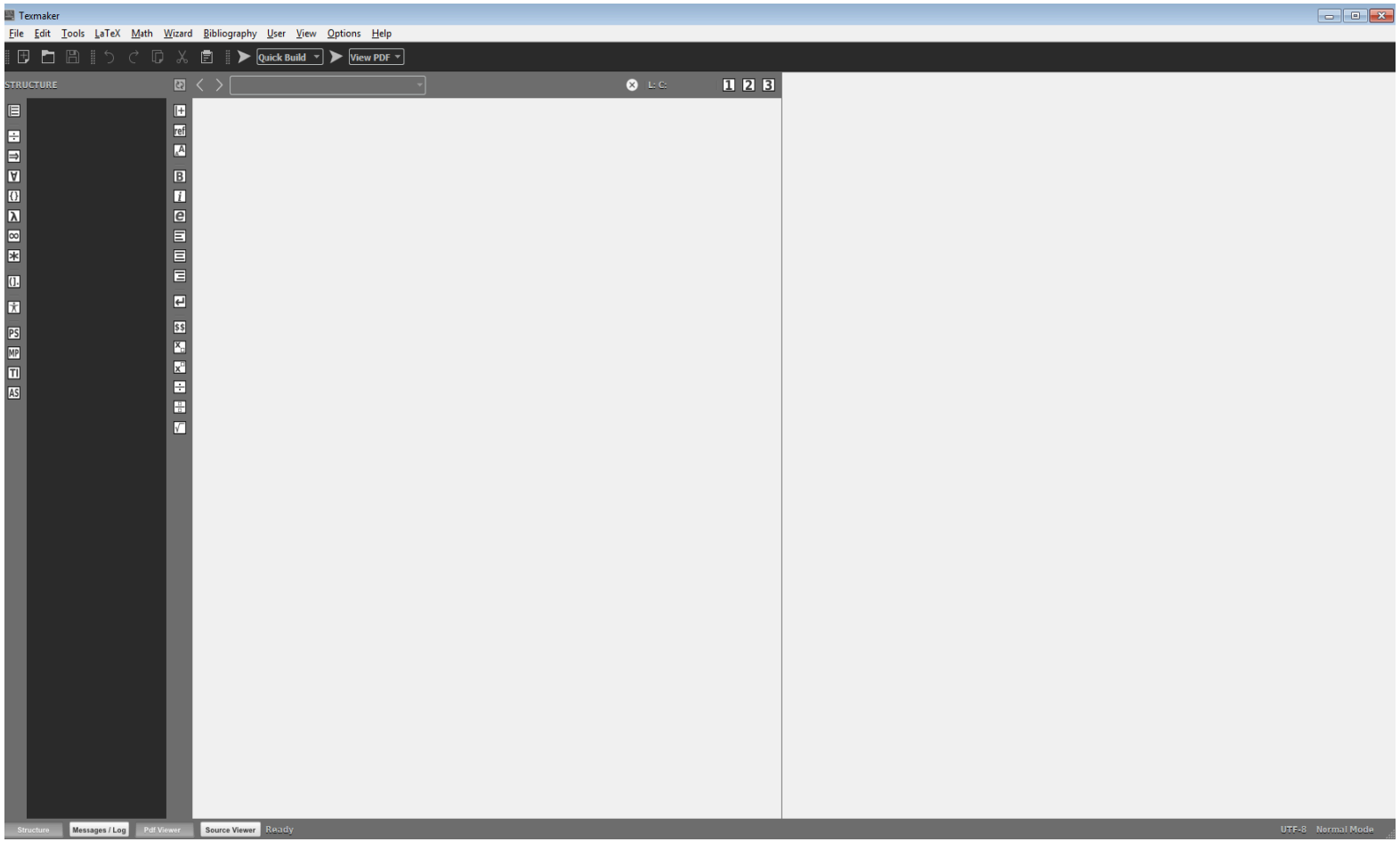
Kaynak dosya temel olarak iki kısma ayrılabilir: Sahanlık (Preamble) ve Gövde (Body).

Sahanlık kısmında belgemizin biçimlendirmesine ait bilgiler LaTeX komutları yardımıyla girilir. Belgede kullanılacak komutlar için gerekli olan paketler de bu kısma yazılır. Buraya girilen komutlar tüm belgeye otomatik olarak uygulanır. Bu bölümde girilen komutlar kaynak dosya derlendikten sonra çıktıda görünmez.

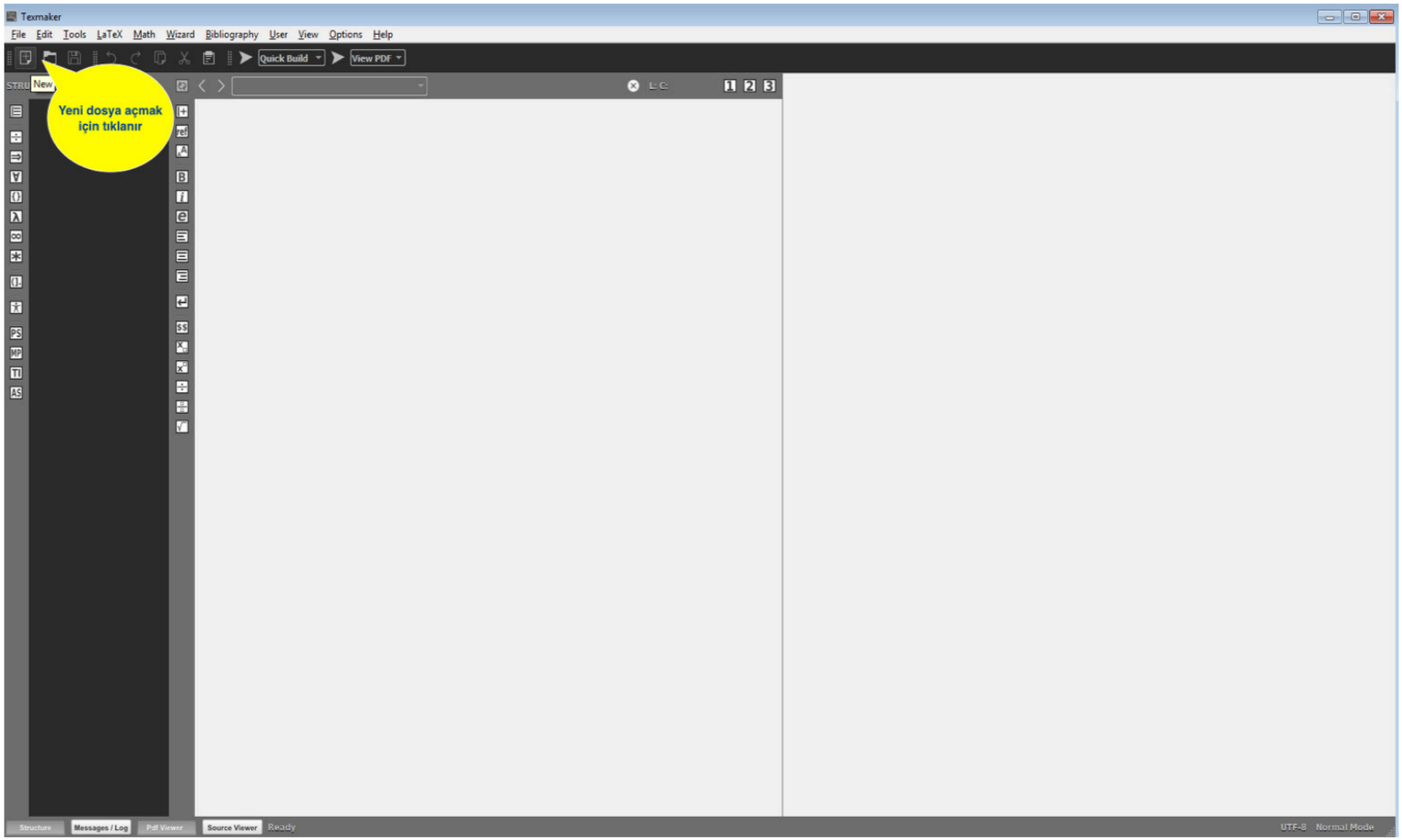
Gövde kısmında ise belgemizin çıktıda görünecek kısmı (metinler, tablolar, şekiller, vb.) girilir.

Basit bir LaTeX kaynak dosyası örneği

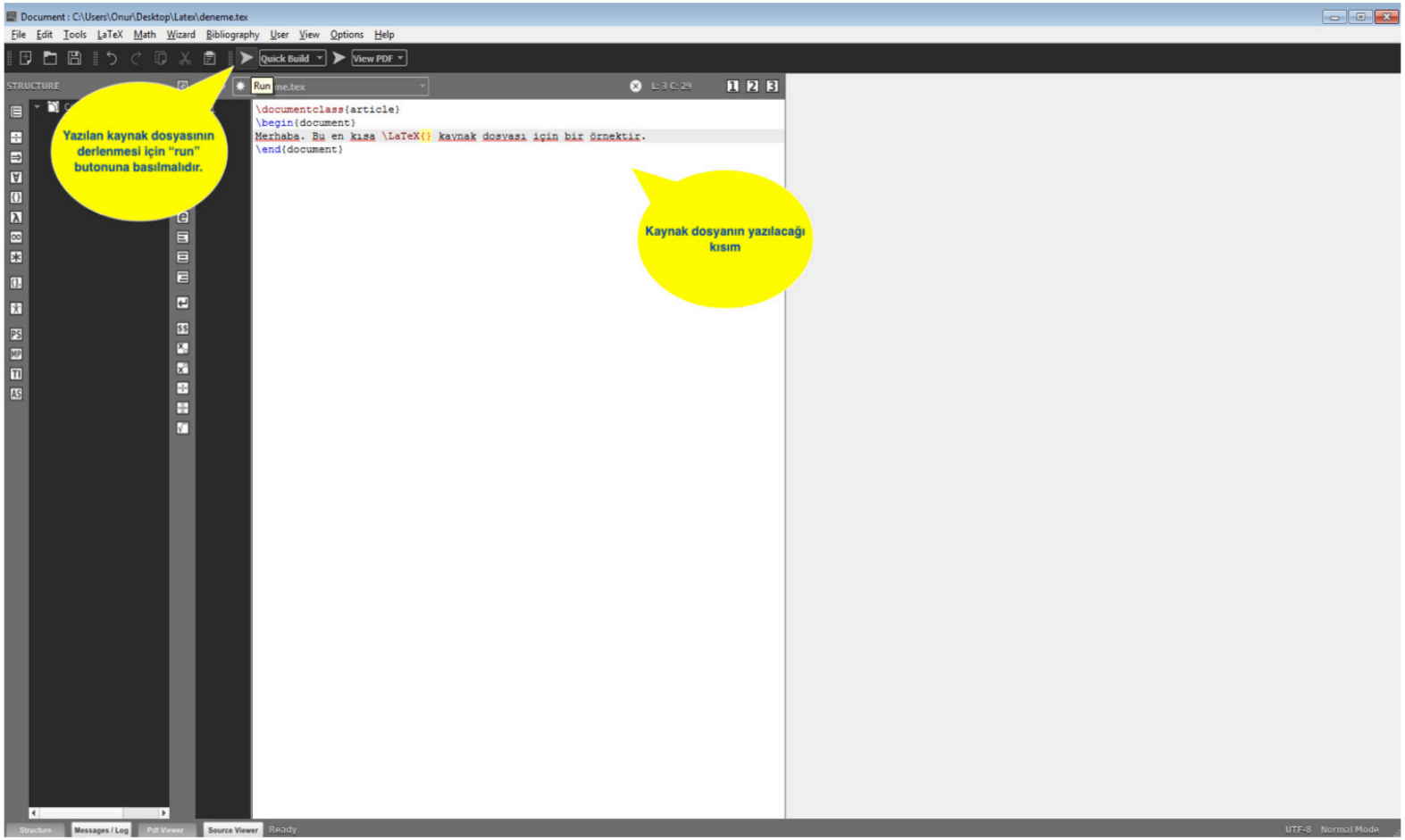
Şimdi bir LaTeX kaynak dosyasının nasıl hazırlanacağını görelim. İlk olarak TexMaker editörünü açalım:



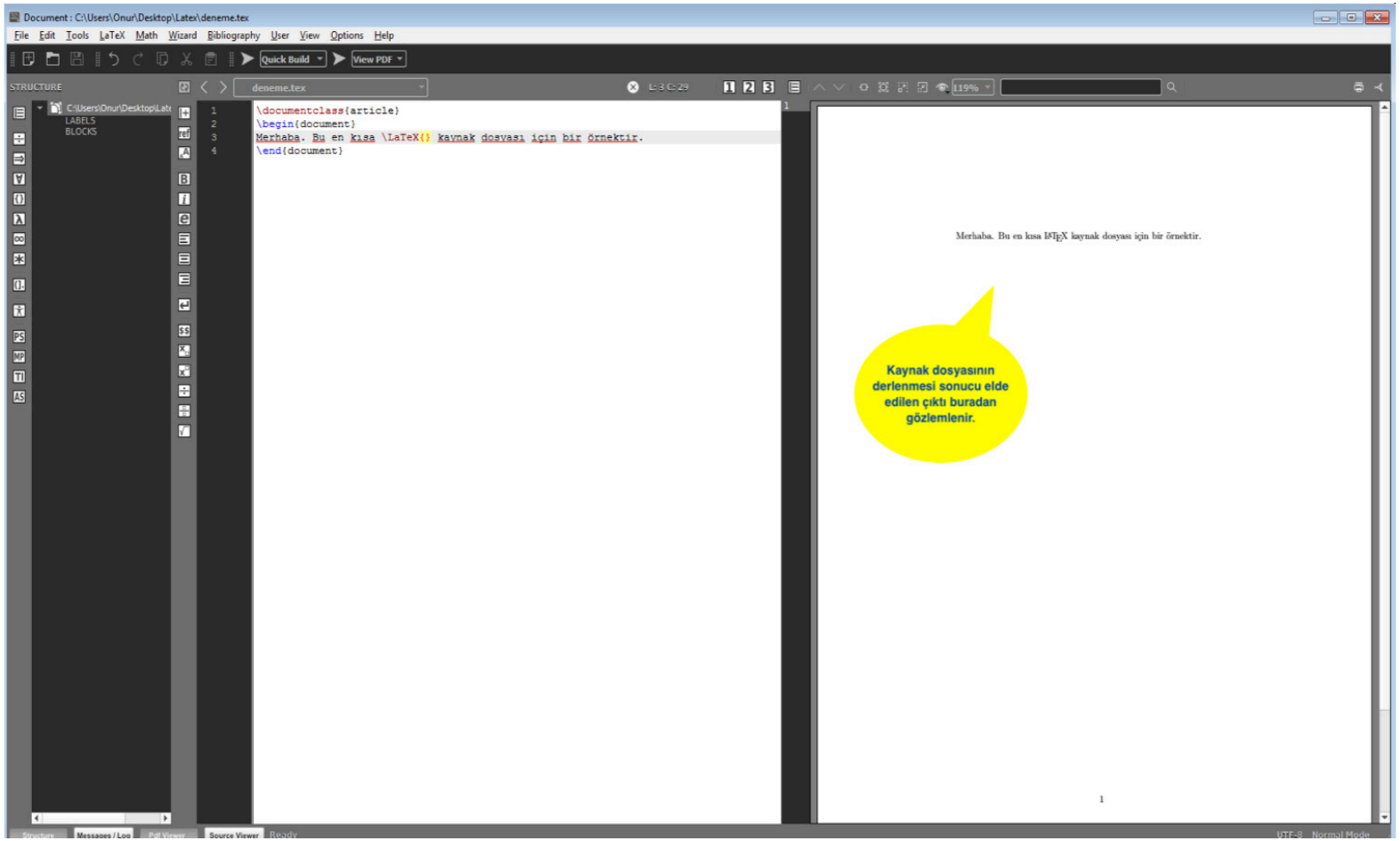
Daha sonra yeni çalışma dosyamızı açalım:



Açılan dosyanın içine gerekli komutları yazarak şekilde gösterilen buton ile derleme yapılır:



Böylece aşağıdaki şekilde de görüldüğü üzere yazdığımız kaynak dosyasının çıktısını elde ederiz:



Şimdi yukarıda yazdığımız kaynak dosyasındaki komutlara değinelim.

Burada \documentclass{} ile başlayıp \begin{document} ile biten kısma preamble (sahanlık), \begin{document} ile \end{document} arasındaki kısma ise body (gövde) denmektedir.

Bir kaynak dosyası belgenin türünü belirleyen \documentclass{} komutu ile başlamalıdır. Burada küme parantezi içerisine yazacağımız doküman türüne göre aşağıdaki tabloda listelenen ifadelerden biri gelir:

* LaTeX komutları \ ile başlar
* Komutlar büyük/küçük harfe duyarlıdır
* Tüm komutlar boşluk veya harf olmayan bir karakteri ile sonlandırılır
* Kaynak dosyasında % simgesinden sonra yazılan komut veya ifadeler derlendikten sonra ortaya çıkmaz

Temel doküman Türleri

article	Bilimsel makaleler ve kısa raporların yazıldığı en temel doküman tipi
report	Article dokuman tipine benzemesine karşın çok sayıda bölümden oluşan uzun raporlar, küçük kitapçıklar, doktora tezleri, vb. için kullanılan doküman tipi
book	Kitaplar için kullanılan dokuman tipi
letter	Mektup yazmak için kullanılan dokuman tipi
slides	Sunum hazırlamak için kullanılan dokuman tipi
beamer	Yine sunum hazırlamak için kullanılan dokuman tipi
proc	konferans bildirileri için, article sınıfından esinlenmiş dosya tipi

\documentclass[]{} komutuna [] şeklinde köşeli parantezi ile ek opsiyonlar eklenebilmektedir. Bu opsiyon sınıflarının bazıları aşağıda listelenmiştir:

10pt, 11pt, 12pt	Dokümanın ana yazı tipi puntosunu belirler. Bir değer belirtilmemişse, 10pt punto varsayılır.
a4paper, letterpaper	Kağıt boyutunu belirler. Varsayılan boyut Amerikan standardı olan letterpaper dir. Bunlara ek olarak a5paper, b5paper, executivepaper, ve legalpaper seçilebilir.
onecolumn, twocolumn	Dokümanın tek sütun veya çift sütun dizileceğini belirtir.
twoside, oneside	Dokümanın kağıdın hep tek tarafına mı yoksa iki tarafına mı basılacağını belirtir.
landscape	Dokümanı enine tutulmuş kağıda basılmak üzere hazırlar.

Not: 12pt'dan daha büyük fontta yazabilmek için "extarticle" doküman tipi kullanılabilir.

Paketler

Bir belge hazırlarken kimi zaman LaTeX komutları yeterli gelmemektedir. Örneğin bazı süslü karakterler kullanmak, grafik eklemek, bazı farklı matematiksel ifadeleri eklemek vb. gibi. Bu gibi ek işleri yapabilmemiz için Preamble kısmında bu işleri sağlayan paketlerin yüklenmesi gerekmektedir. Bir paket şu komutla etkin hale getirilir:

```
\usepackage[opsiyonlar]{paket}
```

Burada paket kısmına kullanılmak istenen paketin adı, opsiyonlar kısmına ise paketin çalışmasında kullanılacak ek opsiyonlar yazılır. Bazı paketler LaTeX kurulumu ile doğrudan gelmesine rağmen çoğu işlevsel paketi ayrıca yüklemek gerekir.

Uluslararası Diller

LaTeX'de İngilizce dışındaki dillerden biriyle yazmak istenildiğinde kaynak dosyanın preamble kısmında bazı ayarların yapılması gerekmektedir. Bunlar:

1. LaTeX'in otomatik olarak ürettiği (İçindekiler, Şekiller Listesi, $\cdots$) başlıkların yeni dile uyarlanması gerekir. Bu uyarlama babel paketinin yüklenmesi ile mümkündür.
2. LaTeX'in yeni dildeki heceleme kurallarını bilmesi gerekir.
3. LaTeX'in o dile has dizgi kurallarını bilmesi gerekir.

Dolayısıyla ilk olarak dosyamızın preamble (sahanlık) kısmına

```
\usepackage[dil]{babel}
```

yazılmalıdır. Burada dil kısma yazım yaptığımız dil adını yazmamız gerekmektedir.

Birden fazla dil yüklemek için ise

```
\usepackage[turkish,french,spanish]{babel}
```

şeklinde dil kısmına dilleri virgül ile ayırarak ekleyebiliriz. Bu komutla listedeki dillerin sonuncusu etkin hale gelir. Etkin dili değiştirmek için

```
\setlanguage{turkish}
```

komutu ile istenilen dil seçilebilir.

LaTeX'in farklı dillerdeki değişik alfabetik karakterleri destekleyebilmesi için \inputenc paketi kullanılır. Bunun için preamble kısmına

```
\usepackage[kodlama]{inputenc}
```

komutu yazılmalıdır. Burada "kodlama" kısmına 'latin1, latin5, ansinew, utf8,$\cdots$" gibi kodlama sistemleri yazılabilir.

Yazı tipi kodlaması ile ilgili diğer önemli bir paket de "fontenc" paketidir. Bu paket hecelemelerin doğru bir şekilde yapılmasını sağlar. LaTeX'in varsayılan yazı tipi kodu *0T1* olup 7-bitlik ASCII sembol setinde 128 sembol saklayabilir. 8-bitlik yazı tipi takımları için geliştirilmiş *T1* kodlaması pek çok Avrupa dilindeki özel karakterleri de kapsayan en genel kullanılan kodlamalardan biridir. Kiril ve Yunan alfabeleri için farklı kodlama tipleri mevcuttur. Bu paket kaynak dosyasının preamble kısmına

```
\usepackage[T1]{fontenc}
```

şeklinde yazılır.

Türkçe Yazmak

İlk olarak dokümanımızda LaTeX'in otomatik olarak ürettiği (İçindekiler, Tablolar Dizini) başlıkların Türkçeleşmesi için \documentclass{}'dan sonra

\usepackage[turkish]{babel}

komutu eklenmelidir.

Daha sonra Türkçe karakterlerin dizilebilmesi için

\usepackage[latin5(veya utf8)]{inputenc}

komutu girilmelidir.

Son olarak hecelemenin doğru olarak yapılabilmesi için

\usepackage[T1]{fontenc}

yazılmalıdır.

Not: latin5 veya utf8 kodlamasının seçimi kullanılan editörün hangi kodlamayı kullandığına bağlıdır. Eğer editörümüz metni ANSI olarak kodlamışsa **latin5**, UTF-8 olarak kodlamışsa da **utf8** kullanmalıyız.

Son güncelleştirmeler ile Türkçe karakterlerin yazılması için kullanılması gereken \usepackage[latin5]{inputenc} veya \usepackage[utf8]{inputenc} komutlarının kullanım zorunluluğu ortadan kalkmıştır. Fakat güncel olmayan sistemin kullanılması durumunda bu komutların girilmesi gerekir.

Şimdi yukarıda gördüğümüz komutları kullanarak yeni bir LaTeX dosyası hazırlayalım:

```
1  \documentclass[11pt,a4paper]{article}
2  \usepackage[turkish,shorthands=off]{babel}
3  \usepackage[T1]{fontenc}
4  \usepackage[latin5]{inputenc}
5  \title{Kısa Bir Deneme}
6  \author{Onur Şahin}
7  \date{}
8  \begin{document}
9  \maketitle
10 Bu kısma \LaTeX hakkında bir şeyler yazacağız
11 \end{document}
```

Yukarıdaki komutların sonucunda aşağıdaki çıktı elde edilir:

Kısa Bir Deneme

Onur Şahin

Bu kısıma LaTeXhakkında bir şeyler yazacağız.

1

Yukarıdaki örnekten de görüleceği üzere \title{} komutu ile yazımızın başlığı, \author{} komutu ile bu çalışmanın yazarlarını ve \date{} ile de tarihi belirtiyoruz. Tüm bu ifadelerin dosyamızda görünür olabilmesi için ise. \maketitle komutunu kullanıyoruz.

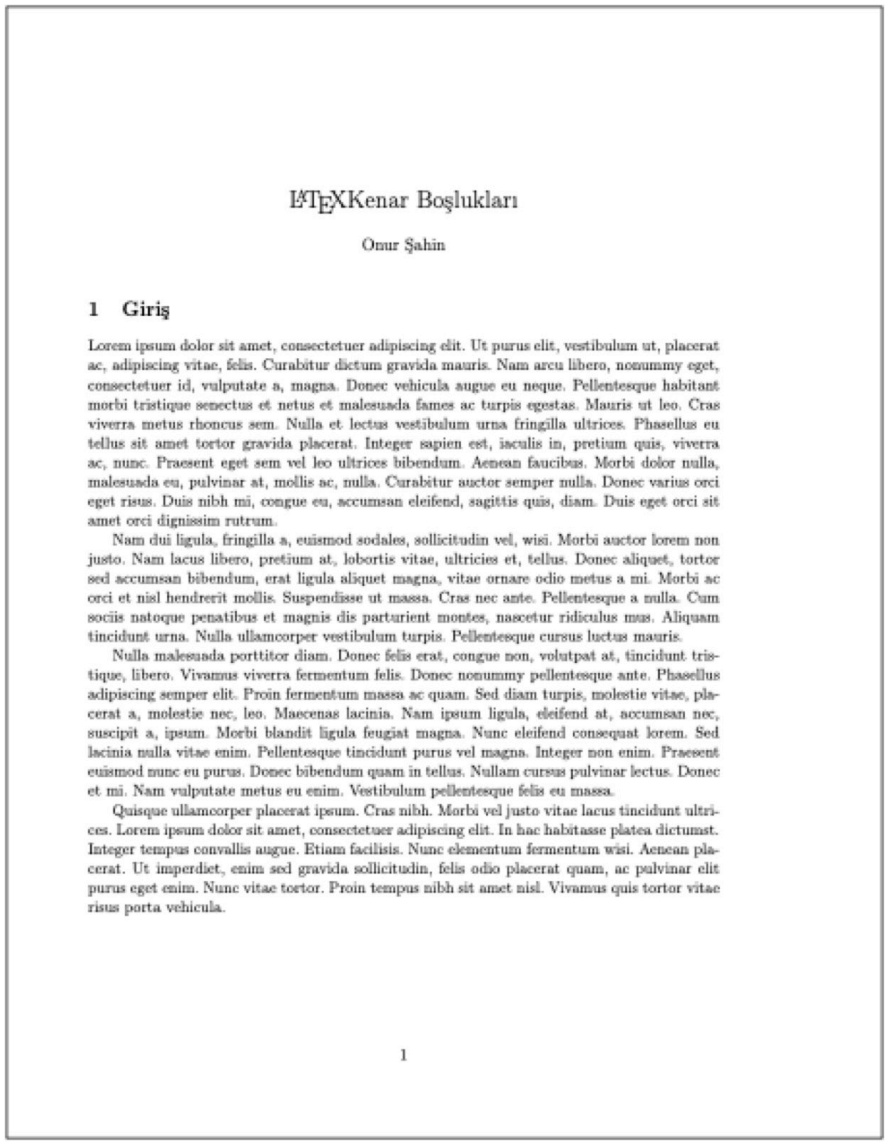
Bu çıktıdan görülebilmektedir ki yazımızın kenar boşlukları oldukça fazla. LaTeX'de sayfa düzenini istediğimiz şekilde belirlemek mümkün. Bunun için gerekli paketi yükleyerek istediğimiz ayarları belirleyebiliriz.

Kenar Boşlukları

Bir belgeyi yazmaya başlamadan önce yazdıklarımızın nasıl görüneceğini belirleyen bazı ayarları yapmamız gerekmektedir. Bunlardan ilki kenar boşluklarıdır. Hazırlamış olduğumuz bir belgenin kenar boşluklarını ayarlayabilmek için \usepackage[opsiyon]{geometry} paketini kullanabiliriz. Burada opsiyon kısmına sayfanın istenilen kenar boşlukları için gerekli değerler girilecektir. Örneğin:

```
1  \documentclass[11pt,a4paper]{article}
2  \usepackage[turkish,shorthands=off]{babel}
3  \usepackage[T1]{fontenc}
4  \usepackage[latin5]{inputenc}
5  \usepackage[left=2cm,right=4cm,top=3cm,bottom=3cm]{geometry}
6  \usepackage{lipsum}
7  \title{\LaTeX Kenar Boşlukları}
8  \author{Onur Şahin}
9  \date{}
10 \begin{document}
11 \maketitle
12 \lipsum[1-4]
13 \end{document}
```

Komutu sonucu aşağıdaki gibi kenar boşlukları belirlediğimiz şekilde bir sayfa üretilir:



LaTeXKenar Boşlukları

Onur Şahin

1 Giriş

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

1

Satır Aralığı

LaTeX'de satır aralıklarını ayarlamak için "setspace" paketi kullanılmalıdır. Bu paket \usepackage{setspace} şeklinde yüklendikten sonra satır aralıklarını

\singlespacing
\onehalfspacing
\doublespacing
\setstretch{değer}

komutlarıyla belirleyebiliriz. İlk üç komut sırasıyla bir, bir buçuk ve iki satır aralığı için kullanılırken son komut satır aralığını istediğimiz bir değer olarak belirlememize olanak sağlar.

Satır aralık komutları iki şekilde uygulanabilir. İlk olarak eğer bu komutlardan birini "preamble" kısmına yani "\begin{document}" komutundan önce yazarsak tüm dosyadaki metinlere uygulanmış olur.

Eğer dosyamızdaki metinleri farklı satır aralıkları olacak şekilde yazmak istiyorsak "\begin{document}" komutundan sonra farklı satır aralıklara sahip olacak metinleri

```
\begin{doublespacing}
...
\end{doublespacing}
```

veya özel aralık vermek için

```
\begin{spacing}{istenilen değer}
...
\end{spacing}
```

komutları arasına yazarak elde etmek mümkündür.

Belge içerisinde yalnızca belirli satırların arasındaki boşluğu değiştirmek istiyorsak bu satırlardan önce

```
\setlength{\baselineskip}{x\baselineskip}
```

komutu yazılmalıdır. Burada "x" kısmına yazılacak değer ile standart satır aralığının kaç katı satır aralığı olacağı belirtilmektedir.

```
\documentclass[11pt,]{article}
\usepackage[turkish,shorthands=off]{babel}
\usepackage[T1]{fontenc}
\usepackage[latin5]{inputenc}
\usepackage{lipsum}
\usepackage{setspace}
\usepackage[left=2cm,right=4cm,top=3cm,bottom=3cm]{geometry}
\title{\LaTeX Kenar Boşlukları}
\author{Onur Şahin}
\date{}
\begin{document}
\maketitle
\begin{spacing}{1}
\lipsum[1]
\end{spacing}
\begin{spacing}{2}
\lipsum[2]
\end{spacing}
\begin{spacing}{3}
\lipsum[3]
\end{spacing}
\end{document}
```

LaTeXKenar Boşlukları

Onur Şahin

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed

1

Paragraf Ayarları

Metindeki paragraf arasındaki boşlukların ayarlanması

```
\setlength{\parskip}{istenilen değer}
```

komutunun preamble'a eklenmesi ile mümkündür.

Paragraf başlarındaki girintinin ne kadar olacağını ayarlamak için

```
\setlength{\parindent}{istenilen değer}
```

komutuyla ayarlayabiliriz. Burada dikkat edilecek bir hususta girinti verildiğinde metinde bölüm başlığının hemen altındaki satır başı dışındaki tüm satır başlarında girinti verilmiş olur. Bölüm başlığının altındaki satır başının da girintili olması için

```
\usepackage{indentfirst}
```

paketinin yüklenmesi gereklidir.

Bir satırın özel olarak satır başından başlamasını istiyorsak satırın öncesine \indent, eğer satır başı yapmamasını istiyorsak \noindent komutlarını eklememiz yeterlidir.

Sayfa numaraları

Yazmış olduğumuz bir dosyada sayfa numaralarının sayfanın neresinde ve ne şekilde olacağı, hatta nerede başlayıp nerede duracağını belirlememiz mümkündür.

İlk olarak sayfa numaralarının sayfanın neresinde görüneceğini belirlemek için preamble kısmında

```
\pagestyle{opsiyon}
```

komutu kullanılır. Burada opsiyon kısmına "plain, headings, empty" seçeneklerinden biri gelebilir. Plain, varsayılan numaralandırma biçimi olup sayfa numarasını sayfanın en altında ortalanmış olarak yazdırır. Headings, sayfa numarasını sayfanın en üstünde sağa yaslanmış olarak yazdırır. Son olarak empty seçeneği ise sayfa numarasının yazılmamasını sağlar.

Yukarıdaki komutların dışında sayfa numaralarını sayfanın istenilen yerine yazılmasını sağlayan daha genel komutlarda mevcuttur. Bunun için \usepackage{fancyhdr} paketini yüklememiz gerekmektedir. Bu paket yüklendikten sonra preamble kısmına

```
\pagestyle{fancy}
\fancyhf{}
\lhead{\thepage}
\renewcommand{\headrulewidth}{0pt}
```

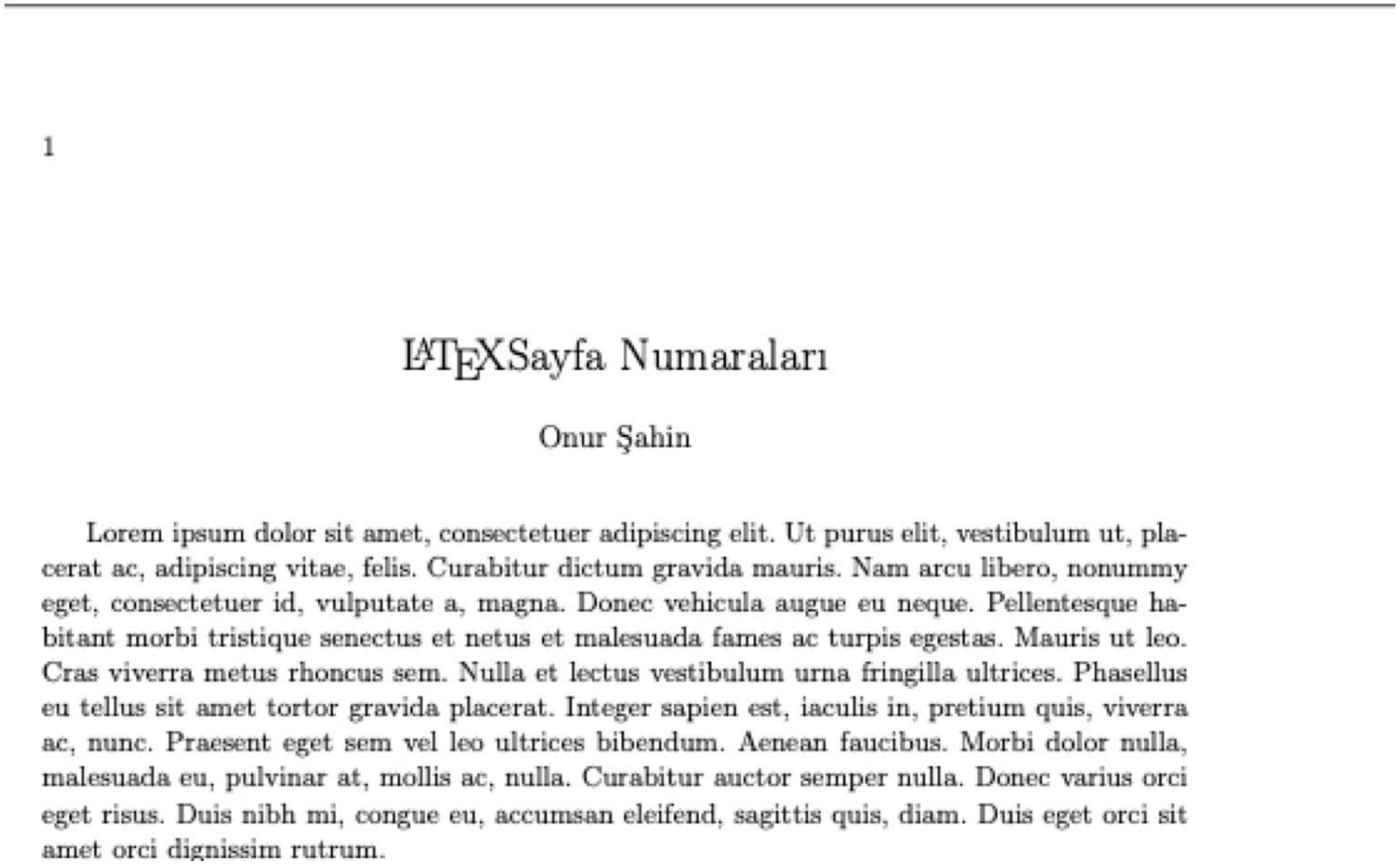
komutları girilmelidir. Burada "lhead" 'de l left(sol), head(yukarı) kısmını ifade ettiğinden sayfa numarası sol üst köşeye yazdırılır. Bu kısma "rhead, chead, lfoot, rfoot ve cfoot" seçenekleride gelebilir. Bu ifadeler sırasıyla sayfa numarasının sayfanın sağ üst, orta üst, sof alt, sağ alt ve orta alt kısmına yazdırılmasını sağlar.

Bu komutlar ile sayfanın üstünde bir başlık çizgisi oluşmaktadır. "\renewcommand{\headrulewidth}{0pt}" komutu ile bu çizginin kalınlığını sıfır yaparak görünmemesini sağlıyoruz.

Örneğin sayfa numarası sol üstte olan bir dokümanın LaTeX kodu:

```
\documentclass[11pt,a4paper]{article}
\usepackage[turkish,shorthands=off]{babel}
\usepackage[T1]{fontenc}
\usepackage[latin5]{inputenc}
\usepackage[left=2cm,right=4cm,top=3cm,bottom=3cm]{geometry}
\usepackage{lipsum}
\usepackage{fancyhdr}
\pagestyle{fancy}
\fancyhf{}
\lhead{\thepage}
\renewcommand{\headrulewidth}{0pt}
\renewcommand{\footrulewidth}{0pt}
\title{\LaTeX Sayfa Numaraları}
\author{Onur Şahin}
\date{}
\begin{document}
\maketitle
\thispagestyle{fancy}
\lipsum[1-20]
\end{document}
```

Böylece aşağıdaki çıktı elde edilir:



1

LaTeXSayfa Numaraları

Onur Şahin

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Not: Yapılan bu sayfa numaralandırması "title" sayfasına etki etmediğinden dolayı "\maketitle" komutundan sonra "\thispagestyle{fancy} komutu kullanılmıştır.

Sayfa numaralarının hangi türde olacağını belirtmek için

\pagenumbering{tür}

komutu kullanılır. Burada "tür" kısmına aşağıdaki ifadelerden biri gelebilir:

arabic:	Standart numaralar (1, 2, 3, 4,...)
roman:	Küçük Roma rakamları (i, ii, iii, iv...)
Roman:	Büyük Roma rakamları (I, II, III, IV...)
alph:	Küçük harfler (a, b, c, d,...)
Alph:	Büyük harfler (A, B, C, D,...)

"\pagenumbering" komutu preamble kullanılırsa tüm dokümanı etkiler. Eğer dokümanın belli bir kısmı bir türde, diğer kısmını başka türden numaralandırmak istiyorsak her kısmın başına istenilen türde komut eklemeliyiz.

Örneğin dokümanın ilk 3 sayfasını küçük Roma rakamları geri kalanını standart rakamlar ile numaralandırmak istiyorsak: ilk üç sayfadan önce \pagenumbering{roman} ve 4. sayfanın başına da \pagenumbering{arabic} yazmamız gerekmektedir.

Eğer bu örnekte 4. sayfanın "1" değil de "4" ile numaralandırılmasını istiyorsak \pagenumbering{arabic} komutundan sonra \setcounter{page}{4} komutunu eklememiz gerekir.

Bir sayfanın numaralandırmamasını istiyorsak o sayfanın başına

\thispagestyle{empty}

yazmamız yeterlidir.

Kelime arası boşluklar

LaTeX dosyasında kelimeler arasındaki boşlukların sayısından bağımsız olarak LaTeX tek bir boşluk olarak kabul eder. Yani iki kelime arasında ne kadar boşluk koyarsanız koyun doküman derlediğinizde çıktıda tek boşluk görürsünüz.

Paragraflar, satırlar, şekiller vb. yapılar arasında dikey boşluk bırakmak için

```
\vspace{istenilen değer}
```

komutu kullanılabilir. Benzer şekilde biten bir satır veya bir şekilden sonra

```
\\[istenilen değer]
```

komutu ile de dikey boşluklar ayarlanabilir.

Yatay boşluklar ise

\hspace{istenilen değer}

komutuyla ayarlanabilmektedir. Bunun yanı sıra: \, \,, \:, \:,...gibi komutlar ile de küçük yatay boşluklar verilebilir.

Bazı özel karakterlerden sonra boşluk bırakılsa dahi çıktıda boşluksuz olarak görünmektedir. Örneğin "\LaTeX kaynak dosya" ifadesini derlediğimizde "LaTeX kaynak dosya" şeklinde bir çıktı elde ederiz. {} ikili küme parantezi tek bir boşluk anlamına gelmektedir. Yani yukarıdaki ifade "\LaTeX{} kaynak dosya" şeklinde yazılırsa "LaTeX kaynak dosya" elde edilir.

Not: Yukarıda ve LaTeX'deki diğer komutlarda uzunluk olarak kullanılacak birimler: "mm, cm, in, pt, em, ex,..." 'den herhangi biri olabilir:

mm	milimetre $\approx 1/25$inch	pt	punto $\approx 1/3$mm
cm	santimetre=10mm	em	'M' harfinin genişliği
in	inch=25.4mm	ex	'x' harfinin yüksekliği

Satır ve sayfa kesme

Yazdığımız bir dokümanda yeni bir paragraf açmadan alt satıra geçmek istiyorsak

\\ veya \newline

komutları kullanılabilir.

*

komutu ise yeni bir sayfaya geçmeden alt satıra geçilmesini sağlar.

Yeni bir paragraftan başlamak için ise boş bir satır bırakmak ya da \par komutunu kullanmak yeterlidir.

\newpage

komutu ile de yeni bir sayfa başlamaktadır.

\linebreak[n], \nolinebreak[n], \pagebreak[n], \nopagebreak[n]

komutları sırasıyla satır kes, satır kesme, sayfa kes ve sayfa kesme komutlarıdır. Burada n parametresi 0 ile 4 arasında olup, n 4'den küçük seçilirse, sonucun kötü olmasında LaTeX'in istediğimizin göz ardı etmesini sağlar.

Burada \newline ile \linebreak[n] komutları arasında ufak bir fark vardır. \linebreak[n] komutunu verdiğimizde LaTeX yarım kalan satırı hala sağa yaslamaya çalışır fakat \newline komutunda hemen alt satıra geçilir.

Satırları hizalama

LaTeX satırları sayfanın iki tarafına yaslanmış ve eşit uzunlukta olacak şekilde yazmaya çalışır. Fakat biz satırları sağa, sola veya ortaya gelecek şekilde de yazabiliriz. Bunun için metinleri sırasıyla

```
\begin{flushleft}          \begin{flushright}          \begin{center}
...                    ,   ...                    ,   ...
\end{flushleft}            \end{flushright}            \end{center}
```

komutları arasına yazabiliriz.

Kelime Bölme

LaTeX satır sonlarında belirlenen sayfa ayarlarına sığmayan kelimeleri hecelere böler. Kimi zaman bu hecelemeler doğru olmayabilir. Bu gibi durumlarda kelimenin nereden bölünmesi gerekli ise o hecenin başına \- simgesini koymamız yeterlidir. Örneğin: "bu kelime buradan bö\-lünsün". Ya da bölünecek kelimenin nasıl hecelendiğini bö\-lün\-sün belirterek LaTeX'e kelimenin nasıl bölüneceğini öğretebiliriz.

Bir veya daha fazla kelimenin tüm metinde nasıl bölüneceğini tek tek kelimeleri "\-" ile bölerek yazman çok zordur. Bunun yerine dokümanın başında bir kelimenin nasıl heceleneceğini LaTeX'e tanıtmak daha kolay olur. Bunun için \hyphenation{kelime} komutunu kullanmalıyız. Örneğin: dokümanın preamble kısmında \hyphenation{he-ce-le-me} komutu girilirse LaTeX tüm doküman boyunca bu kelimenin gösterildiği şekilde hecelendiğini anlar ve kelimeyi bu hecelemeye uygun şekilde böler.

Özel Karakterler ve Simgeler

LaTeX'de özel karakterlerin (\, $, %, &, ^, _,#,) pek çoğunun özel bir işlevi vardır. Dolayısıyla bu karakterleri metin içerisinde görüntüleyebilmek için başlarına \ (ters bölü) simgesini yazmak gerekir. Fakat bu durum \karakteri için geçerli değildir. Çünkü meydana gelecek \\ ifadesi yeni satıra geçmek için kullanılan komutlardan biridir. Dolayısıyla \karakterini yapmak için metin içerisinde \textbackslash komutu kullanılabilir.

Özel karakterleri yazmak için diğer bir yöntem ise yazmak istenilen ifadeyi \verb|...| komutunun içine yazmaktır. Bu durumda LaTeX yazılan karakterin veya komutun metin olarak yazılmak istendiğini anlar ve onu bir karakter ya da komut olarak çalıştırmaz. Benzer işleve sahip diğer bir yöntem ise istenilen ifadenin

\begin{verbatim}...\end{verbatim}

komutu arasına yazılmasıdır.

LaTeX birçok dilde bulunan aksanlı harfleri ve işaretleri desteklemektedir. Örneğin İngilizcede bulunmayan Türkçe karakterlerin kullanılabilmesi için latin5 kodlamasını çalıştırmamız gerektiğini önceki bölümlerde görmüştük. latin5 kodlaması Türkçe dışında pek çok Avrupa dilindeki özel karakterleri de desteklemektedir. Eğer İngilizce veya Türkçe karakterleri içermeyen başka bir klavye kullanıyorsak özel Türkçe karakterler şu şekilde yazılabilir:

Â, â	\^{A}, \^{a}
Ç, ç:	\c{C}, \c{c}
Ğ, ğ:	\u{G}, \u{g}
İ, ı:	\.I, \i
Ö, ö:	\"O, \"o
Ş, ş:	\c{S}, \{s}
Ü, ü:	\"U, \"u

Metin Stilleri

LaTeX'de bir metnin kalın, italik ve altı çizili olarak yazılması isteniyorsa ilgili ifadeler sırasıyla

\textbf{}, \textit{}, \underline{}

komutlarının içerisine yazılmalıdır. Örneğin kaynak dosyaya

```
Bir metni \textbf{kalın}, \textit{italik} ve
\underline{altı çizgili} yazmak için yazılması
gereken komutlar yukarıda verilmiştir.
```

ifadesi eklendiğinde elde edeceğimiz çıktı

Bir metni **kalın**, *italik* ve altı çizgili yazmak için yazılması gereken komutlar yukarıda verilmiştir

şeklinde olacaktır.

Bir kelimeye vurgu yapmak istiyorsak \emph{} komutu kullanılır. Bu komut normal yazılan metinde vurgu yapılan kelimeyi italik, italik yazılan metinde kelimeyi normal olarak yazdırır.

Bir cümledeki tüm kelimeleri küçük veya büyük yazdırmak için sırasıyla

\uppercase{}, \lowercase{}

komutları kullanılabilir. Bu komutlar özel karakterlerin büyük ya da küçük yazılmasında tam doğru sonuç vermediğinden daha güçlü komutlar olan

\MakeUppercase{}, \MakeLowercase{}

komutları da kullanılabilir.

Bir cümledeki kelimelerin yalnızca ilk kelimelerini büyük yazmak için \usepackage{mfirstuc} paketi ile birlikte

\capitalisewords{}

komutu kullanılabilir.

Yazı Tipi

Hazırladığımız belgedeki tüm metnin yazı tipini ve büyüklüğünü kaynak dosyanın preamble kısmında belirleyebiliriz. Fakat özel olarak belge içinde bazı yerlerde yazı tipini ve büyüklüğünü istediğimiz şekilde değiştirmemiz de mümkündür.

Eğer kaynak dosyamızın başında bir değişiklik yapılmadıysa yazdığımız belgeniz yazı tipi fontu Roman'dır. Fakat belge içerisinde bu fontu Sans Serif ve Typewriter font tipleri ile değiştirebiliriz. Bunun için gerekli olan komutlar şu şekildedir:

Roman	:	\textrm{}
Sans Serif	:	\textsf {}
Typewriter	:	\texttt {}

Yukarıda verilenlerden farklı font tipleri kullanabilmek için dokümanın başında o fonta ait style dosyasının yüklenmesi gerekmektedir. Bu font tiplerinin kaynak dosyada yazımı ve bunun sonucunda çıktısı ise şu şekildedir.

```
\textrm{Roman yazı tipi}, \textsf{Sans Serif yazı
tipi} ve \texttt{Typewriter yazı tipi}
```

Roman yazı tipi, Sans Serif yazı tipi ve `Typewriter yazı tipi`

Bir font sitili ile yazarken başka bir font sitiline geçmek için

Roman	:	\rmfamily
Sans Serif	:	\sffamily
Typewriter	:	\ttfamily

komutları kullanılabilir. Örneğin:

```
\texttt{Bu komut ile cümlenin sitili değişecektir.}
\sffamily
Bu komuttan sonra cümlenin sitili belirtilen sitil
ile değişir.
```

`Bu komut ile cümlenin sitili değişecektir`
Bu komuttan sonra cümlenin sitili belirtilen sitil ile değişir.

Yazı Büyüklüğü

Yazı boyutunun değiştirilmesi için gerekli komutlar ve bu komutlar sonucu elde edilen çıktı aşağıdaki tabloda verilmiştir:

{\tiny Büyüklük }	:	Büyüklük
{\scriptsize Büyüklük}	:	Büyüklük
{\footnotesize Büyüklük}	:	Büyüklük
{\small Büyüklük}	:	Büyüklük
{\normalsize Büyüklük }	:	Büyüklük
{\large Büyüklük}	:	Büyüklük
{\Large Büyüklük }	:	Büyüklük
{\LARGE Büyüklük}	:	Büyüklük
{\huge Büyüklük}	:	Büyüklük
{\Huge Büyüklük }	:	Büyüklük

10pt, **11pt** ve **12pt** font büyüklüklerinin yukarıdaki komutlar ile nasıl değişeceği aşağıda verilen tablodan görülebilir:

	Standart Font Büyüklüğü		
Komut	10pt	11pt	12pt
\tiny	5pt	6pt	6pt
\scriptsize	7pt	8pt	8pt
\footnotesize	8pt	9pt	10pt
\small	9pt	10pt	11pt
\normalsize	10pt	11pt	12pt
\large	12pt	12pt	14pt
\Large	14pt	14pt	17pt
\LARGE	17pt	17pt	20pt
\huge	20pt	20pt	25pt
\Huge	25pt	25pt	25pt

Eğer metin içerisindeki yazının boyutunu özel bir değer ile değiştirmek istiyorsak ilgili metin

{\fontsize{size}{baselineskip}\selectfont Metin}

komutundaki "metin" kısmına girilmelidir. Bu komutta "size" kısmı yazının boyutunu, "baselineskip" ise ardışık iki satır arasındaki boşluğu belirler. Genel kural olarak "baselineskip" font boyutunun 1.2 katıdır. Fakat daha farklı satır aralıkları için bu değer istenildiği şekilde değiştirilebilir. Örnek vermek gerekirse:

```
{\fontsize{15pt}{18pt}\selectfont \lipsum[1][1-4]}
```

komutu

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

Sayfayı Sütunlara Ayırma

Bir dokümanın genel olarak tek veya iki sütun olarak yazılmasının \documentclass[]{} dokümanın başlangıç komutundaki seçeneklerin yazıldığı köşeli parantez içine yazılarak mümkün olduğunu görmüştük. Eğer doküman içinde yalnızca belli bölgeleri tek veya iki sütun olarak yazmak istiyorsak bu kısımlardan önce \onecolumn veya \twocolumn komutları yazılmalıdır. Bu iki komutta verildikleri yerden yeni bir sayfa başlatmaktadırlar.

Dipnotlar

Bir metinde bahsedilen konu hakkında kullanılan kaynağa atıf vermek ya da daha ayrıntılı bilgi vermek için dipnotlar kullanılabilir. LaTeX'de dipnot vermek için
\footnote{dipnot metni}
komutu kullanılır.

Kutular

LaTeX'de bir metni kutu içine almanın birden fazla yolu vardır. Bunlardan ilki:

\framebox[genişlik][hizalama]{metin}

komutudur. Burada "metin" kısmına kutu içerisine alınmak istenen ifade yazılmalıdır. Genişlik seçeneği kutunun genişliğini vurgulayıp: *mm*, *cm*, *in* gibi değerler ile belirlenebilir. Eğer kutunun tam bir satır genişliğinde olması isteniyorsa bu seçeneğe "\textwidth" yazılır. Kutunun genişliğinin içerisindeki metinin genişliği kadar olması isteniyor ise "\width" yazılmalıdır.

"hizalama" seçeneği ise meni kutunun içerisinde nasıl hizalanacağını belirler. Bu kısma yazılabilecek ifadeler şunlardır:

- l: sola hizalama
- r: sağa hizalama
- c: ortalama
- s: iki yana yaslama

Bu komut ile ilgili birkaç örnek verelim:

```
\framebox[10cm][r]{Kutu içerisinde bir metin}
```

Kutu içerisinde bir metin

```
\framebox[2cm][l] {Kutu içerisinde bir metin}
```

Kutu içerisinde bir metin

Bir metnin çevreleyen bir kutu oluşturmak için \fbox{metin} ve \frame{metin} komutları da kullanılabilir.

framebox, **fbox** ve **frame** komutları yalnızca tek satırdan meydana gelen kutular oluştururlar. Eğer kutu içine almak istediğimiz metin bir satırdan fazla ise bu komutlar işe yaramayacaktır. Bu durumun üstesinden gelmek için iki yol vardır. Birincisi,

\parbox[konum]{genişlik}{metin}

komutudur. Burada konum parametresi kutunun bulunduğu yere göre düşey konumunu belirler ve **c (orta)**, **l (sol)** ve **r (sağ)** seçeneklerinden biri olabilir.

\parbox komut ile yazılan metin bir kutu içinde yazılmaz. Metnin kutu içinde olması için

```
\framebox{\parbox[konum]{genişlik}{metin}}
```

şeklinde yazılmalıdır.

Birden fazla satırı kutu içinde yazdırmak için ikinci komut ise

```
\begin{minipage}[konum]{genişlik}
  metin
\end{minipage}
```

komutudur. Bu komut aslında sayfayı sütunlara bölerek her sütuna girdi yazılmasını sağlar. Yani art arda yazılan bu komut ile sayfa o kadar sütuna bölünmüş olur. *konum* parametresi "\framebox" komutundakine benzer seçeneklerden birini alır. Ayrıca yine bu komut ile yazılan metin kutu içerisinde olmadığından ifadeyi kutuya alabilmek için bu komut \framebox{} komutunun içinde yazılmalıdır.

Not: parbox komutu içerisinde her komut kullanılamaz iken minipage komutunda neredeyse çoğu şey mümkündür ve bu yüzden oldukça kullanışlıdır.

Başlıklar ve Bölümler

Article (makale) sınıfı bir belgede kullanılabilecek başlık (section), alt başlık (subsection) ve alt alt başlık (subsubsection) komutları şunlardır:

\section{...}, \subsection{...} ve \subsubsection{...}

Kitap (book) ve rapor (report) sınıfı belgelere ise kullanılacak bölüm (chapter) komutu,

\chapter{...}

şeklindedir. Başlık ve bölüm isimlendirirken herhangi bir numara yazılmaz çünkü LaTeX numaralandırmayı otomatik olarak yapar. Makale sınıfı belgelerde başlık numaraları 1'den başlar ve diğer başlıklar ardışık olacak şekilde numaralandırılır. Kitap ve rapor sınıfı belgelerde ise ilk bölümün numarası 0'dır.

Article sınıfı belgelerde bölüm, letter(mektup) sınıfı belgelerde hem başlık hem de bölüm bulunmaz. Bundan dolayı article sınıfı bir belgeyi bir kitabın bölümü olarak ekleyebiliriz.

Başlık ve bölümlerin numaralandırılmamasını istiyorsak bu komutları aşağıdaki şekilde yazmalıyız:

\section*{...} \subsection*{...}
\subsubsection*{...} \chapter*{...}

Belge İçi Atıf

Bir belgenin içinde bulunan başlık, denklem, tablo, şekil, vb. atıfta bulunabilmek için atıfta bulunulmak istenen ifade \label{etiket} komutu ile etiketlenir. Burada etiket kısma ilgili ifade ile alakalı istenilen bir kısaltma yazılabilir.
Etiketlenen ifadeye atıf yapmak için \ref{etiket} ve \pageref{etiket} komutları kullanılır. Bunlardan ilki atıf verilecek ifadenin numarasını ikincisi ise bulunduğu sayfa numarasını yazdırır.
Matematiksel denklemlere verilen atıflarda denklem numarasın genellikle parantez içinde olması beklenir. Bu durumda denklem atıfının "(\ref{etiket})" yazılması gerekir. Fakat, \eqref{etiket} komutu direkt olarak parantez içerisinde atıf oluşturur.

İçindekiler Tablosu

Bir belgenin içindekiler tablosunu oluşturmak için
\tableofcontents
komutu kullanılır. Bu komut kullanıldığı yerde İçindekiler tablosunu oluşturur. Belgede yapılan güncellemelerin İçindekiler Tablosunda çıkması için bazen birden fazla derleme yapmak gerekebilir:

```
\documentclass[11pt,a4]{article}
\usepackage[turkish,shorthands=off]{babel}
\usepackage[T1]{fontenc}
\usepackage[latin5]{inputenc}
\usepackage[left=2cm,right=4cm,top=3cm,bottom=3cm]{geometry}
\usepackage{lipsum}
\title{Başlıklar ve İçindekiler Tablosu Hakkında}
\author{Onur Şahin}
\date{}
\begin{document}
\maketitle
\tableofcontents
\section{Giriş}
\subsection{Deneme 1}
\lipsum[1-3]
\subsection{Deneme 2}
\lipsum[3-5]
\section{Gelişme}
\subsection{Gelişmenin Denemesi}
\lipsum[5-6]
\end{document}
```

Yukarıdaki kaynak dosyasının derlenmesi sonucu aşağıdaki belge elde edilir.

Başlıklar ve İçindekiler Tablosu Hakkında

Onur Şahin

İçindekiler

1 Giriş **1**
1.1 Deneme 1 1
1.2 Deneme 2 2

2 Gelişme **2**
2.1 Gelişmenin Denemesi 2

1 Giriş

1.1 Deneme 1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Ortamlar (Environments) LaTeX'de bir grup öğeye etki eden komutlardan oluşan bir sistem olarak yorumlanabilir. Ortamlar,

\begin{ortam} ...\end{ortam}

şeklindedir. Burada "ortam" kısmına kullanılmak istenen ortamın adı, ... kısmına ise ortama yazılmak istenen ifadeler gelir. Ortamları içi içe yazmak mümkündür. Örneğin,

\begin{document} ...\end{document}

bir ortamdır ve aradaki yazılanları preamble 'da (sahanlıkta) yazılana göre derler. Yukarıda gördüğümüz: metinleri hizalamamız sağlayan

flushleft, flushright, center,

ifadeleri de birer ortamdır.

Şimdi birkaç önemli ortamı daha inceleyelim.

Numaralandırma ve Listeleme

Metin içerisinde belirli ifadeleri numaralandırmak için \begin{enumerate} ...\end{enumerate} ortamı kullanılır. Örneğin:

```
\begin{enumerate}
\item Kış Ayları
\begin{itemize}
\item Aralık
\item[-] Ocak
\end{itemize}
\item Yaz Ayları
\begin{itemize}
\item[a] Temmuz
\item[*] Ağustos
\end{itemize}
\end{enumerate}
```

1. Kış Ayları
   - Aralık
   - Ocak
2. Yaz Ayları
   a. Temmuz
   * Ağustos

Yukarıdaki örnekte de görüldüğü üzere **enumerate** ortamında numaralandırmak istediğimiz maddeleri \item ile yazarız ve bu maddeler otomatik olarak numaralandırılır. \itemize ortamında ise eğer item'a bir şey yazmazsak maddeler bir yuvarlak ile (•) başlar. Görülüğü üzere bu işaret [.] içine istenilen ifadenin yazılması ile değiştirilebilir.

Listeleme için ise

\begin{list}{belirteç}{} ...\end{list}

ortamı kullanılır. Bu ortamın kullanılması **enumerate** ortamı neredeyse aynıdır. Yine maddeler \item ile belirtilir. Bu ortamın farkı, **belirteç** yerine yazılacak her bir maddenin solunda belirmesidir. Yani **enumerate** gibi ardışık numaralandırma ya da **itemize** gibi tek tek ifade belirtmek yerine **belirteç** kısmına yazılacak ifade her maddenin başında olacaktır.

Alıntılama

Başka bir kaynaktan alıntı yapmak veya önemli cümleleri vermek için **quote** ortamı kullanılır. Alıntı yapmak için ilgili metin aşağıda verilen komutlar arasına girilir:

\begin{quote} ...\end{quote}

Örneğin:

```
Atatürk'ün bilime
verdiği değer
şu sözleri ile
anlaşılabilir:
\begin{quote}
Eğer bir gün benim
sözlerim bilimle ters
düşerse, bilimi seçin.
\end{quote}
```

Atatürk'ün bilime verdiği değer şu sözleri ile anlaşılabilir:

> *Eğer bir gün benim sözlerim bilimle ters düşerse, bilimi seçin.*

Özet (Abstract)

Bilimsel yayınların başında yayın hakkında kısa bir fikir vermek amacıyla özet (abstract) eklenir. Bunun için **abstract** aşağıda verilen ortam kullanılır:

\begin{abstract} ...\end{abstract}

Örneğin:

```
 \begin{abstract}
Okuyucuya makale
hakkında bir fikir
verecek bilgiler
bu ortama girilir.
\end{abstract}
```

Özet

Okuyucuya makale hakkında bir fikir verecek bilgiler bu ortama girilir.

Tablolar

LaTeX'de tablo hazırlamak için **tablo (tabular)** ortamı kullanılmaktadır. Bu ortamın komutu:

\begin{tabular}{özellikler} ...\end{tabular}

şeklindedir. Buradaki **özellikler** argümanı tablonun formatını ve sütunlarının nasıl hizalanacağını belirler. Tablo ortamının özelliklerini şöyle sıralayabiliriz:

- Sola yaslanan sütun için özellikler kısmında l, sağa yasalanan sütun için r ve ortalana sütun için ise c parametreleri kullanılır.
- Parametrelerin arasına veya dışına konulan | çizgisi sütunlar arasına veya tablo dışına düşey bir çizgi çizdirir.
- Tablonun sütunlarını ayırmak için "ampersand (&)" işareti kullanılır.
- Tabloda yeni bir satıra geçmek için biten satırın sonuna \\ komutu eklenir.
- Tabloda yatay çizgileri oluşturmak için \hline komutu kullanılır.

```
\begin{tabular}{l | r}
ln 1 & 0 \\
ln2 & 0.69
\end{tabular}
```

ln 1	0
ln2	0.69

```
\begin{tabular}{| r | c
|}
\hline
ln 1 & 0\\
\hline
ln 2 & 0.69
\hline
\end{tabular}
```

ln 1	0
ln 2	0.69

Yukarıdaki örneklerden görüleceği üzere: ilk tabloda ilk sütun sola ikinci sağa yaslanmış olup yalnızca iki sütun arasında bir düşey çizgi oluşturulmuştur. İkinci tabloda ise ilk sütun sağa yaslanmışken ikinci sütun ortalanmıştır. Bu tabloda satırlar ve sütunlar arasındaki çizgilerin yanı sıra tablonun dışına da bir çerçeve çizilmiştir.

Tüm bunlar **özellikler** argümanındaki parametrelerin yazılışı ve **hline** komutundan kaynaklıdır.

Bir tabloda yalnızca belli sütunların altında yatay çizgi oluşturmak istiyorsak \cline{i-j} komutunu kullanmalıyız. Burada **i.** sütundan başlayıp **j.**'na çizilen bir yatay çizgi oluşacaktır.

```
\begin{tabular}{r | c | c | c}
a & b & c & d\\
\cline{2-3}
a & 2& 3& 4\\
\end{tabular}
```

a	b	c	d
1	2	3	4

Tablonun bir hücresindeki metinin çok uzun olması durumunda LaTeX metni bölerek aşağı satıra geçmek yerine onun aynı satıra yazar. Çok uzun bir metnin bölünerek alt satıra yazılmasını sağlamak için \p{genişlik} komutu kullanılır. Burada **genişlik** kısmına uygun bir uzunluk birimi girilir.

```
\begin{tabular}{| p{4cm} | l |}
\hline
2019 yılında memur maaşlarına
yapılan zam oranı & %5.5\\
\hline
2020 yılında memur maaşlarına
yapılan zam oranı & %7.37\\
\hline
\end{tabular}
```

2019 yılında memur maaşlarına yapılan zam oranı	%5
2020 yılında memur maaşlarına yapılan zam oranı	%7.37

Bir tablonun sütunlarında dikey hizalama yapmak için array paketi ile birlikte: üst için t{boyut}, orta için m{boyut} ve alt için de b{boyut} komutları kullanılır.

@{...} komutu sütunlar arasına ne yazdırılacağını belirleyen bir komuttur. Eğer parantez içi boş bırakılırsa sütunlar arası boş kalır. Bu özellik sütun başındaki ve sonundaki boşlukları yok etmek için kullanılabilir. Örneğin:

```
\begin{tabular}{l}
\hline
Boşluklar var\\
\hline
\end{tabular}
```

Boşluklar var

```
\begin{tabular}{@{}l@{}}
\hline
Boşluklar yok\\
\hline
\end{tabular}
```

Boşluklar yok

Çok sütunlu satır oluşturmak için
\multicolumn{sayı}{konum}{metin}
komutu kullanılır. Bu komutta **sayı** kaç sütunu kaplayacağını, konum sütunun nasıl hizalanacağı ve metinde bu çoklu sütuna ne yazılacağını belirler. Örneğin:

```
\begin{tabular}{| l | c | c |}
\hline
\multicolumn{3}{| c |}{Ülke Listesi}\\
\hline
Ülke Adı & Para Birimi & Dili\\
\hline
Türkiye & Türk Lirası& Türkçe\\
\hline
Almanya & Euro & Almanca\\
\hline
Amerika & Dolar & İngilizce\\
\hline
\end{tabular}
```

Ülke Listesi		
Ülke Adı	Para Birimi	Dili
Türkiye	Türk Lirası	Türkçe
Almanya	Euro	Almanca
Amerika	Dolar	İngilizce

Çok sütunlu satır oluşturmaya benzer olarak çok satırlı sütun oluşturmak için \package{multirow} paketi ile birlikte

\multicolumn{sayı}{genişlik}{metin}

komutu kullanılır. Benzer şekilde **sayı** kısmı kaç satırı kaplayacağını, **genişlik** sütunun genişliğini ifade eder. Örneğin:

```
\begin{tabular}{| c | c | c |}
\hline
Sütun 1 & Sütun 2 & Sütun 3\\
\hline
\multirow{3}{2cm}{Çoklu satır} & hücre 2 & hücre 3\\
& hücre 5 & hücre 6\\
& hücre 8 & hücre 9\\
\hline
\end{tabular}
```

Sütun 1	Sütun 2	Sütun 3
Çoklu satır	hücre 2	hücre 3
	hücre 5	hücre 6
	hücre 8	hücre 9

Not: **tabular** ortamında oluşturulan tablolar aynı sayfada kalacak şekilde ayarlanır. Eğer bir sayfaya sığmayacak tablo oluşturmak istiyorsak \usepackage{longtable} paketi ile birlikte longtable ortamını kullanmalıyız.

Tabloları Özelleştirme

1. Çok sütunlu tablolarda birden fazla sütun *{sayı}{pozisyon} ifadesi ile oluşturulabilir. Örneğin *{9}{l} ifadesi ile 9 tane sola yaslanmış sütun oluşturulur. Örneğin:

```
\begin{tabular}{*{5}{l}}
\hline
Ülke Adı & Para Birimi & Dili & Nüfus & Başkent\\
\hline
Türkiye & Türk Lirası & Türkçe & 84 milyon &
Ankara\\
\hline
\end{tabular}
```

komutu bize 5 sütunlu sola yaslanmış bir tablo verir.

Ülke Adı	Para Birimi	Dili	Nüfus	Başkent
Türkiye	Türk Lirası	Türkçe	84 milyon	Ankara

2. Bir sütunun tamamında uygulanması için array paketi ile birlikte istenen komut, özellikler kısmında sütunun önüne >{\komut} şeklinde girilmelidir. Burada komutlar deklarasyon şeklinde olmalıdır. Yani sütundaki tüm elemanların kalın yazılması isteniyorsa >{\textbf} yerine >{\bfseries} yazılmalıdır. Örneğin:

```
\begin{tabular}{|>{\bfseries} l | c | c |}
\hline
Ülke Adı & Para Birimi & Dili \\
\hline
Türkiye & Türk Lirası & Türkçe\\
\hline
Almanya & Euro & Almanca\\
\hline
Amerika & Dolar & İngilizce\\
\hline
\end{tabular}
```

Ülke Adı	Para Birimi	Dili
Türkiye	Türk Lirası	Türkçe
Almanya	Euro	Almanca
Amerika	Dolar	İngilizce

elde edilir. Burada kullanılacak deklarasyonların tablosu aşağıdaki gibidir:

Komut	Deklarasyon	Açıklama
\emph	\em	Vurgulama
\textrm	\rmfamily	Roman font ailesi
\textsf	\sffamily	San-Serif font ailesi
\texttt	\ttfamily	Typewritter font ailesi
\textbf	\bfseries	Kalın(bold) karakter
\textit	\itshape	İtalik karakter
\textsl	\slshape	Eğik(slanted) karakter

3. Bir tablonun satırları değişken olacak şekilde renklendirmek için ilk olarak \usepackage[table]{xcolor} paketi yüklenmelidir. Daha sonra tablonun başına \rowcolors{sayı}{1. renk}{2. renk} komutu eklenerek satırları belirlenen renklerde olacak şekilde bir tablo elde edebiliriz. Burada sayı renklendirmenin kaçıncı satırdan başlayacağını, 1. renk tek numaralı satırların rengini ve 2. renk ise çift numaralı satırların rengini belirler. Örneğin:

```
 \rowcolors{2}{red}{white}
\begin{tabular}{| l | c | c |}
\hline
Ülke Adı & Dili & Para Birimi\\
\hline
Türkiye & Türk Lirası & Türkçe\\
\hline
Almanya & Euro & Almanca\\
\hline
Amerika & Dolar & İngilizce\\
\hline
İngiltere & Pound & İngilizce\\
\hline
Rusya & Ruble & Rusça
textbackslash\ \hline
\end{tabular}
```

komutu ile

Ülke Adı	Dili	Para Birimi
Türkiye	Türk Lirası	Türkçe
Almanya	Euro	Almanca
Amerika	Dolar	İngilizce
İngiltere	Pound	İngilizce
Rusya	Ruble	Rusça

tablosu elde edilir. Bu komutun yalnızca bir tabloya etki etmesi için tabular ortamı ile birlikte {...} arasına yazılmalıdır.

4. İstenilen sütunların renklerini değiştirmek için \cellcolor{renk} komutu sütunların konumunu özellikler argümanında istenilen sütunun önüne eklenmelidir. Eğer yalnızca bir hücrenin rengini değiştirmek istiyorsak bu komut o hücrenin başına eklenmelidir. Bu komutun kullanışı şu şekildedir:

```
\begin{tabular}{|>{\cellcolor{blue}} l | c | c
|>{\cellcolor{green}}c |}
\hline
Ülke Adı & Para Birimi & Dili & Bulunduğu
Kıta\\
\hline
Almanya & Euro & \cellcolor{yellow}Almanca &
Avrupa\\
\hline
Amerika & Dolar & İngilizce & Amerika\\
\hline
\end{tabular}
```

komutu ile

Ülke Adı	Para Birimi	Dili	Bulunduğu Kıta
Almanya	Euro	Almanca	Avrupa
Amerika	Dolar	İngilizce	Amerika

tablosu elde edilir.

5. Bir tablodaki çizgilerin rengini değiştirmek istiyorsak \arrayrulecolor{renk} komutu \begin{tabular} komutunda sonra yazılmalıdır. Örneğin:

```
\begin{tabular}{| l | c | c |}
\arrayrulecolor{red} \hline
Ülke Adı & Para Birimi & Dili \\
\hline
Almanya & Euro & Almanca \\
\hline
\end{tabular}
```

komutu ile

Ülke Adı	Para Birimi	Dili
Almanya	Euro	Almanca

tablosu elde edilir.

6. Bir tablodaki çizgilerin kalınlığını ayarlayabilmek için \renewcommand{\arrayrulewidth}{punto} komutu kullanılır. Bu komutun kullanışı şu şekildedir:

```
{\renewcommand{\arrayrulewidth}{2pt}
\begin{tabular}{| l | c | c |}
\hline
Ülke Adı & Para Birimi & Dili \\
\hline
Almanya & Euro & Almanca \\
\hline
\end{tabular}
}
```

komutu ile

Ülke Adı	Para Birimi	Dili
Almanya	Euro	Almanca

tablosu elde edilir.

Yüzer Nesneler (Floats)

Hazırladığımız bir dokümana bir tablo veya şekil eklemek istediğimizde bunlar genellikle kaynak dosyada eklediğimiz konumda belirir. Fakat tablo veya şekil eğer eklediğimiz kısımda sayfaya sığmıyorsa iki sayfaya bölünemediğinden istenmeyen bir sonuç elde edilmiş olur. Bu nesneleri yeni bir sayfaya eklemek bir çözüm gibi görünse bile bu durumda da pek çok yarı boş sayfası olan bir doküman elde edilmiş olur.

Yukarıda bahsedilen sorunun çözümü eklenen nesneleri kayar hale getirmektir. Böylece bir sayfadaki yerine sığmayan tablo veya şekil bir sonraki sayfaya eklenip onun boş bırakacağı yere ise metin eklenecektir.

Kayar nesneler için, biri şekiller biri de tablolar olmak üzere iki ortamımız bulunmaktadır. Şimdi bu ortamların nasıl çalıştığına bakalım.

Kayar nesneler için şekiller ve tablolar ortamı sırasıyla

```
\begin{figure}[konum]          \begin{table}[konum]
...                            ...
\end{figure}[konum]            \end{table}
```

komutlarıyla oluşturulur. Burada konum parametresi ile tablonun dokümanda yerleştirileceği yer belirlenmektedir. Buraya gelebilecek değerler ve karşılıkları aşağıdaki tabloda verilmiştir:

Seçenek	Açıklama
h	Tablonun oluşturulduğu yere. Genellikle küçük yüzer nesneler içindir
t	Sayfanın üst tarafına
b	Sayfanın alt tarafına
p	Sadece yüzer nesnelerin olduğu ayrı bir sayfaya
!	Mutlaka diğer seçeneklerden biri

Bu parametreler şu şekilde kullanılır. Örneğin [ht] parametresi, tablonun oluşturulduğu yere, eğer bu mümkün değilse sayfanın üstüne yerleştirilmesini sağlar. Bu iki seçenek mümkün değil ise LaTeX tabloyu diğer seçeneklerden uygun olanı seçerek yerleştirir. Eğer parametreyi [!ht] şeklinde kullanırsak bu durumda LaTeX bu iki seçenekten birini uygulamaya zorlar. Eğer herhangi bir komut parametresi girilmez ise varsayılan olarak [tbp] olarak uygulanır.

Konum parametresi olarak tek bir seçenek kullanması sorun yaratabilir. [h] seçeneğinin tek kullanılması yüzer nesnelerin yerleştirilmesinde sorun çıkarabilir. Bu yüzden bu seçeneğin tek kullanılmaması daha uygundur. LaTeX'in son sürümlerinde bu seçenek otomatik olarak [ht] olarak işlem görür.

Yüzer nesneler de kullanılacak ek komutlar şu şekildedir:

\caption{...} komutu ile yüzer nesnelere açıklama konulmaktadır. Bu komut \end{tabular} komutundan sonra konursa açıklama tablonun altında, eğer \begin{tabular} komutundan önce konulursa açıklama tablonun üstünde olur.

\label{...} komutu ile yüzer nesnelere doküman içinde atıfta bulunabilmek için etiket oluşturulur.

Eğer tablonun ortalanmasını istiyorsak \begin{tabular} komutundan önce \centering komutu yazılmalıdır.

\listoftables ve \listoffigures komutları \tableofcontents komutu gibi çalışarak sırasıyla Tablo ve Şekiller listesi çıkarmaktadır. Bu komutlar kullanıldıkları yerde bu listeleri türetir. Bu listelerde tablonun veya şeklin altyazısı olduğu yazdırılır. Bu yüzden eğer altyazı uzun ise daha kısa kullanılacak bir altyazı \caption komutunda köşeli parantez ile verilebilir. Yani komut,

\caption[kısa altyazı]{uzun altyazı}

şeklinde girilirse kısa altyazı kısmına yazılan ifade Tablolar veya Şekiller listesinde, uzun altyazı kısmına yazılan ifade ise yüzer nesnelerin altında belirir. Şimdi Tablo ortamına dair bir örnek verelim:

```
 \begin{table}[!ht]
\begin{tabular}{| l | c | c |}
\hline
Ülke Adı & Para Birimi & Dili\\
\hline
Türkiye & Türk Lirası & Türkçe\\
\hline
Almanya & Euro & Almanca\\
\hline
Amerika & Dolar & İngilizce\\
\hline
İngiltere& Pound & İngilizce\\
\hline
\end{tabular}
\caption{Ülke Bilgileri}
\label{ülkeler}
\end{table}
```

Yukarıdaki komut ile aşağıdaki tablo elde edilir:

Ülke Adı	Para Birimi	Dili
Türkiye	Türk Lirası	Türkçe
Almanya	Euro	Almanca
Amerika	Dolar	İngilizce
İngiltere	Pound	İngilizce

Tablo 1: Ülke Bilgileri

Bu tabloda görüldüğü üzere alt yazı bilgisi caption ile belirtilmiştir. İleride bu tabloya atıf verilmek istenirse \ref{ülkeler} komutunun kullanılması yeterlidir.

Burada değinilecek bir diğer nokta caption olarak üretilen Tablo adını ve bunun sitilini istediğimiz şekilde değiştirebileceğimizdir. Bunun için \usepackage{caption} paketi kullanmak gerekir. Bu paket yüklendikten sonra:

\captionsetup[ortam adı]{name={İstenilen ad},labelfont={font adı}, textfont={font adı} labelsep=isim ayracı}
komutu ile bir yüzer nesnenin açıklama başlığının formatı ayarlanabilir. ortam adı kısmına hangi yüzer nesnenin adı gelir ise onun başlık ayarları değişecektir. Burada name={İstenilen ad} ile başlığın adını değiştirmek istediğimiz ifade yazılır. labelfont={font adı} ve textfont={font adı} ile sırasıyla başlığın ve açıklamanın fontunun nasıl yazdırılacağı belirlenir. Buralara bf, sc, it gibi font sitilini belirten font adları, small, Large, Huge gibi büyüklük belirten ifadeler ya da color={renk} ifadesinde renk kısmına blue, red, green gibi renk belirten ifadeler gelebilir (renk için xcolor paketinin yüklenmesi gereklidir). Son olarak labelsep=isim ayracı ifadesi ile başlık adı ve ondan sonra gelecek yüzer nesnenin numarasının nasıl ayrılacağı belirlenebilir. Buraya empty, colon, period, space gibi ifadeler gelebilir. Daha fazla ayrıntı için caption paketi incelenebilir.

Vurgulayacağımız bir diğer husus başlık numaralarının nasıl değiştirilebileceğidir. Standart ayarlarda nesne başlıkları ardışık olarak numaralandırılmaktadır. Eğer bunu bölüm ya da başlık numaralarına göre numaralandırmak istiyorsak:

\usepackage{amsmath} paketi ile birlikte \numberwithin komutunu kullanmalıyız. Bu komut şu şekilde kullanılır: \numberwithin{ortam adı}{neye göre numaralandırılacağı}

Burada hangi yüzer nesnenin numaralandırmasını değiştirmek istiyorsak ortam adı kısmına onun adı yazılacaktır. Nesnenin neye göre numaralandırılacağı ise ikinci parantez içine yazılır. Örneğin başlığa (section) göre numaralandırmak için buraya section yazmalıyız.

Bir dokümana bir şeklin nasıl ekleneceğini daha sonraki kısımlarda ayrıntılı olarak incelenecektir.

Matematiksel İfadelerin Yazılması

LaTeX'in en büyük avantajlarından biri şüphesiz ki matematiksel ifadelerin bir dokümana dizilmesinde ortaya çıkmaktadır. Diğer yazım programlarına kıyasla LaTeX matematiksel ifadelerin yazımını oldukça kolay hale getirmekle birlikte bunların çıktısının da daha profesyonel olması sağlamaktadır. Bu bölümde bir matematiksel formülün nasıl yazılacağını, matematiksel formüller için kullanılan ortamları, matematiksel sembolleri, matematiksel fontlar vb. konuları sırasıyla göreceğiz.

LaTeX ile matematiksel ifadeleri yazdırmak için iki yol vardır. Bunlardan ilki matematiksel ifadelerin satır içinde, ikincisi ise yeni bir satırda yazılmasıdır.

Bir metinde satır içine matematiksel bir ifade yazdırmak istiyorsak ya bu ifadeyi `\begin{math}...\end{math}` arasına ya da `$...$` arasına yazmalıyız. Örneğin:

```
Dik kenarlarının
uzunlukları
\begin{math}a\end{math}
ve
\begin{math}b\end{math}
olan bir üçgenin
uzun kenarının
karesinin uzunluğu
\begin{math}a^2+b^2
\end{math} toplamına
eşittir.
```

Dik kenarlarının uzunlukları a ve b olan bir üçgenin uzun kenarının karesinin uzunluğu $a^2 + b^2$ toplamına eşittir.

Yukarıdaki örnekten de görüldüğü üzere bir satır içinde matematiksel ifade yazmak için sürekli math ortamını kullanmak oldukça yorucu bir iştir. Bu yüzden bu gibi durumlarda aynı işlemi gören $...$'i kullanmak daha uygun olur. Yukarıdaki örneği $ işaretini kullanarak yazarsak görüleceği üzere aynı çıktıyı alırız:

```
Dik kenarlarının uzunlukları
$a$ ve $b$ olan bir üçgenin
uzun kenarının karesinin
uzunluğu $a^2+b^2$ toplamına
eşittir.
```

Dik kenarlarının uzunlukları a ve b olan bir üçgenin uzun kenarının karesinin uzunluğu $a^2 + b^2$ toplamına eşittir.

LaTeX'de math ortamı ile ilgili şunları söyleyebiliriz:

- LaTeX, math ortamında boşluklar dikkate almaz. Boşluk bırakmak için\, \,, \:, \:, \quad ve \qquad komutlarından biri kullanılır. Benzer şekilde \\ komutu da işlevsizdir.
- math ortamında sitil komutları şu şekildedir. \displaystyle, \textstyle, \scriptystyle ve \scriptscriptstyle.
- LaTeX, math ortamında satır içinde yazılan matematiksel ifadeyi bir satıra sığdırabilmek için \textstyle sitilinde yazar. Eğer bir matematiksel ifadeyi \displaystyle sitilde yazdırmak istiyorsak denklemin başına \displaystyle yazmak yeterlidir. Eğer bir dokümandaki tüm matematiksel ifadeleri \displaystyle sitilde yazdırmak istiyorsak preamble'a \everymath{\displaystyle} yazılmalıdır.

LaTeX'de daha uzun matematiksel formülleri ya da denklemleri ayrı bir satırda yazdırmak istiyorsak bu ifadeleri

\begin{displaymath}...\end{displaymath}

arasına ya da \[...\] arasına yazmalıyız. displaymath ortamı ya da aynı işlevi gören \[...\] komutu denklemleri numaralandırmaz dolayısıyla bu denklemlere atıf vermemiz mümkün değildir. Ayrıca bu komutlar ile denklemlerin alt alta dizilmesi de mümkün değildir. Bu yüzden matematikse ifadelerin yazılmasında kullanılacak ve oldukça kullanışlı olan bazı ortamlara ilerleyen kısımlarda değineceğiz. Şimdi ilk olarak displaymath ortamının nasıl kullanıldığına dair bir örnek verelim:

```
Fermat'ın son teoremi şu şekildedir. $n$
ikiden büyük tamsayısı için
\begin{displaymath}
a^n+b^n=c^n
\end{displaymath}
denklemini sağlayan $a$, $b$ ve $c$ pozitif
tamsayıları bulunamaz.
```

Fermat'ın son teoremi şu şekildedir. n ikiden büyük tamsayısı için

$$a^n + b^n = c^n$$

denklemini sağlayan a, b ve c pozitif tamsayıları bulunamaz.

Yukarıdaki örnekten de görülebileceği üzere displaymath ortamı matematiksel ifadeyi yeni bir satırda vermektedir. Yukarıdaki denklemi \[\] ile yazdırmak sonucu değiştirmeyecek, yine aynı çıktıyı elde edecektik.

Burada displaymath ortamı ile ilgili vurgulamamız gereken birkaç husus vardır. Aşağıda verilecek tüm özellikler ayrıca ileride vereceğimiz diğer matematik ortamlarında da geçerlidir:

- math ortamı için yukarıda verilmiş olan tüm özellikler displaymath ortamı içinde geçerlidir.
- displaymath ortamında boş bir satır bırakılamaz. Her bir matematiksel ifade bir satır işgal eder. Eğer boş bir satır bırakılırsa kaynak dosya derlendiğinde hata verir.

- displaymath ortamı, bu ortamda yazılan denklemden önce ve sonra bir boşluk bırakır. Bu yüzden bu ortamda bir denklem yazıldığından fazladan bir boşluk bırakmaya gerek yoktur. Aksi taktirde denklemlerden önce ve sonra gereksiz boşluklar meydana gelir.
- Matematik formatında her bir karakter bir matematiksel değişken olarak kabul edilir. Eğer matematik formatında düz bir metin yazılmak isteniyorsa, bu metin \textrm{...} komutuyla yazılmalıdır.
- Matematik formatında bir denklemin bazı yerlerini düz harfler ile yazmak isteniyorsa \textrm{...} komutu kullanılamaz. Çünkü bu komut ile metin formatına geçildiği için bazı matematiksel ifadelerin yazımını gerçekleştiremez. Bu yüzden bu gibi durumlarda \mathrm{...} komutu kullanılır. Matematik formatında kullanılabilecek yazı tipleri aşağıdaki tabloda verilmiştir:

Komut	Çıktı	Gerekli Paket
\textrm{ABCabc123}	ABCabc123	
\text{ABCabc123}	ABCabc123	amstext veya amsmath
\textnormal{ABCabc123}	ABCabc123	
\mbox{ABCabc123}	ABCabc123	
\mathrm{ABCabc123}	ABCabc123	
\mathit{ ABCabc123}	$ABCEabc123$	
\mathnormal{ ABCabc123}	$ABCEabc123$	
\mathcal{ ABCabc123}	$\mathcal{ABCE}\dashv\lfloor\rfloor\infty\in\ni$	
\mathscr{ ABCabc123}	$\mathscr{ABCE}$	mathrsfs
\mathfrak{ ABCabc123}	$\mathfrak{ABCEabc123}$	amsfonts veya amssymb
\mathbb{ ABCabc123}	$\mathbb{ABCE}\nexists\nvDash\nVdash$	amsfonts veya amssymb

Temel Matematik Komutları

LaTeX'de her matematiksel ifadeye karşılık bir komut bulunmaktadır. Dolayısıyla bir matematiksel ifadenin doğru bir şekilde derlenebilmesi için ona karşılık gelen komutun doğru yazılması gerekmektedir. Günümüzde pek çok editör programı çoğu matematik komutunu kısa yol olarak içerisinde barındırmaktadır. Fakat, biz yine de bazı temel komutları vereceğiz. Unutulmamalıdır ki burada bahsedemediğimiz daha pek çok komut olacaktır.

Üsler ve İndisler:

Bir matematiksel ifadede üs için ^ komutu, indis için ise _ komutu kullanılır. Örneğin:

```
$a^2$, \quad $x^{10}$,
\quad $y_5$ \quad
$b_{n_k}$
```

a^2, x^{10}, y_5, b_{n_k}

Yukarıdaki örnekten de görüleceği üzere eğer üsse veya indise birden fazla ifade yazacaksak bunları küme parantezi içinde yazmalıyız aksi takdirde yalnızca ilk ifade üs ya da indis olarak belirecektir.

Kesirler:

Kesirler \frac{...}{...} komutu ile yazdırılır. Burada ilk parantez içine yazılan ifade pay, ikinci paranteze yazılan ifade payda olacaktır. Örneğin:

```
$\frac{5}{3}$, \quad
$\frac{x^2}{x^2+y^2}$,
\quad $x^{\frac{1}{2}}$
```

$\frac{5}{3}$, $\frac{x^2}{x^2+y^2}$, $x^{\frac{1}{2}}$

Karekök:

Karekök \sqrt{...} yazılır. Eğer n. kök yazmak istiyorsak \sqrt[n]{...} komutu kullanılır. LaTeX karekökün boyunu otomatik olarak ayarlamaktadır. Eğer sadece kök işareti yazılmak isteniyorsa \surd{...} komutu kullanılır. Örneğin:

```
$\sqrt{x^2+y^2}$, \quad
$\sqrt[4]{(x+y)^3}$, \quad
${\surd{x}}$
```

$\sqrt{x^2+y^2}$, $\sqrt[4]{(x+y)^3}$, $\surd{x}$

Vektörler:

LaTeX'de vektörü yazdırabilmek için yani bir ifadenin üzerine bir ok eklemek için \vec komutu kullanılır. Eğer A noktasından B noktasına bir vektörü yazmak istiyorsak ise \overrightarrow{...} komutu kullanılmalıdır. Örneğin:

```
$\vec v$, \quad
$\overrightarrow{AB} $
```

$\vec{v}$, $\overrightarrow{AB}$

İki vektörün noktasal çarpımında "nokta" işaretini yapmak için \cdot kullanılır. Örneğin:

```
$ w=\vec v \cdot u $
```

$w = \vec{v} \cdot \vec{u}$

Matematiksel ifadelerde üç nokta koymak için \ldots, satırın aşağısında, ve \cdots, satırın ortasında, komutları kullanılır. Örneğin:

```
$a_0,a_1\ldots,a_n$ ve
$a_0,a_1\cdots,a_n$
```

$a_0, a_1, \ldots, a_n$ ve
$a_0, a_1, \cdots, a_n$

Ayrıca dikey ve çapraz noktalar için \vdots ve \ddots komutları kullanılır.

Bir ifadenin üstüne veya altına çizgi çekmek için sırasıyla \overline{...} veya \underline{...} komutları kullanılır. Örneğin:

```
$\overline{x+y}$, \quad
$\underline{x+y}$
```

$\overline{x+y}$, $\underline{x+y}$

Bir ifadenin üstüne veya altına küme parantezi eklemek için sırasıyla \overbrace{...} veya \underbrace{...} komutları kullanılır. Örneğin:

```
$\overbrace{x+y+z}^3$,
\quad $\underbrace{x+y}_5$
```

$\overbrace{x+y+z}^{3}$, $\underbrace{a+b+c+d}_{4}$

Yukarıda verilen komutlara benzer şekilde pek çok daha komut mevcuttur. Bunlardan birkaçı ve çıktısı aşağıda verilmiştir:

```
$\widetilde{abc}$, \quad
$\widehat{abc}$, \quad $\hat{a}$,
\quad $\tilde{a}$, \quad $\bar{a}$,
\quad $\dot{a}$, \quad $\ddot{a}$
```

$\widetilde{abc}$, $\widehat{abc}$, $\hat{a}$, $\tilde{a}$, $\bar{a}$, $\dot{a}$, $\ddot{a}$

Temel Fonksiyonlar:

Matematik modunda her fonksiyon için ayrı bir komut bulunmaktadır. Eğer bir fonksiyonu komut kullanmadan yazarsak LaTeX fonksiyon yazarken kullanılan karakterleri birer değişkenmiş gibi yazıdırır. Bazı temel fonksiyonların komutları aşağıdaki gibidir:

```
\cos, \sin, \tan, \cot, \sec, \csc,
\cosh, \sinh, \tanh, \coth, \exp,
\log, \ln, \inf, \sup, \min, \max,
\lim, \liminf, \limsup
```

Örneğin:

```
$\sin^2 x+ \cos^2 x=1 $,
$ e^{i\pi}=-1$
```

$$\sin^2 x + \cos^2 x = 1,$$
$$e^{i\pi} = -1$$

Limit, İntegral, Toplam ve Çarpım:

Matematik formatında limit yazdırmak için \lim_{x\rightarrow y} komutu kullanılır. Örneğin:

```
$\lim_{x\rightarrow 0
\frac{\sin x}{x}=1$,
```

$$\lim_{x\to 1} \frac{\sin x}{x} = 1$$

Not: Yukarıdaki örnekten görülmektedir ki denklemi $...$ arasına yazdığımız için yani math ortamında yazdığımız için denklem \textstyle formatında yazıldı. Dolayısıyla $x \to 1$ ifadesi limitin altına olacak şekilde yazdırılmadı. Eğer $x \to 1$'in limitin altında olacak şekilde yazdırmak istiyorsak denklemden hemen önce \displaystyle yazmamız yeterlidir. Ayrıca burada \rightarrow komutu yerine \to komutu da kullanılabilir.

Matematik ortamında integral işareti \int, toplam işareti \sum ve çarpım işareti \prod komutları ile yazdırılır. Bu komutlarda alt ve üst indisler sırasıyla _{...} ve ^{...} komutları ile yapılmaktadır. Örneğin:

```
$\displaytsyle\int_{x=0}^{\infty}
e^{-x^2}d x=\pi/2$,
```

$$\int_0^\infty e^{-x^2}\,dx = \sqrt{\pi}/2 \text{ xx"x"}$$

```
$\displaytsyle\sum_{n=1}^{\infty}
\frac{1}{n^2}=\frac{\pi^2}{6}$,
```

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

```
$\displaytsyle\prod_{n=1}^{3}n=6$,
```

$$\prod_{n=1}^{3} n = 6$$

Karmaşık indisler için amsmath paketi ile birlikte \substack ve \subarray komutu kullanılır. Örneğin:

```
$\displaytsyle\sum_{\substack{0<i<n
\\1<j<m}}P(i,j), \displaytsyle
\sum_{\begin {subarray}{l}i \in I
\\1<j<m\end {subarray}}Q(i,j)$,
```

$$\sum_{\substack{1<i<n \\ 1<j<m}} P(i,j), \sum_{\substack{i\in I \\ 1<j<m}} Q(i,j)$$

Türev:

Türev yazarken \frac komutu ile birlikte kısmi türev için \partial ve adi türev için ise d ifadeleri kullanılır. Örneğin:

```
$\frac{\partial^2u}{\partial x
\partial x}$, $\frac{\d y}{d x
}$
```

$\frac{\partial^2 u}{\partial x \partial y}$, $\frac{dy}{dx}$

Ayrıca normal türevi göstermek için ' işareti ya da \prime komutu kullanılır. Örneğin:

```
$y''', y'', y\prime$
```

$y''', y'', y\prime$

Binom:

Binom ifadesini yazdırmak için amsmath paketi ile birlikte \binom komutu kullanılır. Örneğin:

```
$\displaytsyle\binom{n}{k}$,
```

$\displaystyle\binom{n}{k}$,

Gruplandırıcı İşaretler:

LaTeX'de gruplandırıcı işaretler olan {, [, (, gibi işaretlerin kullanımına dikkat edilmelidir. Matematik formatında açılan her gruplandırıcı işaretin mutlaka kapatılması gerekir. Aksi halde kaynak dosyanın derlenmesi sırasında hata meydana gelir. Bir gruplandırıcı işaretin içine yazılan denklemler bazen bu işaretlere kıyasla oldukça büyük olabileceğinden işaretlerinde bu denklemleri kapsayacak boyutta olması gerekmektedir. Bunun birkaç yöntemi vardır.

i) İlk olarak gruplandırıcı işaretin \left komutu ile başlatıp \right komutu ile bitirmek. Bu durumda LaTeX denklemin boyutuna göre parantezlerin boyutunu otomatik olarak ayarlamaktadır. Eğer \left ile açılmış bir parantezi kapatmak istemiyorsak \right . komutunu kullanmalıyız. Aksi taktirde derleme sırasında hata ile karşılaşırız. Örneğin:

```
\[
\left( \frac{x}{x+y} \right)^2
\]
```

$$\left(\frac{x}{x+y} \right)^2$$

ii) Eğer bir satıra sığmayan bir denklemi $\{\}$ parantezleri arasında yazdırmak için ilk satırda \left\{ ile parantez açıp diğer satır sonunda \right\} ile parantezi kapatarak derlersek hata mesajı alırız. Bu gibi birde fazla satırdan oluşan denklemler denklemi $\{\}$ arasında yazdırabilmek için denklemin başında \left\{ komutu ile bir parantez açıp o satır sonunda \right. komutu, alt satır başında \left. komutu ve denklem sonunda \right\} komutunu kullanmalıyız.

iii) Eğer gruplandırıcı işaretlerin boyutunu elle ayarlamak istiyorsak, bu işaretlerden önce \big, \Big, \bigg veya \Bigg komutlarından birini kullanabiliriz. Örneğin:

```
\big{, \Big{, \bigg{, \Bigg{
```

$\big\{, \Big\{, \bigg\{, \Bigg\{$

Matrisler ve Parçalı Fonksiyonlar:

Matematik formatında matris ya da parçalı fonksiyon yazdırabilmek için array ortamı kullanılmalı. array ortamı aslında yapı itibarı ile tabular ortamı gibi satır ve sütundan oluşan girdileri dizmektedir. Yani \begin{array}{özellikler}...\end{array} şeklinde yazılan ortamda "özellikler" kısmı tabloda olduğu gibi sütunların nasıl dizileceğini belirtmektedir. Örneğin array ortamı kullanılarak bir matris aşağıdaki gibi yazdırılabilir:

```
\[ \left(
\begin{array}{rrr}
-1 & 0 & 3 \\
4 & 2 & -3 \\
5 & - 4 & 2
\end{array}
\right)
\]
```

$$\left(\begin{array}{rrr} -1 & 0 & 3 \\ 4 & 2 & -3 \\ 5 & -4 & 2 \end{array} \right)$$

Yukarıdaki matris örneğinden de görülmektedir ki matris girdilerini parantez içerisinde yazdırabilmek için array ortamını daha önceden de gördüğümüz şekilde \left(...\right) arasında yazdık.

Parçalı tanımlı fonksiyonlarda matrislerin yazımına benzer olarak array ortamı ile yazdırılabilmektedir. Örneğin:

```
\[|x|= \left\{
\begin{array}{rrc}
x & , & x \ge 0 \textrm{
ise} \\
-x & , & x < 0 \textrm{
ise} \\
\end{array}
\right.
\]
```

$$|x| = \left\{ \begin{array}{rrc} x & , & x \ge 0 \text{ ise} \\ -x & , & x < 0 \text{ ise} \end{array} \right.$$

Yukarıdaki örnekten de görülebileceği üzere \left\{ ile açılan parantez kapatılmadığından son komut \right.'dır. Parçalı tanımlı fonksiyonlar amsmath paketi ile birlikte cases ortamı ile de yazılabilir.

Equation Ortamı

Matematiksel ifadeleri yazmak için kullanılan ortamlardan biridir. displaymath ortamından farklı olarak yazılan denklemler numaralandırılır. Matematiksel ifadeler \begin{equation}...\end{equation} arasına yazılır. Denklemleri etiketlemek için \label komutu kullanılır. Önceki kısımlarda bahsettiğimiz gibi etiketlenen bir denkleme \ref{label} komutu ile göndermede bulunulabilir. amsmath paketi ile birlikte gelen \eqref{label} komutu denklem numarasını direkt olarak () içerisinde yazdırır. Eğer \equation ortamında yazılan bir denklemin numaralandırılmamasını istiyorsak ya \begin{equation*}...\end{equation*} komutu kullanılmalı ya da denklemin sonuna \notag veya \nonumber komutları yazılmalıdır. Örneğin:

```
\begin{equation}
\lim_{x \to
\infty}\frac{1}{x}=0
\end{equation}
```

$$\lim_{x \to \infty} \frac{1}{x} = 0 \qquad (1)$$

```
\begin{equation*}
\int_{0}^{\infty}
e^{-x^2}dx
=\frac{\sqrt{\pi}}{2}
\end{equation*}
```

$$\int_0^{\infty} e^{-x^2}\,dx = \frac{\sqrt{\pi}}{2}$$

Eğer denklem numaralarını sağda değil de solda olmasını istiyorsak documentclass ifadesine opsiyon olarak leqno eklemek yeterlidir. Yani \documentclass[leqno]{article} şeklinde yazılmalıdır.

Bir denkleme istediğimiz numarayı ya da karakteri atamak istiyorsak amsmath paketi ile birlikte \tag{·} komutu kullanılır. Örneğin:

```
\begin{equation}\tag{$\star$}
\lim_{x \to
\infty}\frac{1}{ x}=0
\end{equation}
```

$$\lim_{x\to\infty}\frac{1}{x}=0 \qquad (\star)$$

Eğer tag komutu ile atadığımız karakter ya da numaranın parantezsiz olarak yazdırılmasını istiyorsak tag* komutunu kullanmalıyız.

Denklemlerin numaralandırılmasını ardışık değil de section'a göre yapılmasını istiyorsak table ortamındakine benzer olarak amsmath paketi ile birlikte \numberwithin komutunu kullanmalıyız. Bu komut şu şekilde kullanılır: \numberwithin{equation}{section}. Eğer numaralandırmayı subsection'a göre yapmak istersek komutta section yerine subsection yazmamız yeterlidir.

Eqnarray Ortamı

Birden fazla satırdan oluşan matematiksel ifadeleri yazdırmak için eqnarray ortamı kullanılır. eqnarray ortamında yazılan denklemler equation ortamında olduğu gibi numaralandırılır. Numarasız denklemler elde etmek için eqnarray* ortamı kullanılır.

eqnarray ortamında denklemler aslında ilk sütunu sağa yaslı, ikinci sütunu ortalı ve son sütunu sola yaslı olan 3 sütunlu bir tablo gibi dizilir. Burada ilk sütun denklemin sol tarafı ikinci sütun $=$ işareti ve son sütun ise denklemin sağ tarafı gibi düşünülebilir. Burada sütunlar, sütun ayracı olan & işareti ile ayrılmaktadır. Alt satırdaki denkleme geçmek için denklem sonuna \\ yazılır. Örneğin:

```
\begin{eqnarray}
\cos 2x&=&\cos^2 x-\sin^2 x\\
&=&2\cos^2 x -1\\
&=&1-2\sin^2 x
\end{eqnarray}
```

$$\begin{aligned}
\cos 2x &= \cos^2 x - \sin^2 x && (2)\\
&= 2\cos^2 x - 1 && (3)\\
&= 1 - 2\sin^2 x && (4)
\end{aligned}$$

Yukarıdaki örnekten de görüleceği üzere eqnarray ortamında yazılan her satırdaki denklem numaralandırılmaktadır. Eğer bir satırdaki denklemin numaralandırılmamasını istiyorsak o satırın sonuna \nonumber \notag yazmalıyız.

Yine yukarıdaki örnekten görülmektedir ki ='in iki yanında fazla boşluk bulunmaktadır. Bu boşlukları azaltmak için denklemin başına \setlength \arraycolsep{2pt} komutu eklenmelidir. Örneğin:

```
{\setlength \arraycolsep{2pt}
\begin{eqnarray*}
\cos 2x&=&\cos^2 x-\sin^2 x\\
&=&2\cos x^2 x-1\\
&=&1-2\sin x^2 x
\end{eqnarray*}}
```

$$\begin{aligned}\cos 2x &= \cos^2 x - \sin^2 x \\ &= 2\cos^2 x - 1 \\ &= 1 - 2\sin^2 x\end{aligned}$$

Burada \arraycolsep{2pt} komutu içindeki değer küçültüldükçe boşluklar azalmaktadır.

Eğer eqnarray ortamında yazılan denklemde eşitliğin sol tarafındaki denklem çok uzun ise bu durumda ilk satıra yazılan denklemler o satıra sığmayabilir. Bu duruma denklemlerin nereden bölüneceğine biz müdahale edebiliriz. Fakat bunun için ayrıca \lefteqn{...} komutu bulunmaktadır. Denklemdeki uzun satır bu komutun içine yazılarak onun bir satırda kalması sağlanır. Örneğin:

```
\begin{eqnarray*}
\lefteqn{x^5+5x^4+10x^3+10x^2+5x+1}\\
&=&(x+1)^5
\end{eqnarray*}
```

$$\begin{aligned} x^5+5x^4+10x^3+10x^2+5x+1 \\ = \quad (x+1)^5 \end{aligned}$$

Align Ortamı

Birden fazla satırdan oluşan matematiksel ifadelerin dizilmesi için amsmath paketi ile gelen pek çok ortam bulunmaktadır (align, falign, gather, multiline, split). Şimdi align ortamını inceleyelim.

align ortamı ile eqnarray ortamı arasındaki farkları şu şekilde sıralayabiliriz:

- eqnarray ortamında iki hizalama noktası (&) varken align ortamında tek hizalama noktası vardır.
- align ortamında, eqnarray ortamındaki denklemleri alt alta dizerken $=$ öncesinde ve sonrasında meydana gelen fazla boşluklar meydana gelmez.
- eqnarray ortamında yazılan denklemler yeni sayfaya bölünürken, align ortamını buna izin vermez.

Yukarıda listelenen özellikler align ortamının avantajları gibi göründüğünden, alt alta dizili denklemlerin yazımında align ortamının kullanılmasını tavsiye edilmektedir. Fakat align ortamının kullanımı için amsmath paketinin yüklenmesi gerektiğinden, bu paketin yüklenemeyeceği durumlarda eqnarray ortamının kullanılması daha uygun olacaktır.

Şimdi align ortamının nasıl çalıştığını inceleyelim.

```
\begin{align*}
(x+y)^3&= (x+y)(x+y)^2\\
&=(x+y)^2+2xy+y^2)\\
&=x^3+3x^2y+3xy^2+y^3)
\end{align*}
```

$$\begin{aligned}(x+y)^3 &= (x+y)(x+y)^2\\ &= (x+y)(x^2+2xy+y^2)\\ &= x^3+3x^2y+3xy^2+x^3\end{aligned}$$

Bu örnekten de görüldüğü üzere yukarıda bahsedildiği gibi align ortamında yalnızca tek bir bir hizalama noktası kullanılmıştır. Burada denklem $(x+y)^3$'den sonra hizalanmıştır.

Bu ortamda denklemleri numarasız yazmak istiyorsak align* ortamını kullanmalıyız.

Bir satırda birden fazla olan eşitlikler de alt alta align ortamı ile sıralanabilir. Örneğin:

```
\begin{align*}
x&=y & w&=z & a&=b+c \\
2x&=-y & 2w&=z^2 & a&=b \\
5x-3&=2y+8 & w+5&=z-2 & ab&=c
\\
\end{align*}
```

$$\begin{aligned} x &= y & w &= z & a &= b + c \\ 2x &= -y & 3w &= z & a &= b \\ 5 - 3x &= 2y + 8 & w + 5 &= z - 2 & ab &= c \end{aligned}$$

Denklemlerin alt alta dizilmesinde görüldüğü üzere her denklem ayrı ayrı numaralandırılır. Eğer yazılan tüm denklemlerin ortak tek bir numara almasını istiyorsak equation ortamında split veya alignedat ortamları kullanılabilir.

Tanımlar, Teoremler,...

LaTeX ile hazırlanan bir dokümanda tanım, teorem, kanıt, örnek vb. gibi ifadeleri yazarken ayrı bir ortam tanımlayarak bu ortam içerisinde yazmak gerekir. Bunun için kaynak dosyanın preamble kısmına

```
\newtheorem{kısa ad}[sayaç]{tam ad}[section]
```

komutu girilmelidir. Burada kısa ad ortama vereceğimiz kısa adı ifade eder. Ortamı yazmak için kullanılacak komut kısmına bu ad yazılır. Tam ad bu ortamın çalıştırılması sonucu çıktıda elde edeceğimiz adı göstermektedir. sayaç ise oluşturulan ortamın neye göre numaralandırılacağını belirler. Eğer buraya önceki oluşturulan ortamlardan birinin kısa adı gelirse, o ortamın numarasını takip edecek şekilde ortam numaraları meydana gelir. Son olarak [section] komutu ise ortamın section numaralarını baz alarak numaralandırılmasını sağlar. Örneğin:

```
\newtheorem{tnm}{Tanım}[section]
\newtheorem{teo}[tnm]{Teorem}
```

komutları preamble eklenir ve doküman içerisinde sırasıyla

```
\begin{tnm}
$A$ ve $B$ boştan farklı iki küme olmak
üzere $A$'nın her elemanını $B$'nin yalnız
bir elemanına götüren bağıntıya $A$'dan
$B$'ye fonksiyon denir.
\end{tnm}
```

Tanım 4.1

A ve B boştan farklı iki küme olmak üzere A'nın her elemanını B'nin yalnız bir elemanına götüren bağıntıya A'dan B'ye fonksiyon denir.

elde edilir. Bu ortamlar: örnek, sonuç, önerme, aksiyom,... şeklinde çoğaltılabilir.

```
\begin{teo}
$f:X \rightarrow Y$ fonksiyonun tersinin
olabilmesi için gerek ve yeter şart bu
fonksiyonun birebir ve örten olmasıdır.
\end{teo}
```

Teorem 4.2

$f : X \to Y$ fonksiyonun tersinin olabilmesi için gerek ve yeter şart bu fonksiyonun birebir ve örten olmasıdır.

Bu örneklerden görüldüğü üzere tanım ortamı "section" numarasını baz alarak numaralandırılmış, teorem ortamı ise tanım ortamını takip ederek numaralandırılmıştır. Burada istenilen ortam \label{·} ile etiketlenerek gerekli yerde atıf verilebilir. Ayrıca tanımlanan yeni ortam \begin{ortam adı}[açıklama] şeklinde yazılırsa burada açıklama kısmına yazılacak ifade elde edilecek çıktıda () küme parantezi içinde belirecektir. Örneğin yukarıdaki örnek:

```
\begin{teo}[Terslenebilir fonksiyon]
$f:X \rightarrow Y$ fonksiyonun tersinin
olabilmesi için gerek ve yeter şart bu
fonksiyonun birebir ve örten olmasıdır.
\end{teo}
```

Teorem 4.3 (Terslenebilir fonksiyon)

$f : X \to Y$ fonksiyonun tersinin olabilmesi için gerek ve yeter şart bu fonksiyonun birebir ve örten olmasıdır.

LaTeX'de amsthm paketi ile tanımlanan bu ortamların stillerine müdahale edilebilir. Bunun için amsthm paketi yüklendikten sonra preamble'a \theoremstyle{still adı} komutu girilmelidir. Bu komut kendinden sonra gelen ortama etki etmektedir. Burada sitiller; definition(kalın başlık, düz metin), remark(italik başlık, düz metin) ve plain(varsayılan yani kalın başlık, düz metin) şeklindedir.

amsthm paketi ile birlikte proof (kanıt) adlı bir ortam da gelmektedir. Adı üzerinde bu ortam ile bir teoremin kanıtı verilmektedir.

```
\begin{proof}[Terslenebilir fonksiyon]
$x=3$, $y=4$ ve $z=5$ için $x^2+y^2=z^2$
olur.
\end{proof}
```

Kanıt.

$x = 3$, $y = 4$ ve $z = 5$ için $x^2 + y^2 = z^2$ olur. □

Görüldüğü üzere bu ortamda kanıtın sonunda kanıtın bittiğini gösteren □ işareti bulunmaktadır. Bazı durumlarda bu işaret son satırda yalnız kalabilir. Bu durumda \qedhere komutu ile doğru yere alınabilir.

Şekil Ekleme

Bir LaTeX dosyasına şekil ekleyebilmek için graphicx paketi kullanılmalıdır. Bu paket yüklendikten sonra şekil eklemek için

\includegraphics[opsiyonlar]{dosya adı}

komutu kullanılır. Bu komutta [opsiyonlar] kısmına eklenecek şeklin boyutu, açısı gibi bazı ek seçenekler gelebilir. Bunlara birazdan değineceğiz. {dosya adı} kısmına ise eklemek istediğimiz şeklin adı gelir. Burada dikkat edilmesi gereken şey eklemek istediğimiz şekil dosyasının LaTeX dosyası ile aynı klasörün içinde bulunması gerekmektedir. Aksi halde derleme sırasında hata meydana gelir. Eğer şekil dosyası farklı bir klasör içinde ise bu durumda komut {dosyanın yeri/dosya adı} şeklinde olmalıdır. Farklı bir konumda bulunan bir dosyanın yerini preamble da \graphicspath{{dosya yeri/}} komutu ile belirlemekte mümkündür. Fakat, tüm bu ek komutlar yerine şekil dosyasının aynı klasörde bulundurulması daha işlevseldir.

LaTeX'e eklenecek şekil dosyalarının ne uzantılı olduğu önemlidir. Resim dosyaları; "jpeg, png, bmp,..." gibi bitmap sınıfı ya da "eps, ps,..." gibi vektör sınıfı resimler olabilirler. Eğer LaTeX dosyasına "ps (postscript), eps (encapsulated postscript)" gibi vektör sınıfı bir resim dosyası ekleniyorsa bu durumda dosyamızı dvi olarak derleyebiliriz. Daha sonra bu dvi dosyasından da pdf dosya elde edilebilir. Fakat, LaTeX dosyasına eklenen şekil dosyası "jpg, png" gibi bitmap sınıfı bir dosya ise bu durumda dosyamızı dvi yapamayız. Bundan dolayı dosyayı "LaTeX" ya da "PDFLaTeX" olarak derlemeliyiz. Ancak bu bahsedilen durum artık editörlerin son versiyonlarında ortadan kalkmış ve pek çok editör şekil dosyasının sınıfına bakmadan, dosyaları sorunsuz olarak derlemektedir.

Eklenecek şekil dosyasının türünün önemi olmamasına rağmen vektör sınıfı dosyalar büyüklüğünden bağımsız olarak kalite kaybına uğramadıklarından dolayı daha çok tercih edilmektedirler. Bu yüzden LaTeX dosyalarına eklenecek şekillerin "eps" ya da "ps" uzantılı olması önerilir. Bir şekil dosyası "eps" ya da "ps" tip dosyalara dönüştürmenin pek çok yolu vardır. Örneğin "ghostscript" programı ile bu işlem gerçekleştirilebilir.

Şimdi bir şekil dosyasının eklenmesi ile ilgili bir örnek verelim.

```
\includegraphics{homer.eps}
```

Şimdi bu komuta eklenebilecek opsiyonlardan bahsedelim. İlk olarak eklenecek şeklin boyutu ayarlanabilir. Bunun iki yolu vardır. Birincisi: scale=değer komutunu kullanmaktır. Burada "değer" kısmına gelen ifade şeklin ne kadar büyüyeceği ya da ne kadar küçüleceğini be-

Örneğin scale=0.5 yazılırsa şekil "0.5" kat olarak yani yarısı olacak şekilde bastırılır, eğer scale=2 yazılırsa şekil "2" kat büyüklükte olacak şekilde bastırılır. Örneğin:

```
\includegraphics[scale=.5]{homer.eps}
```

Şeklin boyutunu ayarlamak için ikinci seçenek ise width=xbirim, height=y birim komutudur. Burada "width=xbirim" ile şeklin genişliğinin kaç birim, "height=ybirim" ile de şeklin yüksekliğinin kaç birim olacağı belirlenir. Burada birim olarak "mm, cm, in, dots" gibi ölçü birimleri alınabilir. Örneğin:

```
\includegraphics[width=6cm,
height=4cm]{homer.eps}
```

Görüldüğü üzere bu tarz bir boyut belirlemede en ve boy oranının iyi ayarlanması gerekir. Bu gibi durumlarda keepaspectratio opsiyonu ile en boy oranı uygun olan değer baz alınarak korunur. Örneğin:

```
\includegraphics[width=6cm,
height=4cm,keepaspectratio]{homer.eps}
```

Eğer şeklin genişliğini metnin genişliği kadar olmasını istiyorsak width=\textwidth komutunun eklenmesi yeterlidir.

Opsiyonlara eklenebilecek diğer bir seçenek ise şeklin ne kadar döndürüleceğini belirleyen angle=derece komutudur. Burada "derece" kısmına yazılacak değer ile şeklin ne kadar döndürüleceği belirlenir. Örneğin:

```
\includegraphics[angle=45]{homer.eps}
```

graphicx paketi ile LaTeX dosyasına şekil eklemenin yanı sıra getirmiş olduğu bazı ek komutlar sayesinde metin ve diğer ifadeler üzerinde de düzenlemeler yapılabilmektedir. Bu komutlardan iki tanesinden bahsedelim.

İlk komutumuz

\scalebox{yatay ölçek}[düşey ölçek]{ifade}

dır. Bu komut ile istenilen bir metnin, şeklin veya matematiksel ifadenin ne kadar büyütüleceğini ya ada küçültüleceği belirlenir. Burada "yatay ölçek" ifadenin yatay olarak "düşey ölçek" de düşey olarak ne kadar ölçeklendirileceğini belirler. "ifade" kısmına ise ölçeklendirmek istediğimiz şey yazılır. Örneğin:

```
\scalebox{2}[1.5]{$x^2+y^2$}
```

$$x^2 + y^2$$

Diğer bir komut ise

\rotatebox[origin=döndürme merkezi]{döndürme açısı}{ifade}

dır. Bu komut ile istenilen bir metnin, şeklin veya matematiksel ifadenin ne kadar döndürüleceği belirlenir. Burada "döndürme merkezi" ile şeklin nereye göre döndürüleceği, "döndürme açısı" ile kaç derece döndürüleceği belirlenir. Örneğin:

```
\rotatebox[origin=c]{45}{$x^2+y^2$}
```

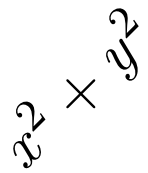


Table ortamında olduğu gibi şekillere atıf vermek ve şekiller listesi oluşturabilmek için eklenecek şekilleri figure ortamında yazmalıyız. Bunun için:

```
\begin{figure}[konum]
\centering
\includegraphics[opsiyonlar]{dosya adı}
\caption{···}
\label{···}
\end{figure}
```

komutu ile şekil eklenmelidir. Burada konum, table ortamında olduğu gibi h,t,b,p parametrelerinden biri veya birkaçı olabilir. \label şekle atıf verebilmek için kullanacağımız etiketi, \caption ise şekle ekleyeceğimiz açıklamayı yazmamızı sağlayan komutlardır. \centering ise eklenecek şeklin ortalanmasını sağlar. Örneğin:

```
\begin{figure}[!ht]
\centering
\includegraphics{homer.eps}
\caption{Homer Simpson ve donut'ı}
\end{figure}
```

Şekil 1: Homer Simpson ve donut'ı

table ortamında olduğu gibi figure ortamında da eklenen şekillerin açıklama başlığı olan Şekil adını ve sitilini istediğimiz gibi değiştirebiliriz. Bunun için benzer şekilde caption paketi ile birlikte

\captionsetup[figure]{name={İstenilen ad},labelfont={font adı}, textfont={font adı} labelsep=isim ayracı}

komutu kullanılmalıdır. Buradaki bazı temel ayarların nasıl yapıldığını hatırlamak için table ortamına bakabilirsiniz.

Vurgulayacağımız son nokta ise Şekil başlığının ardışık değil de section gibi herhangi bir başlığa bağlı olarak değişmesini istiyorsak amsmath paketi ile birlikte \numberwithin{figure}{section} komutu kullanılabilir.

Kaynakça Oluşturma

Hazırlanan bir dokümana kaynakça ekleyebilmek için thebibliography ortamı kullanılır. Bu ortamın komutları şu şekildedir:

```
\begin{thebibliography}{referans sayısı}
\bibitem[etiket]{referans etiketi}
\end{thebibliography}
```

Burada referans sayısı kaynakçada girilecek maksimum kaynak sayısını belirtir. etiket kaynakçadaki referanslara verilecek etiketi ve son olarak referans etiketi'de kaynakçada verilen referanslara metin içinde verilecek atıflarda kullanılacak etiketi belirtir. Metin içinde bir kaynağa etiket vermek için \cite{referans etileti} komutu kullanılır. Örneğin:

```
\begin{thebiblography}{9}
\bibitem{knuth} Knuth, D. E., & Bibby, D. (1984).
The texbook (Vol. 15). Reading: Addison-Wesley.
\bibitem{lamport}Leslie Lamport, LaTeX: a document
preparation system, Addison Wesley, Massachusetts,
2nd edition,1994. \end{thebiblography}
```

Kaynaklar

[1] Knuth, D. E., & Bibby, D. (1984). The texbook (Vol. 15). Reading: Addison-Wesley.

[2] Leslie Lamport, LaTeX: a document preparation system, Addison Wesley, Massachusetts, 2nd edition, 1994.

Yukarıdaki şekilde oluşturulan kaynaklara metin içinde atıf verelim. Örneğin:

```
Donald Knuth \cite{knuth} kitabından TEX
programını ve işleyişini sunmuştur. Daha
sonra Leslie Lamport, TEX üzerinde çalışan ve
TEX'i kullanmayı kolaylaştıran LATEX yazılımını
\cite{lamport} eserinde verilmiştir.
```

```
Donald Knuth [1] kitabından TEX programını
ve işleyişini sunmuştur. Daha sonra Leslie
Lamport, TEX üzerinde çalışan ve TEX'i kullanmayı
kolaylaştıran LATEX yazılımını [2] eserinde
verilmiştir.
```

Dokümana eklenen kaynakçanın içindekiler tablosuna eklemenin iki yolu vardır. Bunlardan ilki \begin{thebibliography}'nin başına:

\addcontentsline{toc}{section}{istenilen ad}

komutunun eklenmesidir. Burada istenilen ad kısmına ne yazılırsa içindekiler tablosunda o belirir. Burada problem ise bu komutla içindekiler tablosuna eklenen kaynakça başlık numarası almaz.

İkinci yöntem ise tocbibind paketini kullanmaktır. Bu paket

\usepackage[numbib]{tocbibind}

şeklinde eklenir. Burada numbib opsiyonu içindekiler tablosunda kaynakçanın başlık numarası almasını sağlar. Eğer başlık numarası istenmiyorsa bu opsiyon kaldırılabilir.

Bir dokümana kaynak eklemenin daha profesyonel yöntemi BibTEX kullanmaktır. BibTEX ile genel olarak kullanılacak kaynaklar ".bib" uzantılı bir dosyada uygun bir formatta kaydedilerek bu dosya istenilen her TEX dosyasında kullanılabilir. BibTEX'in nasıl kullanıldığı burada ayrıca anlatılmayacaktır. İlgili kişiler, gerekli dokümanlardan "bib" dosyasının oluşturulmasını ve kullanımını kolayca öğrenebilir.

Dizin Ekleme

Bir kitapta dizin oluşturmak oldukça önemlidir. LaTeX ile dizin oluşturmakta oldukça kolaydır. Bunun için ilk olarak makeidx paketini yüklemek gerekir. Daha sonra dizinlemeyi çalıştırmak için preamble kısmına yani \begin{document}'dan önce \makeindex komutu girilmelidir. Bundan sonra metin içerisinde dizinlemek istediğimiz her kelimeden hemen sonra \index{kelime} komutu yazılmalıdır. Eğer dizinlenen kelimenin bir alt tanımı var ise bu durumda komut \index{kelime!alt kelime} şeklinde girilmelidir. Örneğin dokümanda geçen fonksiyonları dizinlemek istiyoruz. Bu durumda istediğimiz sayfalarda fonksiyonlardan sonra \index{fonksiyon} komutunu girmeliyiz. Eğer ayrıca terslenebilir fonksiyonları da fonksiyonlar dizininin altında vermek istiyorsak bu durumda dizinlemek istediğimiz terslenebilir fonksiyonlar için \index{fonksiyon!terslenebilir fonksiyon} komutu kullanılmalıdır.

Eğer index komutu \index{kelime@istenilen ifade} şeklinde girilirse kelime @ işraretinin solunda yazıldığı gibi dizilenecek fakat @ işaretinin sağında yazıldığı gibi dizin başlığında gözükecektir.

Dizinlediğimiz kelimeleri listelemek için \printindex komutu kullanılır. Bu komut "Dizin" başlığı altında dokümanda \index{kelime} komutu ile yazmış olduğumuz kelimeleri geçtiği sayfa numaraları ile birlikte listeleyecektir.

Son olarak dizin başlığında sayfa numaraları ile dizinlenmiş kelimelerin yanlarında bulunan sayfa numaralarına tıklandığında ilgili sayfaya gidilebilmesi için hyperref paketi kullanılmalıdır.

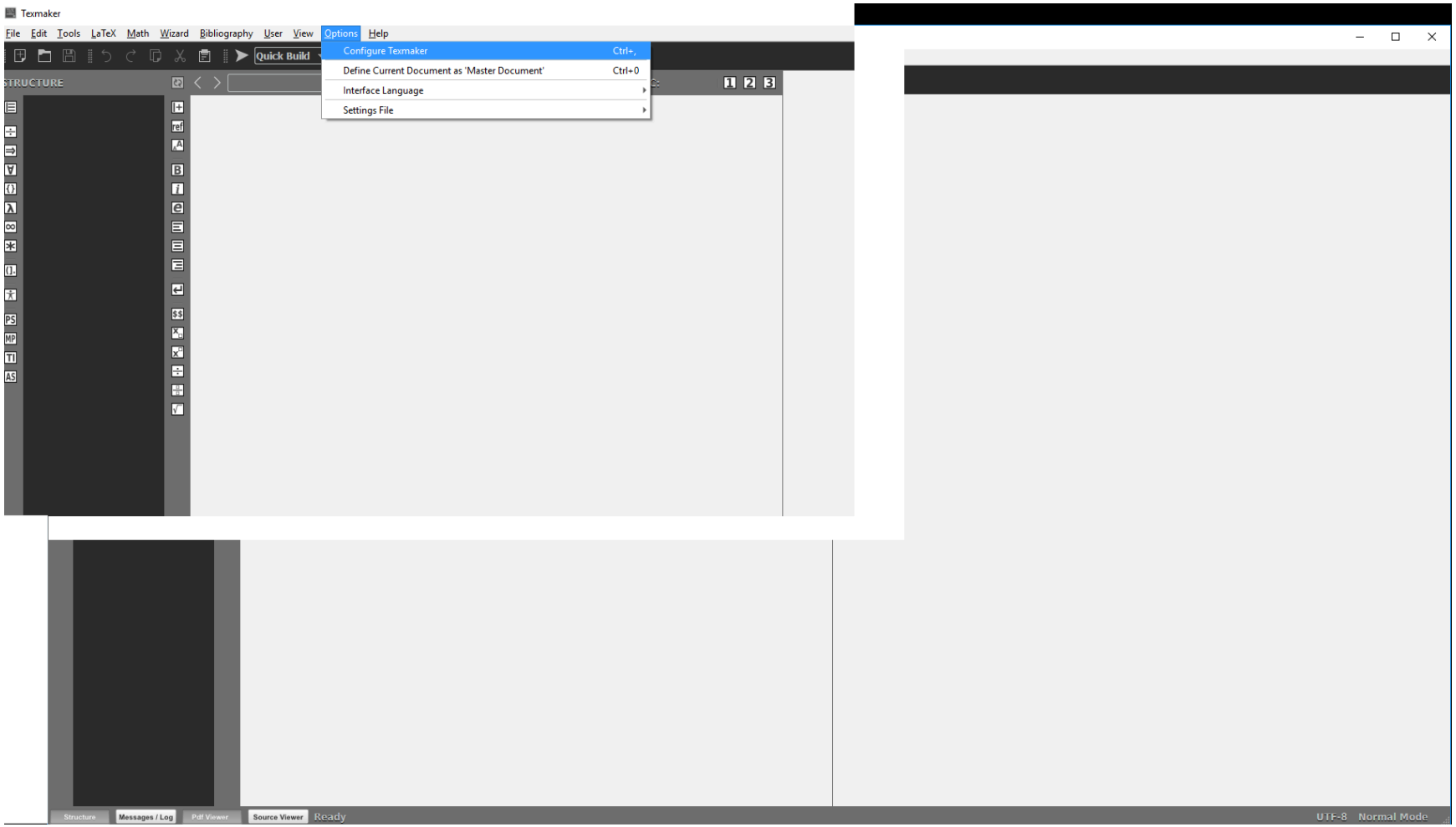
Grafik Çizimi

LaTeX matematiksel ifadelerin yazımında oldukça kolaylık sağlamasının yanı sıra verilen komutlar ile grafik çizimi içinde oldukça kullanışlıdır. LaTeX'de grafik çizimi için farklı paketler ve farklı ortamlar kullanılabilmektedir. picture ortamı LaTeX ile doğrudan çalışan ve çizim için kullanabilecek bir ortamdır. Fakat kendine göre bazı dezavantajları olmasından dolayı biz çizim için pstricks paketi ve bu paket ile gelen komutların kullanımını vereceğiz. Fakat daha güncel olan tikz paketi ile de benzer şekilde grafik çizdirmek mümkündür. Bu iki paketin birbirlerine göre avantaj ve dezavantajları bulunmakla birlikte hangi paketin kullanılacağı tamamen yazarın tercihine kalmıştır.

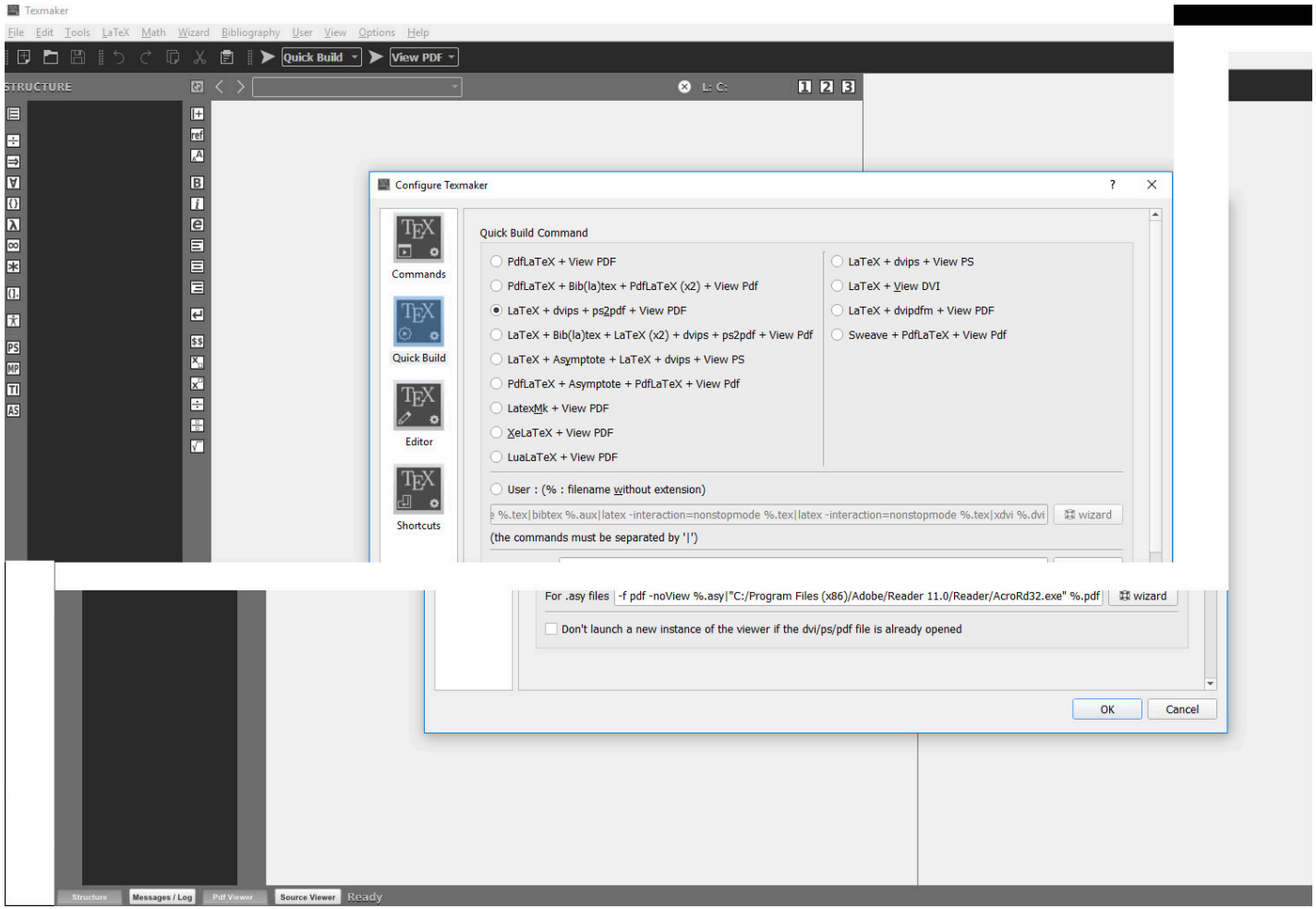
Pstricks Paketi

pstricks ile çizim yapabilmek için ilk olarak \usepackage{pstricks} ile pstricks paketi yüklenmelidir. Burada dikkat edilmesi gereken iki husus vardır. Birincisi, eğer bilgisayarda pstricks paketleri yüklü değilse, "MiKTeX console" yardımıyla alertpstricks ve bu paketle alakalı paketlerin yüklenmesi gerekir. İkinci olarak, pstricks ile çizim yapıldığında LaTeX kaynak dosyasını PdfLaTeX olarak derlenmemesi gerekir. Her ne kadar PdfLaTeX kullanılarak yapılan derlemelerin çalışması için farklı yöntemler bulunsa da bu tür dosyaların derlenebilmesi için en doğru yol, derleme yöntemini LaTeX+dvips+ps2pdf şeklinde belirlemektir. Bu yöntem Texmaker da şu şekilde yapılır:

İlk olarak Texmaker'ın üst araç çubuğundan "Texmaker'ı yapılandır", yani Configure TexMaker, seçilir.


Texmaker
File Edit Tools LaTeX Math Wizard Bibliography User View Options Help
Quick Build
Configure Texmaker Ctrl+,
Define Current Document as 'Master Document' Ctrl+0
Interface Language
Settings File
STRUCTURE
Structure
Messages / Log
Pdf Viewer
Source Viewer
Ready
UTF-8 Normal Mode

Açılan pencerenin sol sütunundaki “Hızlı İnşa”, yani Quick Build, seçilerek gelen pencereden hızlı inşa komutu olarak aşağıdaki fotoğrafta da görüldüğü üzere LaTeX+dvips+ps2pdf+View PDF seçilerek tamam denmelidir.



Böylece Texmaker'da pstricks paketi ile yapılan çizimler sorunsuz bir şekilde derlenebilecektir.

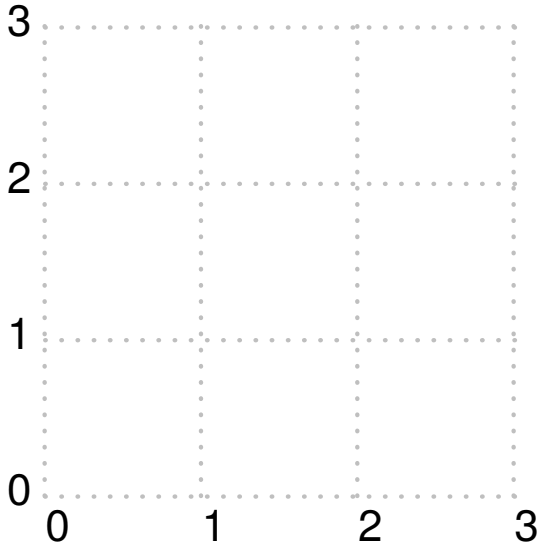
Not: Derleme yöntemi, XeLaTex veya LuaLaTeX seçilerek de çizim yapılabilmektedir.

Şimdi pspicture oramı ile nasıl çizim yapılacağını görelim:

\begin{pspicture}[opsiyon]$(x_0,y_0)(x_1,y_1)$...\end{pspicture}

ile pspicture ortamı oluşturulur. Çizmek istediğimiz grafiklerin komutları bu ortam içerisine girilir. Bu komutta (x_0,y_0) çizim için oluşturulacak dikdörtgensel bölgenin sol alt köşesinin koordinatlarını, (x_1,y_1) ise sağ üst köşesinin koordinatlarını göstermektedir. opsiyon kısmına ise ileride vereceğimiz, grafikleri özelleştirecek bazı ek komutlar gelir. Örneğin [showgrid=true] komutu eklenirse, bölgenin koordinatları belli eden bir ızgara eklenecektir. Hemen bir örnek ile bu komut sonucu nasıl bir çıktı alacağımızı görelim:

```
\begin{pspicture}[showgrid=true](0,0)(3,3)
\end{pspicture}
```

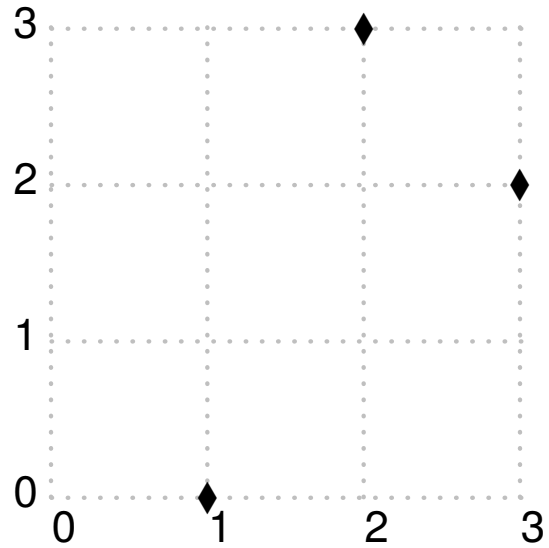


Görüldüğü üzere yukarıdaki komutlar sonucunda üçe üçlük bir karesel bölge elde etmiş olduk. Artık bu bölgede istenilen geometrik şekilleri ya da fonksiyonların karşılık geldikleri eğrileri çizdirebiliriz. pstricks paketinde çoğu geometrik ifadenin çizdirilebilmesi için bir komut bulunmaktadır. Şimdi bu komutları sırasıyla verelim.

Nokta Çizimi

\psdot$(x_0, y_0)$ komutu ile (x_0, y_0) noktasına bir nokta çizdirilir. \psdots$(x_0, y_0)(x_1, y_1)...(x_n, y_n)$ komutu ile ise belirtilen koordinatlara aynı anda birden fazla nokta çizdirilmesi sağlanır. Burada ayrıca opsiyon olarak [dotsize=xpt] eklenebilir. Bu notanın boyutunu belirler. Varsayalan değer "2pt" dur. Ayrıca [dotstyle=stil] opsiyonu ile noktanın sitili de belirlenebilir. Örnek bir iki sitil olarak square, square*, diamond, triangle verebiliriz. Burada komutların "*" lı olanları için dolu olacak şekilde çıktı verir. Örneğin:

```
\begin{pspicture}[showgrid=true]
(0,0)(3,3)
\psdots[dotsize=4pt,dotstyle=diamond*]
(1,0)(2,3)(3,2)
\end{pspicture}
```



Çizgi Çizimi

\psline$(x_0, y_0)(x_1, y_1)$ komutu ile (x_0, y_0), (x_1, y_1) noktaları arasında bir çizgi çizdirilir. Eğer (x_0, y_0) gibi tek bir nokta verilirse bu durumda $(0, 0)$ noktasından (x_0, y_0) noktasına bir çizgi çizdirilir. Benzer şekilde $(x_0, y_0)(x_1, y_1)...(x_n, y_n)$ şeklinde $n + 1$ adet nokta verilirse sırasıyla noktaları birleştiren çizgiler çizdirilir. psline için opsiyonları aşağıdaki gibi sıralayabiliriz.

i) [linewidth=xpt] çizgi kalınlığını belirler. Varsayılan değer "0.8pt" dır.

ii) [linecolor=renk] ile çizgi rengi belirlenebilir.

iii) [linestyle=stil] ile çizgi sitili belirlenir. Burada sitil olarak none, dashed, dotted, solid olabilir. Varsayılan "solid" dir. "dashed" stili için dash=xpt ypt komutu ile çizgilerin boyutu ayarlanabilir. Varsayılan "5pt 3pt" dır. "dotted" sitili için dotsep=xpt ile noktalar arası boşluklar ayarlanabilir. Varsayılan "3pt" dır.

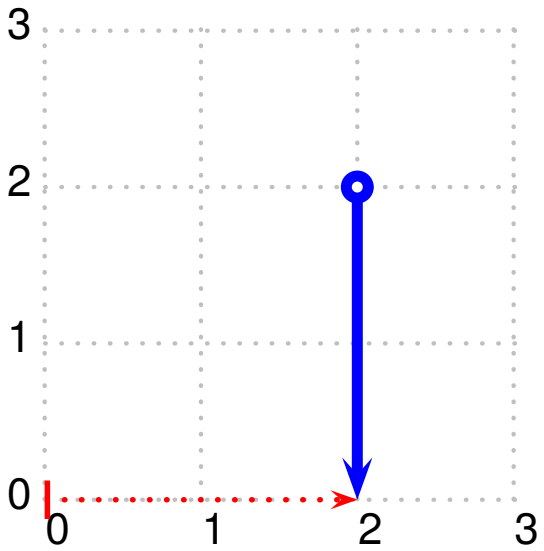
iv) [doubleline=true] ile çift çizgi çizdirilir. doublesep=xpt ile çizgiler arası mesafe ayarlanabilir. Varsayılan "1.25 \pslinewidth" dir.

v) [linearc=xpt]ile çizgilerin ne kadar yuvarlanacağını belirleyebiliriz. Varsayılan "0pt" dır.

vi) {<->}ile çizgilere ok eklenebilir. Komut bu şekilde verilebilirse her çizginin her iki ucuna ok eklenir. Bu komut, {->}, {<-}, {o->}, {|->} şeklinde de kullanılabilir.

Örneğin

```
\begin{pspicture}[showgrid=true](0,0)(3,3)
\psline[linewidth=1pt,linecolor=red,linestyle=dotted,
dotsep=.25pt]{|->}(0,0)(2,0)
\psline[linewidth=2pt,linecolor=blue,linestyle=solid]
{<-o}(2,0)(2,2) \end{pspicture}
```

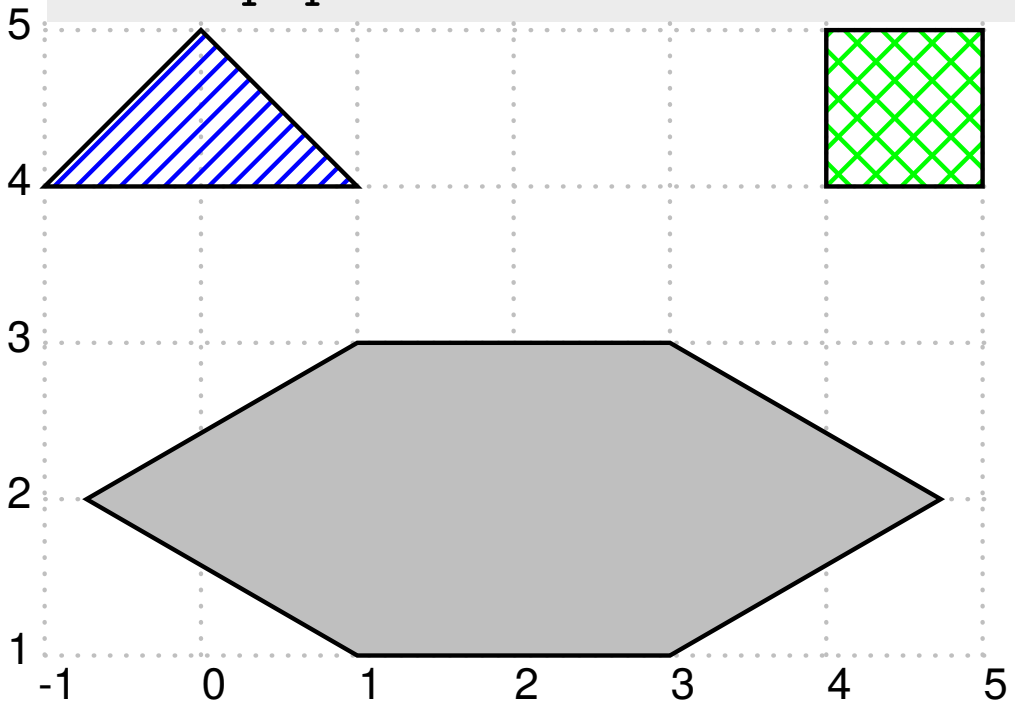


Not: Yukarıdaki örneklerden de görüleceği üzere şekillerimizi figure ortamında çizdirmediğimizden şekillerin dosyadaki yerlerini belirlemek, şekillere açıklama eklemek ve etiketlemek mümkün değildir. Bu yüzden şekilleri figure ortamında çizdirmemiz gerekir. Ayrıca görüleceği üzere şekilleri ortalayarak çizdirmediğimizden sola yaslı olacak şekilde sayfaya dizilmektedir.

Çokgen Çizimi

\pspolygon$(x_0, y_0)(x_1, y_1)...(x_n, y_n)$ komutu, köşeleri belirtilen noktalarda olan bir çokgen çizdirir. Bu komut psline komutuna benzer olarak çalışır fakat kapalı bir bölge çizdirir. Dolayısıyla psline için verilen opsiyonların hepsini bu komut içinde kullanabiliriz. Ayrıca bu komut kapalı bir bölge oluşturduğundan bu bölgeyi istediğimiz bir renk ile boyatabiliriz. Bunun için [fillcolor=renk,fillstyle=sitil] opsiyonları kullanılabilir. Burada sitil olarak şunları kullanabiliriz: none, solid, vlines, hlines, crosshatch. Ayrıca vlines, hlines, crosshatch komutları ile şeklin içini doldurmak için kullanılan çizgiler için bazı ek komutlar mevcuttur. hatchwidth=xpt ile çizgilerin kalınlığı belirtilir. Varsayılan değer "0.8pt" dir. hatchsep=xpt ile çizgiler arasındaki boşluk belirtilir. Varsayılan değer "4pt" dir. hatchcolor=renk ile çizgilerin rengi belirtilir. hatchrot=x ile çizgilerin yaptığı açı belirtilir. Varsayılan değer "45" dir. Bu komutların kullanılışını bir örnek ile gösterelim:

```
\begin{pspicture}[showgrid=true](-1,1)(5,5)
\pspolygon[fillcolor=lightgray,
fillstyle=solid](1,1)(3,1)(4.732,2)(3,3)(1,3)
(-0.732,2)(1,1)
\pspolygon[fillstyle=crosshatch,
hatchcolor=green](4,4)(5,4)(5,5)(4,5)
\pspolygon[fillstyle=hlines,hatchcolor=blue,
hatchsep=2pt](-1,4)(0,5)(1,4)
\end{pspicture}
```



Herhangi bir paket yüklemeksizin grafiklerde ve şekillerde kullanılacak renkler aşağıdaki gibidir:

LaTeX'de kullanılabilecek renkler

İsim	Renk	İsim	Renk	İsim	Renk
red		purple		lime	
green		gray		teal	
blue		orange		pink	
yellow		cyan		olive	
black		magenta		lightgray	
white		brown		darkgray	

Not: opacity=değer seçeneği ile renklerin saydamlığı ayarlanabilir. "1" tam rengi verirken, "0.5" %50 saydamlaştırır.

Not: Doldurma sitillerinden biri de gradient'dir. Bu seçenek ile bir bölge ilk belirlenen renkten ikinci belirlenen renge doğru değişecek şekilde doldurulabilir. Bu komut için pst-grad paketi gereklidir. Daha ayrıntılı bilgi için paket içeriğini inceleyebilirsiniz.

Yukarıda verilen renkler dışında daha pek çok farklı renk kullanmak mümkündür. Bunun için pstricks paketi dvipsnames seçeneği ile yüklenmesi gerekir. Bu durumda 68 farklı renk tanımlanır. Bu renklerin hepsini burada vermek mümkün olmadığından gerekli belgelerden renkler ve adları incelenebilir. Ayrıca xcolor paketi, bir dokümandaki metin, matematiksel ifade, tablo renkleri gibi pek çok şeyin rengini istenildiği şekilde tanımlanmasına ya da tanımlı renklerin kullanılmasına olanak sağlar. Bu paket de dvipsnames seçeneği ile çalıştırılabilir. Burada dikkat edilmesi gereken nokta eğer xcolor paketi pstricks paketi ile beraber çalıştırılacaksa düzgün çalışması için ilk olarak xcolor paketinin yüklenmesi gerektiğidir.

Tanımlı bir rengin ton oranı renk!değer ile belirlenebilir. Burada "!" işareti "%" görevi görmektedir. Yani blue!50 denildiğinde mavi rengi %50 oranıyla gösterilecektir. Örneğin:

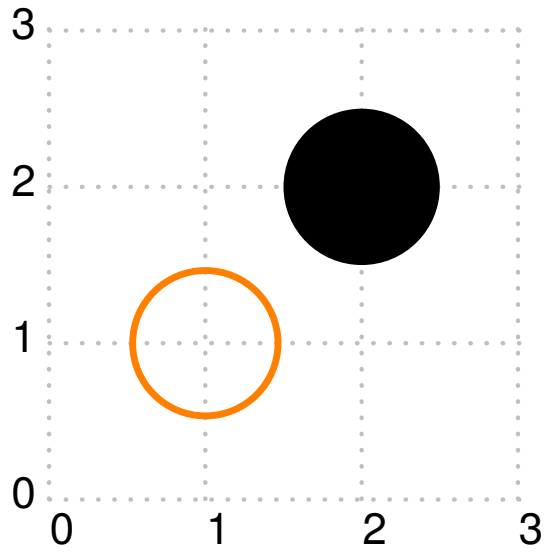
```
\crule[blue]{.35cm}{.35cm}
\crule[blue!50]{.35cm}{.35cm}
```

Çember Çizimi

\pscircle$(x_0, y_0)\{r\}$ komutu merkezi (x_0, y_0) yarıçapı r olan bir çember çizdirir. Bu komuta psline komutuna uygulanan her opsiyon uygulanabilir.

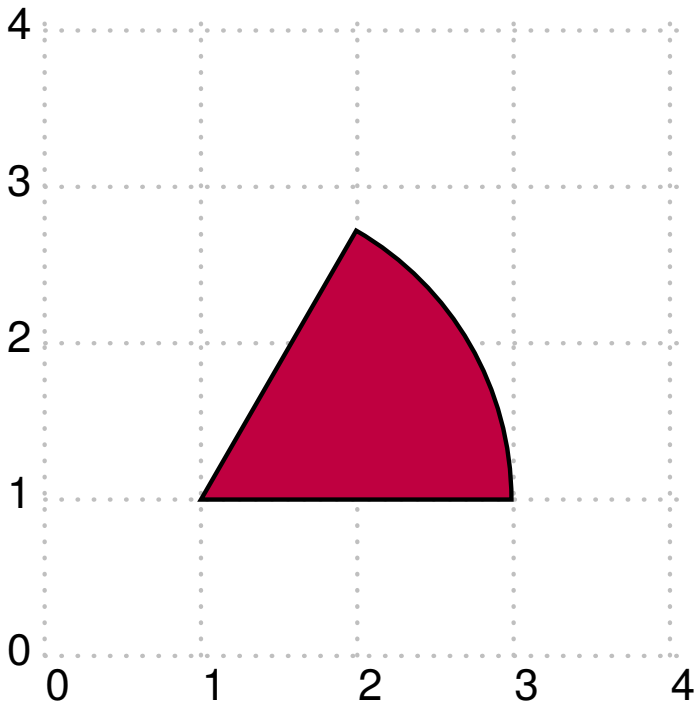
\qdisk$(x_0, y_0)\{r\}$ komutu ise merkezi (x_0, y_0) yarıçapı r olan içi dolu bir disk çizdirir. Örneğin:

```
\begin{pspicture}[showgrid=true]
(0,0)(3,3)
\pscircle[linewidth=1.25pt,linecolor=
orange](1,1){0.5}
\qdisk(2,2){0.5}
\end{pspicture}
```



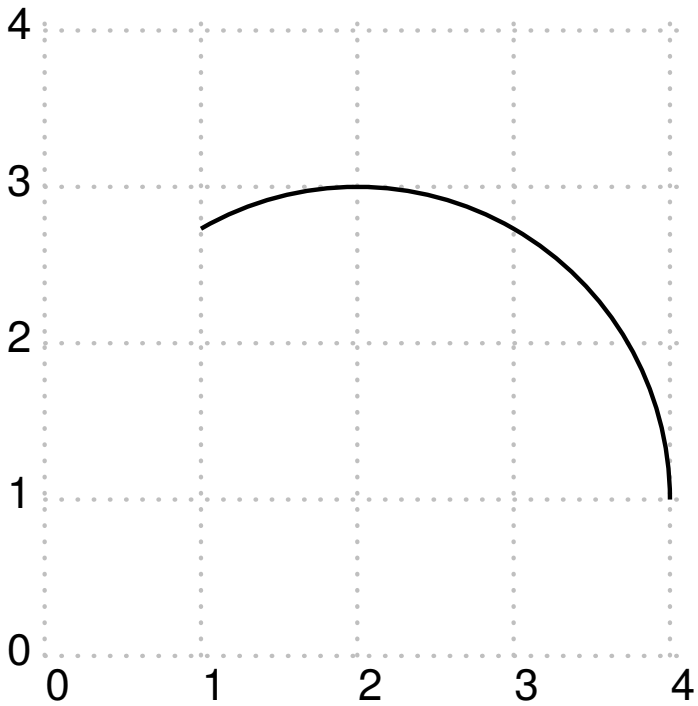
\pswedge $(x_0, y_0)\{r\}\{$açı1$\}\{$açı2$\}$ komutu (x_0, y_0) merkezli r yarıçaplı bir çemberin saat yönünün tersinde açı1'den açı2'ye kadar olacak şekilde dilimini çizdirir. Örneğin:

```
\begin{pspicture}[showgrid=true](0,0)(4,4)
\pswedge[fillcolor=purple, fillstyle=solid](1,1){2}{0}{60}
\end{pspicture}
```



\psarc$(x_0, y_0)\{r\}\{$açı1$\}\{$açı2$\}$ komutu (x_0, y_0) merkezli r yarıçaplı bir çemberin saat yönünün tersinde açı1'den açı2'ye kadar olacak şekilde yayını çizdirir. Örneğin:

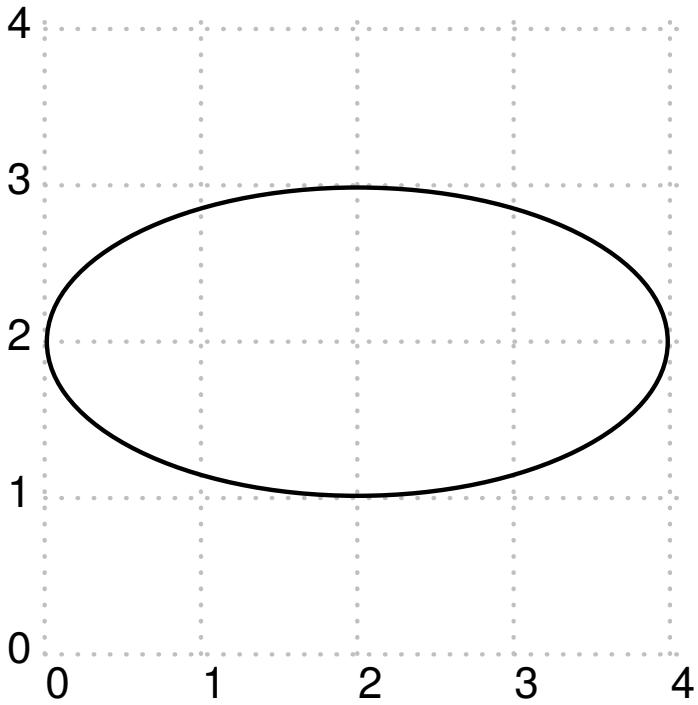
```
\begin{pspicture}[showgrid=true](0,0)(4,4)
\psarc(2,1){2}{0}{120}
\end{pspicture}
```



Elips Çizimi

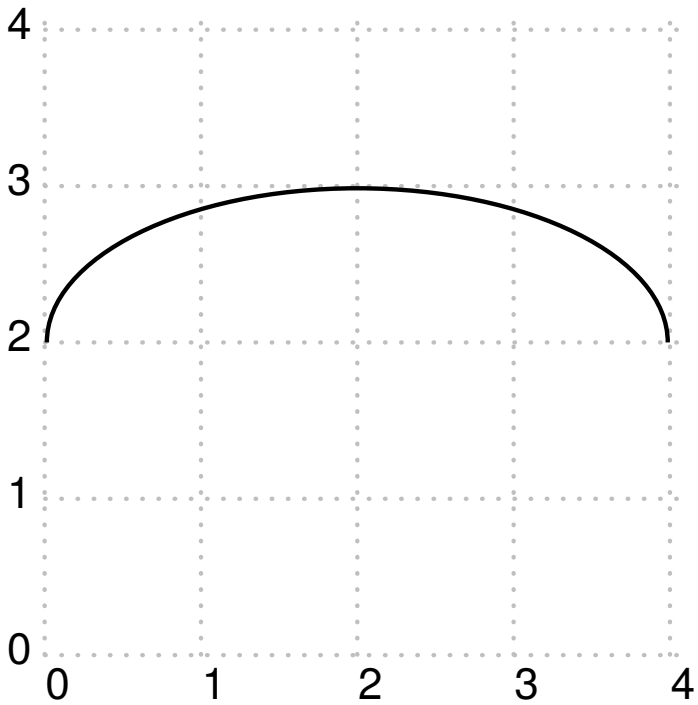
\psellipse$(x_0, y_0)(r_x, r_y)$ komutu merkezi (x_0, y_0) düşey yarıçapı r_x, dikey yarıçapı r_y olan bir elips çizdirir. Örneğin:

```
\begin{pspicture}[showgrid=true](0,0)(4,4)
\psellipse(2,2)(2,1)
\end{pspicture}
```



Çembere benzer şekilde \psellipticarc$(x_0, y_0)(r_x, r_y)$\{açı1\}\{açı2\} komutu merkezi (x_0, y_0) düşey yarıçapı r_x, dikey yarıçapı r_y olan bir elipsin saat yönünün tersinde açı1'den açı2'ye kadar olacak şekilde elips yayını çizdirir. Örneğin:

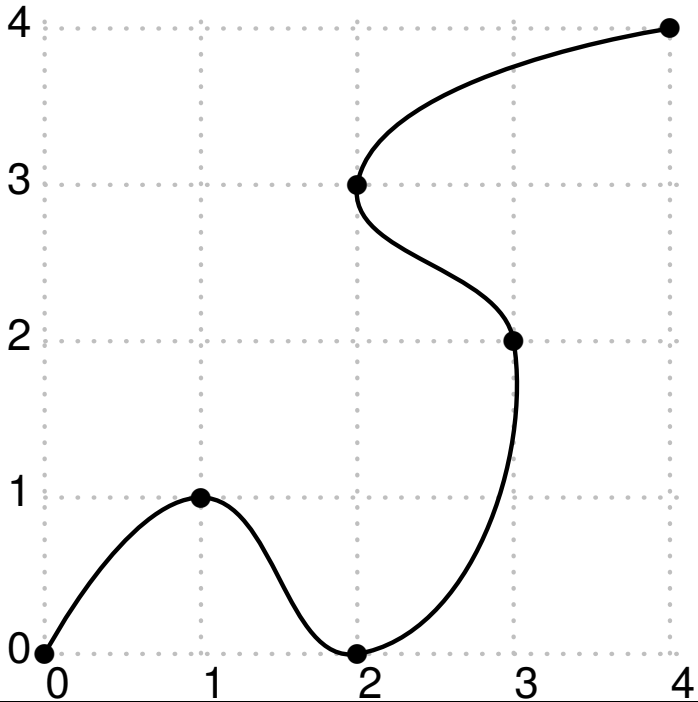
```
\begin{pspicture}[showgrid=true](0,0)(4,4)
\psellipticarc(2,2)(2,1){0}{180}
\end{pspicture}
```



Eğri Çizimi

\pscurve$(x_0, y_0)(x_1, x_2)...(x_n, y_n)$ belirtilen noktalardan geçen bir eğri çizdirir. Eğrinin geçtiği noktaları görmek için showpoints=true seçeneği kullanılabilir. Örneğin:

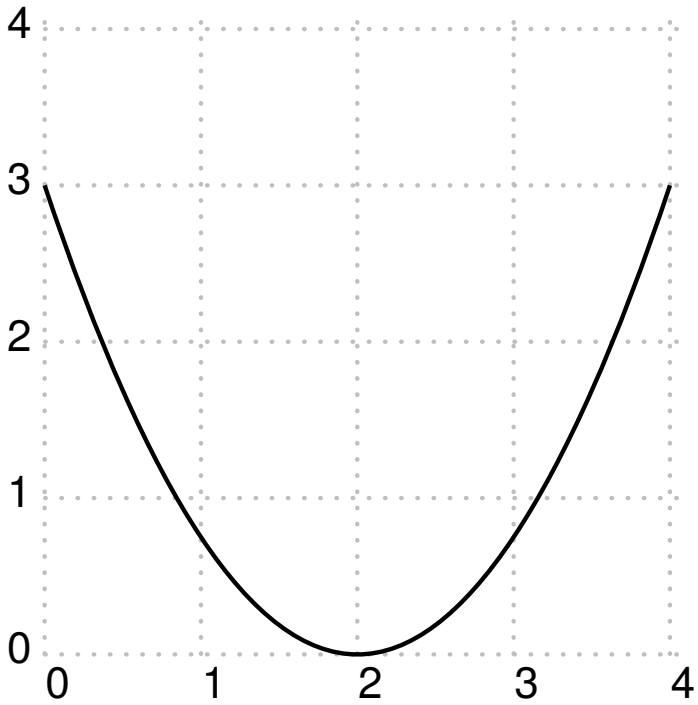
```
\begin{pspicture}[showgrid=true](0,0)(4,4)
\pscurve[showpoints=true](0,0)(1,1)(2,0)(3,2)(2,3)(4,4)
\end{pspicture}
```



Parabol Çizimi

\psparabola$(x_0, y_0)(x_1, y_1)$ başlangıç noktası (x_0, y_0) ve maksimum/minimum noktası (x_1, y_1) olan bir parabol çizdirir. Örneğin:

```
\begin{pspicture}[showgrid=true](0,0)(4,4)
\psparabola(0,3)(2,0)
\end{pspicture}
```



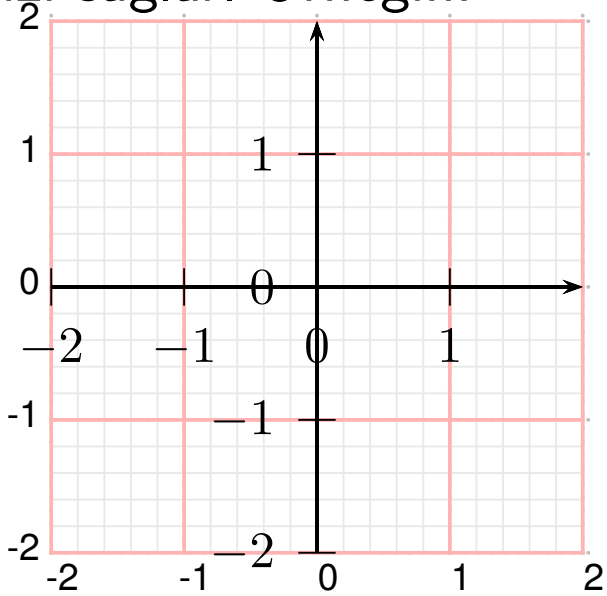
Fonksiyon Grafiklerinin Çizimi

LaTeX'de pstricks paketi ile bazı basit fonksiyonların grafiklerini çizmek mümkündür. Bunun için pst-plot paketini kullanmak gerekir. Bir fonksiyonun grafiğini çizmeden önce eksenleri çizdirelim. Bunun için psaxes komutunu kullanacağız. Bu komut şu şekilde çalıştırılır:

\psaxes {->} $(x_0, y_0)(x_1, y_1)(x_2, y_2)$

Bu komut ile bir analitik düzlem oluşturulmaktadır ve bu düzlemin orjini (x_0, y_0), sol alt köşesi (x_1, y_1), sağ alt köşesi (x_2, y_2) noktaları ile belirlidir. {->} ise eksenin oklarını eklememizi sağlar. Örneğin:

```
\begin{pspicture}[showgrid=true]
(-2,-2)(2,2)
\psaxes{->}(0,0)(-2,-2)(2,2)
\end{pspicture}
```



Yukarıdaki örnekten de görüleceği üzer eksenler ızgaralarla birlikte çizdirildiğinde ızgaraların numaraları ile eksenlerinki aynı anda iyi bir görüntü vermemektedir. Bu yüzden gerekli çizimler yapıldıktan sonra ızgaraların kaldırılması daha iyi bir görüntü verir.

psaxes komutuna gelebilecek opsiyonlar ve bunların işlevleri aşağıda verilmiştir

- Orjini ifade etmek için beliren O ifadesini kaldırmak için showorigin=false seçeneği kullanılabilir.
- ticks seçeneği eksenlerdeki çentikleri ifade etmekte olup: ticks=all hem x hem de y de çentik olmasını, ticks=x yalnız x ekseninde çentik olmasını, ticks=y yalnız y ekseninde çentik olmasını ve ticks=none hiç bir eksende çentik olmamasını sağlar.
- tickstyke seçeneği eksenlerdeki çentiklerin sitilini belirler. tickstyle=full çentiklerin hem eksenin aşağı hem de yukarısında olacak şekilde çizdirir, tickstyle=top çentikleri yalnızca eksenin yukarısında olacak şekilde çizdirir, tickstyle=bottom ise çentikleri yalnızca eksenin aşağında olacak şekilde çizdirir.

- ticksize seçeneği çentiklerin boyutunun ayarlanmasını sağlar. ticksize=xpt şeklinde kullanılır. Varsayılan değer "3pt" dir.
- labels seçeneği eksenlerdeki değerlerin yazdırılıp yazdırılmayacağını belirler. Varsayılan "all" dır. labels=x seçeneği yalnızca x eksenindeki değerlerin, labels=y ise yalnızca y eksenindeki değerlerin yazdırılmasını sağlar. labels=none ise iki eksendeki değerlerinde yazdırılmamasını sağlar.
- labelFontSize eksenlerdeki sayıların boyutunu ayarlar. Bunlara font büyüklüklerini ayarlamak için kullandığımız \tiny, \small... gibi komutlar gelir. Eğer yalnızca x veya y ekseninin boyutunu ayarlamak istiyorsak xlabelFontSize veya ylabelFontSize komutlarını kullanmalıyız. Fakat buradaki komutlar math ortamında yanı $$ arasında yazmak gerekir. Bu komutlar mathLabel=false komutu ile birlikte kullanılırsa math ortamında yazılmasına gerek yoktur.

- xLabels={a,b,c}, yLabels={a,b,c} komutları ile sırasıyla x ve y eksenlerindeki değerler, komutların eşitliklerine yazılan değer ile gösterilebilir.
- xlabelFactor,ylabelFactor komutları, eksen değerlerini bir sayının katı olarak göstermek için kullanılır. Örneğin x eksenindeki değerler 10^6 büyüklüğünde sayılar olmalıysa, opsiyona xLabelFactor=\cdot 10^6 yazılması yeterlidir.
- fractionLabels, xfractionLabels, yfractionLabels, fractionLabelBase, xfractionLabelBase, yfractionLabelBase komutları eksendeki değerlerin kesirli yazılması için kullanılır. Eğer eksen değerleri kesirli yazılmak isteniyorsa fractionLabels opsiyonu ile birlikte hangi eksen kesirli yazılmak isteniyorsa ona ait olan komutlar kullanılmalıdır. Örneğin yalnızca x eksenindeki değerler $1/3$ oranlı olarak yazılmak isteniyorsa yazılması gereken komut [xfractionLabels,xfractionLabelBase=3] dır. Tabii burada eksen noktalarının doğru çıkması için bazı ekstra ayarlar yapmak gerekir. Buna ilerleyen kısımda inceleyeceğiz.

- Ox, Oy seçenekleri sırasıyla orjinin x ve y eksenlerindeki konumunu belirler. Varsayılan değerler "0" dır. Bu değerler tam sayı olmalıdır.
- Dx, Dy seçenekleri eksenlerdeki değerlerin artışını belirler. Dx=x ile x eksenindeki değerler x birim artmaktadır, Dy=y ile ise de y eksenindeki değerler y birim artmaktadır. Varsayılan değerleri "1" dir. Bu değerler ondalıklı olabilir.
- dx, dy seçenekleri eksen etiketleri arasındaki mesafeyi belirler. Örneğin x eksenin pozitif kısmı 4'e kadar değerler alıyor olsun. Bu durumda: dx=0 veya dx=1 olması durumunda x eksenindeki değerler beklendiği gibi 1'den 4'e kadar ardışık sayılar olur. Fakat dx=0.5 alırsak noktalar arası 0.5 birim olacağından 1'i x ekseninde 0.5'e koyar ve 1'den sonra ardışık gelen her sayıyı 0.5 birim mesafe ile yerleştirir. Dolayısıyla 4'e karşılık gelen noktaya 8 yazdırmış oluruz. Benzer şekilde dx=2 olursa 1'i x eseninde 2'ye 2'yi de 4'e yazdırır. Kısacası bir grafik çiziminde x ekseni için girilen değer aralığı"A" ve "dx=a" ise x eksende "A/a" adet nokta gözükür.

Not: Yukarıda verilen tüm seçenekler \psaxes komutundan hemen sonra [] köşeli parantezlerinin içine yazılmalıdır.

pspicture ortamında çizilen bir şekildeki çizgi kalınlığı, rengi, doldurma rengi, doldurma sitili, x ve y eksenindeki uzunluk ölçüsü gibi pek çok seçenek komutların içinde tek tek belirlenebilir. Bazen bir ortamda uygulanmak istenen seçeneği her komut için tek tek yazmak oldukça uzun olabilir. Bu gibi durumlarda istenilen seçeneği ortamın başında, ortam içinde yazılacak her komut için uygulanması sağlanabilir. Bunun için ortam başına \psset{opsiyonlar} komutu kullanılmalıdır. Örneğin:

\psset{fillcolor=blue, linecolor=red}

komutu ile artık o ortamda çizilecek her grafik için doldurma rengi mavi, çizgi rengi kırmızı olacaktır. Benzer şekilde

\psset{unit=k_1}, \psset{xunit=k_2}, \psset{yunit=k_3}

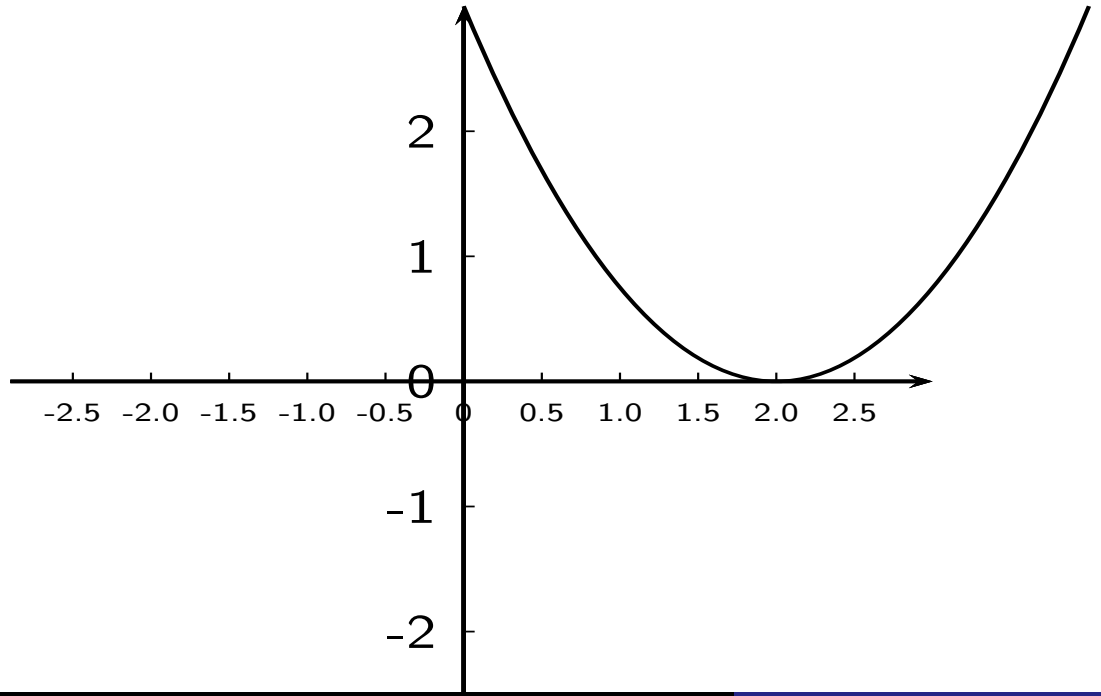
komutu ile sırasıyla tüm eksenlerin ölçüsü k kat, x eksenin ölçüsü k_2 kat ve y eksenin ölçüsü k_3 kat oranında değişir.

Burada dikkat edilecek husus bu komutlar ile belirlenen seçenekler kendinden sonra gelen her pstiricks ortamına etki etmektedir.

Dolayısıyla yapılan ayarların sonraki çizimlere etki etmesi istenmiyorsa ya yeni ortamların başında yeni ayarlar yapılmalı ya da psset komutu ve pstrikcs ortamı {} parantezleri içinde yazılmalıdır.

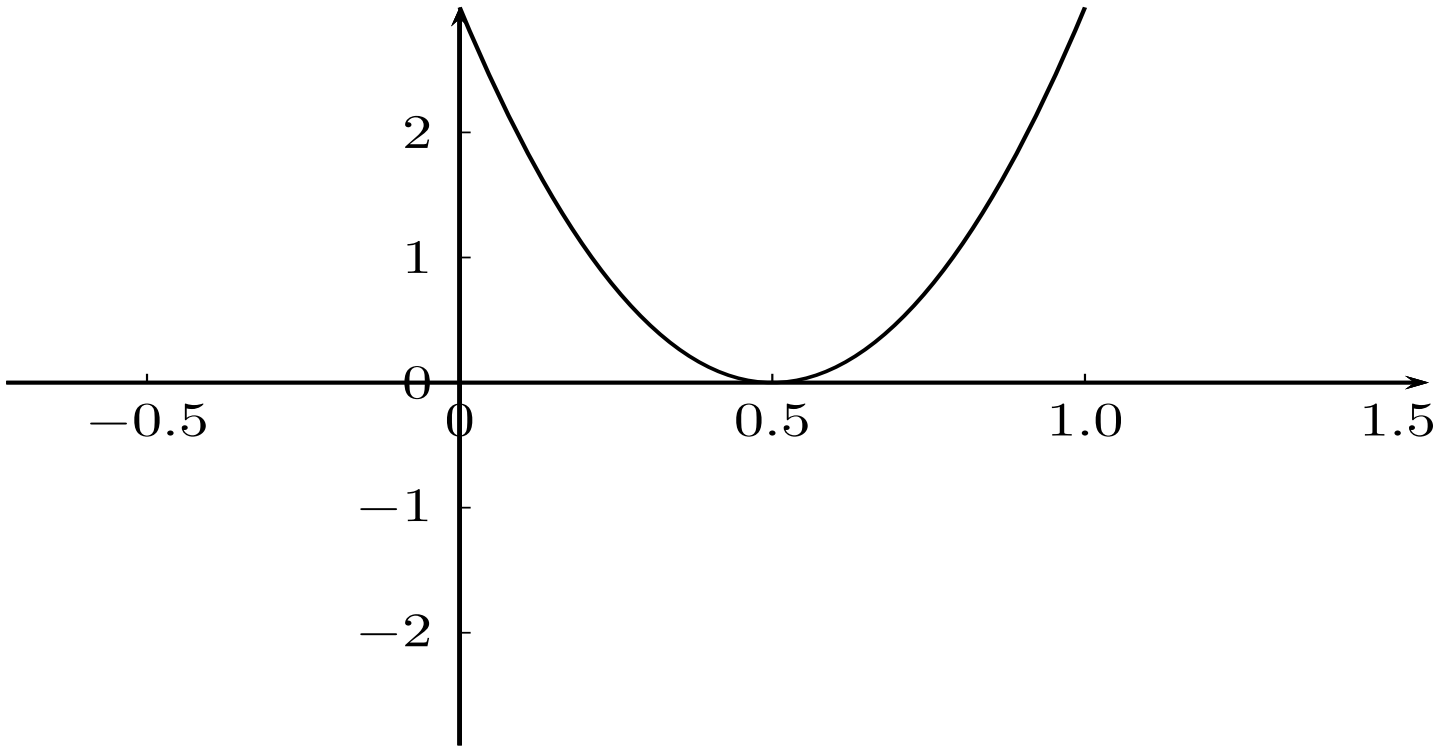
Şimdi yukarıda verilen seçenekleri gösteren bir örnek yapalım:

```
\begin{pspicture}(-3,-3)(3,3)
\psaxes[ticksize=5pt,Dx=0.5,Dy=1,mathLabel=false,
xlabelFontSize=\tiny]{->}(0,0)(-2.9,-2.9)(3,3)
\psparabola(0,3)(2,0)
\end{pspicture}
```



Yukarıdaki örnekte görüldüğü üzere"Dx=0.5" diyerek x ekseni boyunca artımı "0.5" birim olarak ayarladık. Şimdi aynı örnekte "dx" kullanımını inceleyelim.

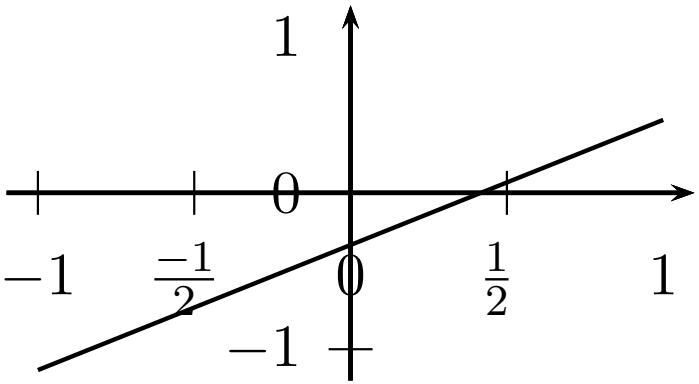
```
\begin{pspicture}(-3,-3)(6.25,3)
\psaxes[ticksize=5pt,Dx=0.5,Dy=1,dx=2]{->}(0,0)
(-2.9,-2.9)(6,3) \psparabola(0,3)(2,0)
\end{pspicture}
```



Yukarıdaki örnekten görüldüğü üzere"dx=2" alarak x eksenindeki noktaların arasını 2 birim olarak ayarlamış olduk. Dolayısıyla x ekseninin pozitif kısmı 6 birim olduğundan 6/2 den toplamda 3 adet noktanın değeri gösterilmiş ve "Dx=0.5" olmasından dolayı bunlar 0.5 birim artışa sahip olmuştur.

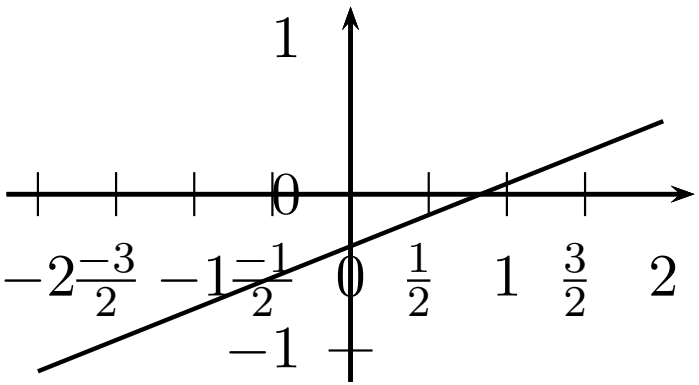
Şimdi eksenlerin nasıl kesirli olarak gösterileceğine dair bir örnek verelim:

```
\begin{pspicture}(-2,-1)(2,1)
\psaxes[xfractionLabels,xfractionLabelBase=2]
{->}(0,0)(-2,-1)(2,1)
\psplot[algebraic,plotpoints=100]
{-2}{2}{0.4*x-1/3} \end{pspicture}
```



Yukarıdaki örnekte görüldüğü üzere x eksenin aralığını $[-2, 2]$ olarak belirlememize rağmen elde ettiğimiz grafikte aralık $[-1, 1]$ şeklindedir. Bu kullandığımız xfractionLabelBase=2 komutunun eksen aralığını da 2 ye bölmesinden kaynaklıdır. Bunu düzeltmek için "dx" komutu ile noktalar arasındaki uzunluğu düzeltmemiz gereklidir. Eğer "dx=0.5" seçersek 2/0.5=4 den toplamda 4 adet 1/2 elde edeceğimiz için uygun aralıkları elde ederiz.

```
\begin{pspicture}(-2,-1)(2,1)
\psaxes[dx=0.5,xfractionLabels,xfractionLabelBase=2]
{->}(0,0)(-2,-1)(2,1)
\psplot[algebraic,plotpoints=50]
{-2}{2}{0.4*x-1/3} \end{pspicture}
```

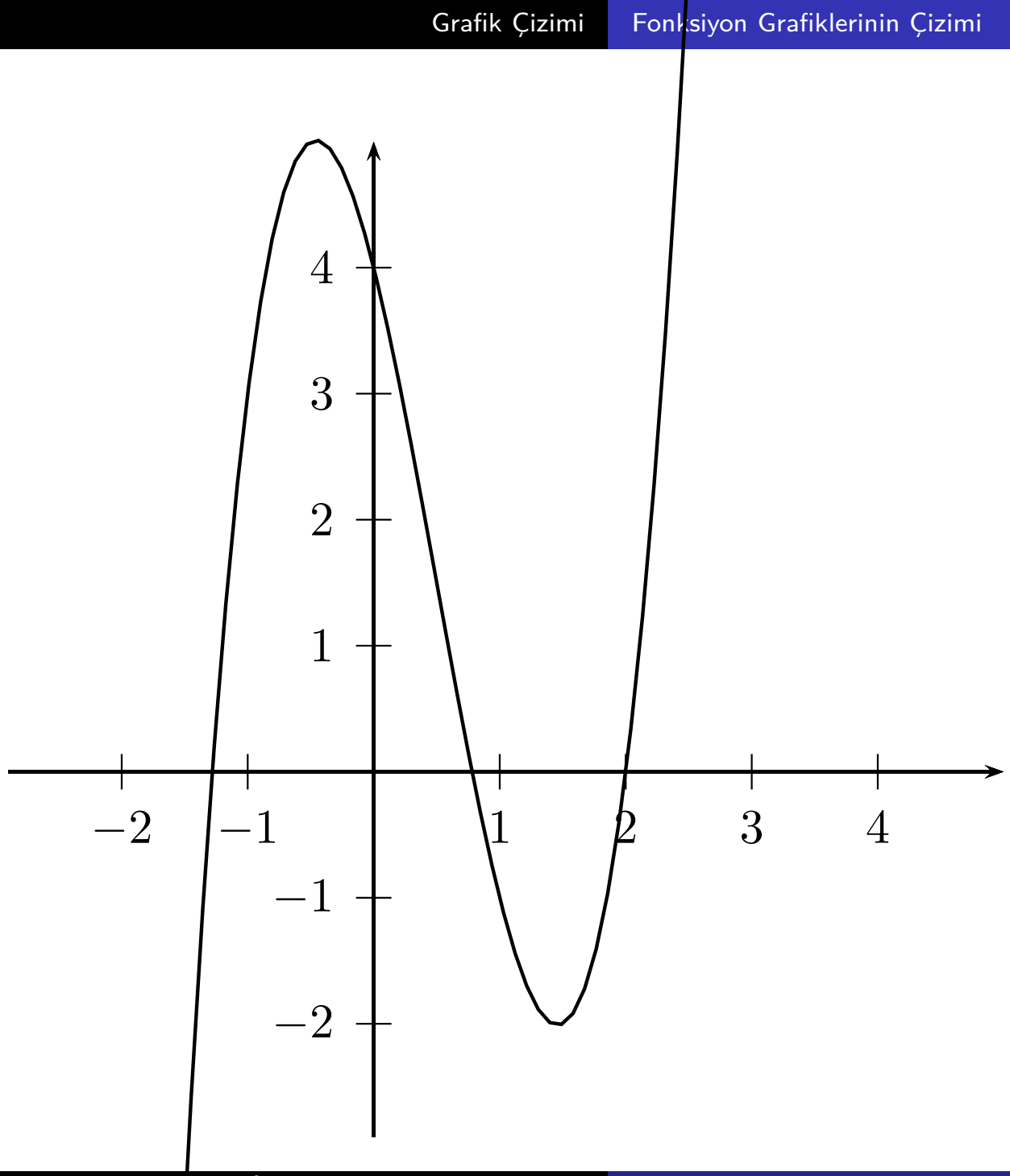


Şimdi pstricks ortamında bazı temel fonksiyon grafiklerinin nasıl çizildiğini görelim. Bunun için psplot komutu kullanılacaktır. Bu komut şu şekilde çalışır:

```
\psplot [opsiyon]{xmin}{xmaks}{fonksiyon}
```

Burada xmin ve xmaks grafiğin x eksenindeki aralıklarını belirler. fonksiyon kısmına ise grafiği çizilecek fonksiyon yazılır. [opsiyon] kısmına ise bu komut için tercih edilecek ekstra seçenekler gelecektir. psplot komutunda fonksiyonlar, PostScript kodu ile yazılmaktadır. Fakat bu kod ile yazım standarttın dışında olduğundan yeni başlayan biri için oldukça karmaşık gelebilir. Bu yüzden biz opsiyon kısmına algebraic seçeneğini ekleyerek fonksiyonları bize daha tanıdık gelen bir şekilde yazacağız. Şimdi bu komut yardımıyla bazı fonksiyonların grafiklerinin nasıl çizildiğine bakalım:

```
\begin{pspicture}(-3,-3)(5.5,5.5)
\psaxes[showorigin=false]{->}(0,0)(-2.9,-2.9)(5,5)
\psplot[algebraic]{-2}{2.5}{2*x^3-3*x^2-4*x+4}
\end{pspicture}
```


4
3
2
1
−1
−2
−2
−1
1
2
3
4

Yukarıdaki örnekte görüldüğü üzere çizdiğimiz grafik pstricks ortamına sığmayıp taşmıştır. Eğer çizeceğimiz grafiğin eksenlerde kalmasını istiyorsak diğer bir deyişle yalnızca eksenlerde kalan kısmını görmek istiyorsak ortamı pstricks* şeklinde kullanmalıyız.

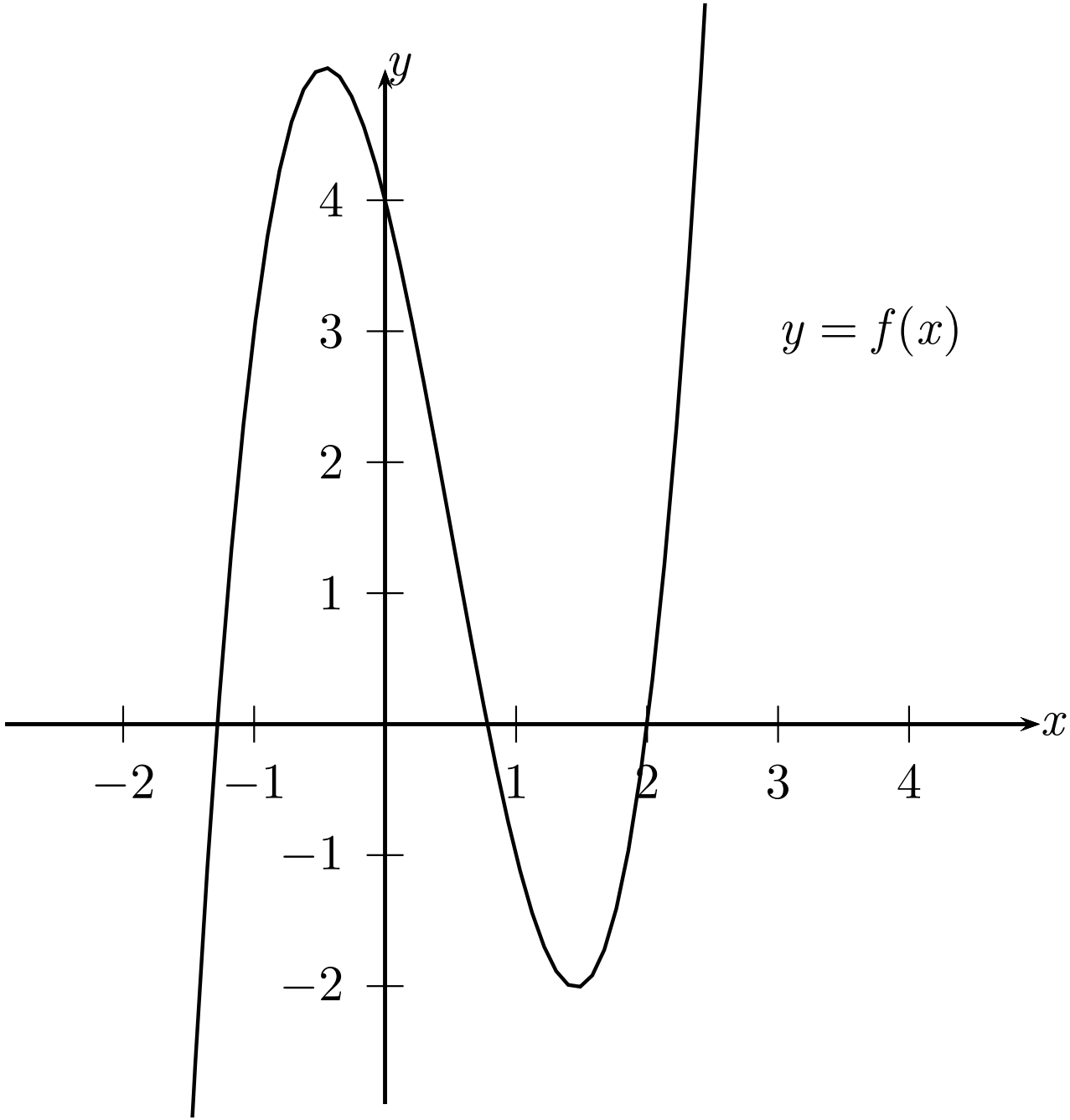
Not: psplot komutunun opsiyon kısmına eklenecek yMaxValue=ymax, yMinValue=ymin komutu ile grafiğin y eksenindeki aralığı belirlenebilir. Burada ymax grafiğin y eksenindeki maksimum değerini, ymin ise grafiğin y eksenindeki minimum değeri belirler.

Bir pstricks ortamına bir metin ya da matematiksel ifade eklemek istiyorsak

```
\rput[opsiyon](x,y){ifade}
```

komutunu kullanabiliriz. Burada (x,y) ifadenin ekleneceği konumu, "ifade" ise ne eklenmek istiyorsak onu ifade eder. opsiyon ise eklenecek ifadenin nasıl hizalanacağını belirtir. yatayda r sağa, l sola, düşeyde ise t yukarı, b aşağı ve B ise satır üzerine göre hizalar. Yukarıdaki örneği bu komutları kullanarak yeniden çizdirelim:

```
\begin{pspicture*}(-3,-3)(5.5,5.5)
\psaxes[showorigin=false]{->}(0,0)(-2.9,-2.9)(5,5)
\psplot[algebraic]{-2}{2.5}{2*x^3-3*x^2-4*x+4}
\rput[l](5,0){$x$} \rput[l](0,5){$y$}
\rput[l](3,3){$f(x)=2x^3-3x^2-4x+4$}
\end{pspicture*}
```


y
x
4
3
2
1
−1
−2
−2
−1
1
2
3
4
y = f(x)

psplot ile çizilecek bazı temel fonksiyonlar ve komutları şu şekildedir:

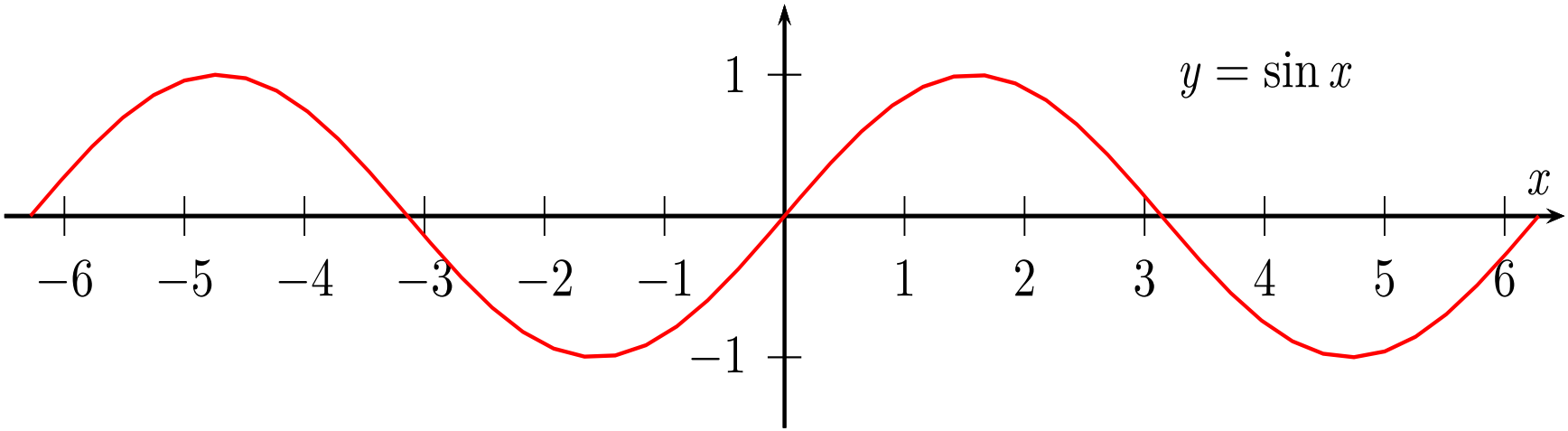
Fonksiyonlar ve komutları

Fonksiyon	Komut	Fonksiyon	Komut
$\sin x$	SIN(x)	$\sinh x$	SINH(x)
$\cos x$	COS(x)	$\cosh x$	COSH(x)
$\tan x$	TAN(x)	$\tanh x$	TANH(x)
$\cot x$	COS(x)/SIN(x)	$\operatorname{arcsinh} x$	ASINH(x)
$\arcsin x$	ASIN(x)	$\operatorname{arccosh} x$	ACOSH(x)
$\arccos x$	ACOS(x)	$\operatorname{arctanh} x$	ATANH(x)
$\arctan x$	ATAN(x)	$\exp x$	EXP(x)
$\operatorname{arccot} x$	1/ATAN(x)	$\sqrt{x}$	sqrt(x)
$\sec x$	1/COS(x)	$\ln x$	Log(x)
$\csc x$	1/SIN(x)	$\log_a x$	Log(x)/Log(a)
$\operatorname{arcsec} x$	ASEC(x)	$\frac{\sin x}{x}$	SINC(x)
$\operatorname{arccsc} x$	ACSC(x)	$\Gamma(x)$	GAMMA(x)

Not: Yukarıda verilen fonksiyonun tanım kümesindeki bir aralıkta grafikleri çizilebilir. Eğer x değerleri bu aralık dışında verilirse grafik çizdirilemez.

Şimdi yukarıda verilen fonksiyonlarla ilgili örnek grafikler çizdirelim:

```
\begin{pspicture*}(-6.5,-1.5)(6.5,1.5)
\psaxes[showorigin=false]{->}(0,0)(-6.5,-
1.5)(6.5,1.5)
\psplot[algebraic,linecolor=red]{-
6.283}{6.283}{SIN(x)}
\rput[l](5,0.15){$x$} \rput[l](0,5){$y$}
\rput[l](4,1){y=$\sin x$}
\end{pspicture*}
```

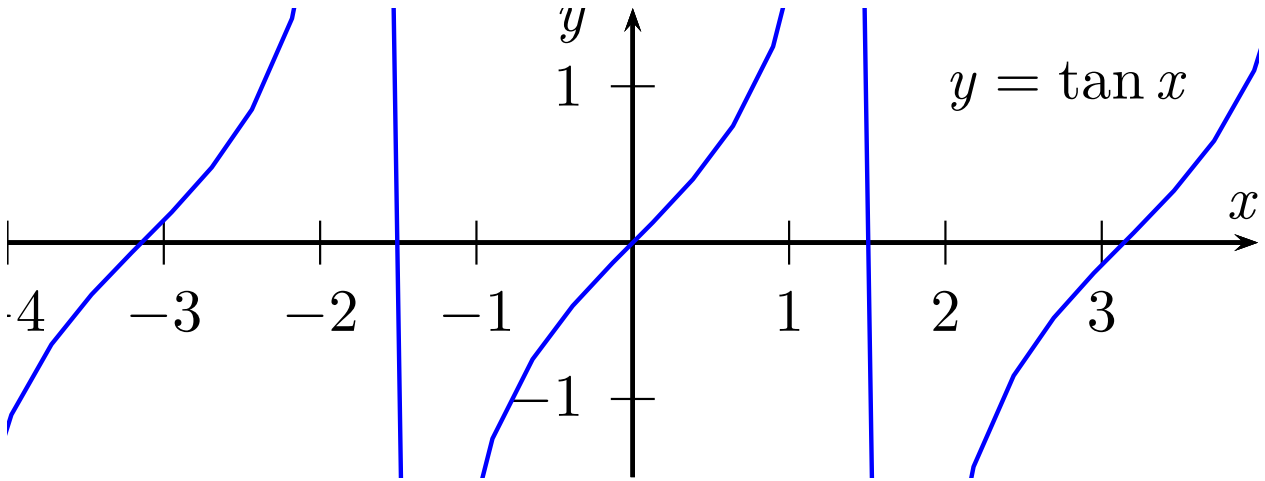


Yukarıdaki grafikte görüldüğü üzere $\sin x$ fonksiyonunun grafiği $[-2\pi, 2\pi]$ aralığında çizilmiştir. Fakat burada aralığın değerlerine 2π'nin sayısal karşılığı yazılmıştır. Eğer aralığı π cinsinden yazmak istersek, \pspciture ortamında π ve katlarının yazılabilmesi için bazı komutlar bulunmaktadır. Bunları şu şekilde sıralayabiliriz:

Komut	Değer
\psPi	π
\psPiTwo	2π
\psPiFour	4π
\psPiH	$\pi/2$

Komut	Değer
\pstPI1	π
\pstPI2	$\pi/2$
\pstPI3	$\pi/3$
\pstPI4	$\pi/4$

```
\begin{pspicture*}(-4,-1.5)(4,1.5)
\psaxes[showorigin=false]{->}(0,0)(-4,-
1.5)(4,1.5) \psplot[algebraic,color=blue]{-
\psPiTwo}{\psPiTwo}{TAN(X)}
\rput[l](3.9,0.15){$x$} \rput[l](-0.15,1.4){$y$}
\rput[l](2,1){y=$\tan x$}
\end{pspicture*}
```



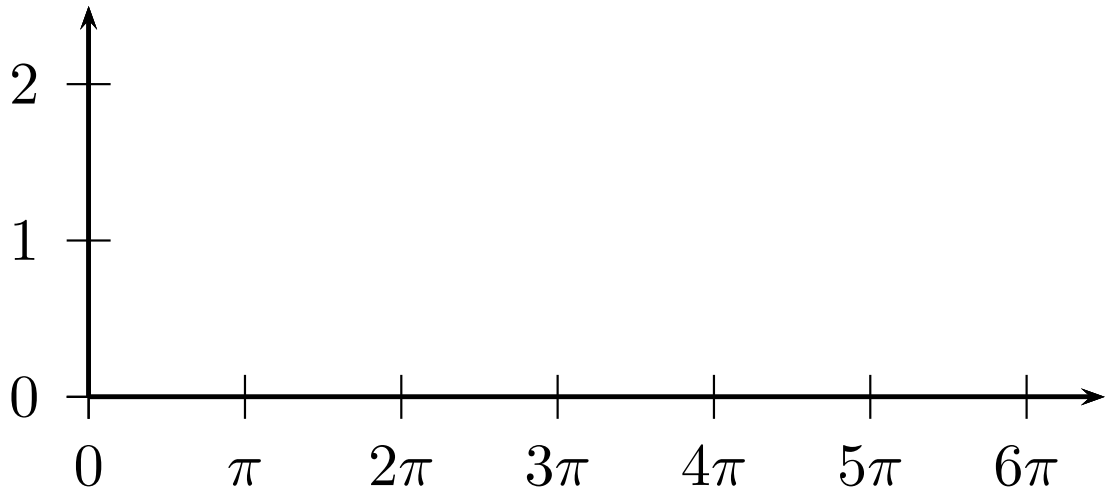
Not: Yukarıdaki grafikte grafik çizgilerinin yeterince düzgün olmadığı görülmektedir. Bu, grafik çiziminde yeterince noktanın kullanılmamasından kaynaklanmamaktadır. Opsiyonlara eklenecek plotpoints komutu ile çizim esnasındaki nokta sayısı artırılarak daha düzgün bir grafik elde edilebilir. Bu komut için varsayılan değer "50" dir.

Not: Trigonometric fonksiyonların grafikleri çizilirken x eksenindeki değerleri π cinsinden yazdırmak mümkündür. Bunun için kullanılacak komutlar:

trigLabels, xtrigLabels, ytrigLabels, trigLabelBase, xtrigLabelBase, and ytrigLabelBase

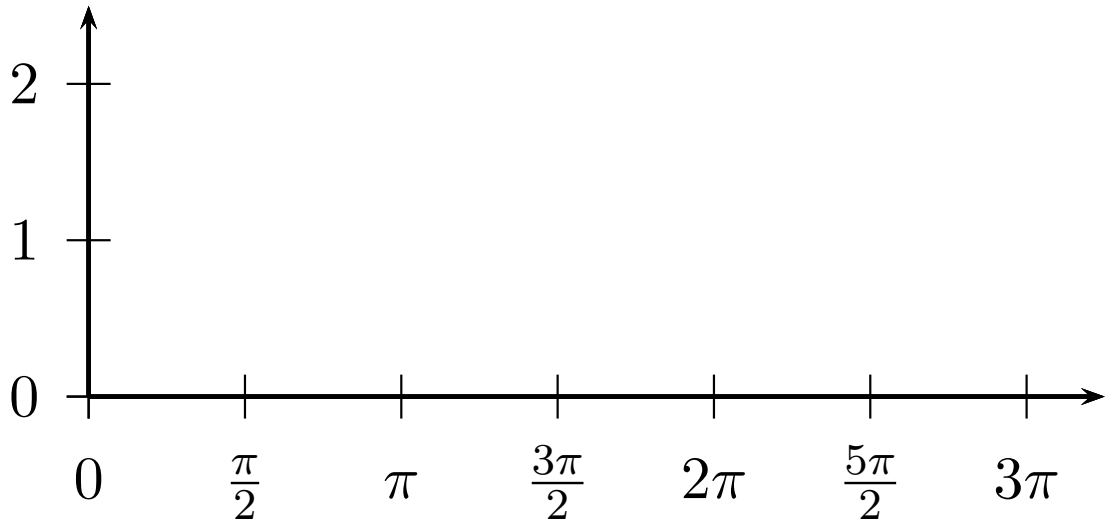
şeklindedir. Bu komutların kullanılması fractionLabel komutuyla aynıdır. Tek fark bu komutlar eksenleri π ye göre kesirli olacak şekilde yazdırmaktadır. Şimdi bu komutların nasıl kullanıldığını görelim:

```
\begin{pspicture}(-0.25,-1.25)(6.5,1.25)
\psaxes[xtrigLabels]{->}(-0.25,-1.25)
(6.25,1.25) \end{pspicture}
```



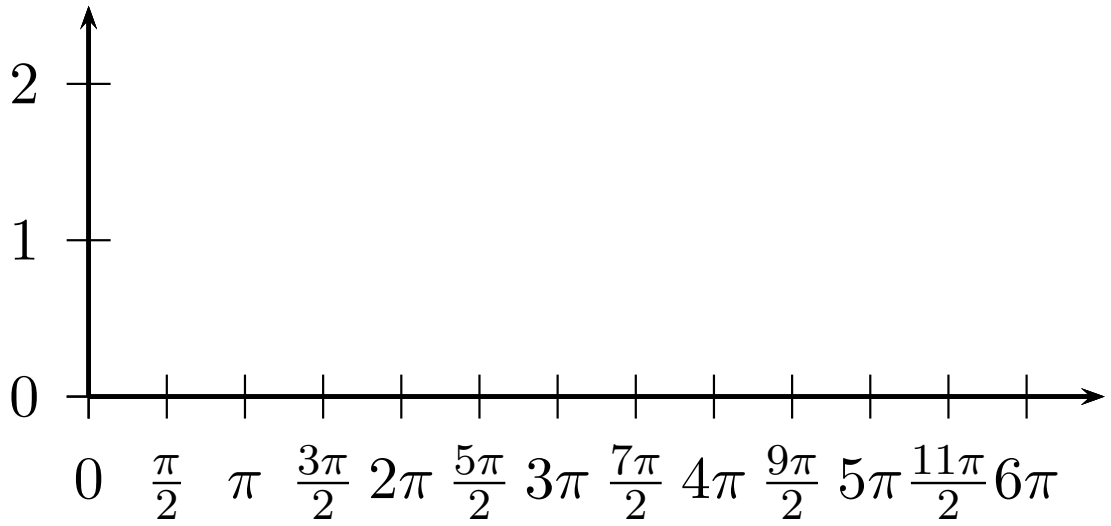
Görüldüğü üzere xtrigLabels opsiyonu koordinat değerlerini π cinsinden yazdırmamızı sağlamaktadır. Eğer bu durumda kesirli yazdırmak istersek:

```
\begin{pspicture}(-0.25,-1.25)(6.5,1.25)
\psaxes[xtrigLabels,xtrigLabelBase=2]{->}(-0.25,-
1.25)
(6.25,1.25) \end{pspicture}
```



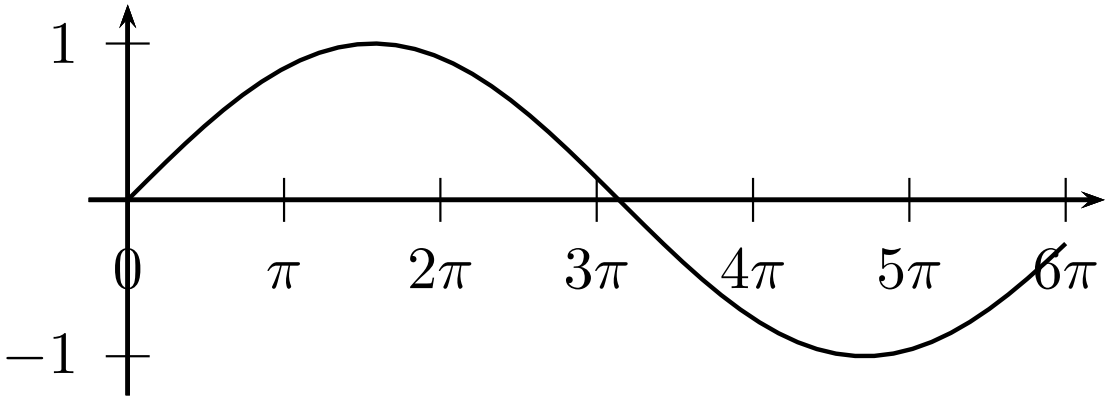
Yukarıda görüldüğü üzere xtrigLabelBase=2 almamızdan ötürü $\pi/2$ aralıklı değerler vermiş ve son değerimiz 3 olmuştur. Bunu olağan değeri olan 6 yapmak için dx=0.5 alırsak:

```
\begin{pspicture}(-0.25,-1.25)(6.5,1.25)
\psaxes[xtrigLabels,xtrigLabelBase=2,dx=0.5]{->}
(-0.25,-1.25)(6.25,1.25)
\end{pspicture}
```



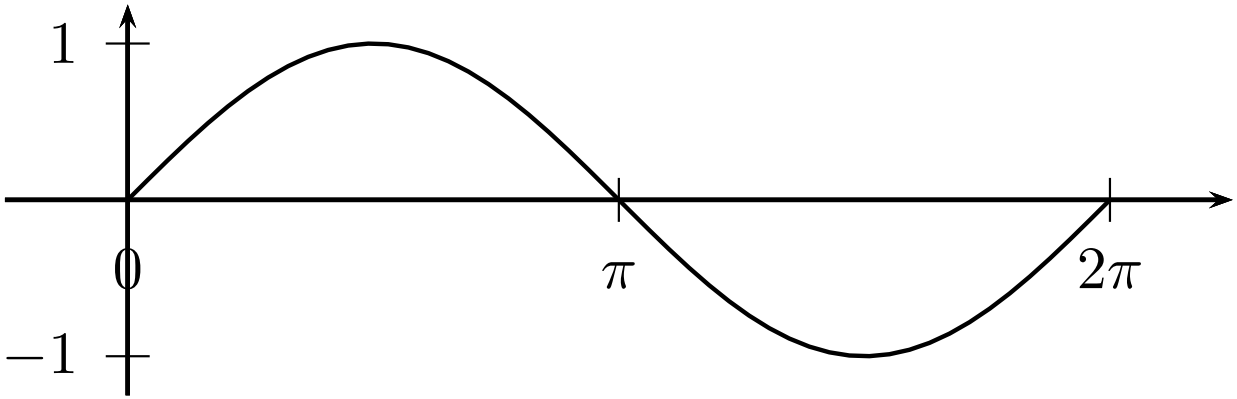
Şimdi koordinat eksenlerinin π cinsinden verildiği durumda trigonometrik fonksiyonların çiziminin nasıl doğru bir şekilde yapılacağını görelim:

```
\begin{pspicture}(-0.25,-1.25)(6.5,1.25)
\psaxes[xtrigLabels]{->}(-0.25,-1.25)
(6.25,1.25)
\psplot[algebraic]{0}{6}{SIN(x)}
\end{pspicture}
```



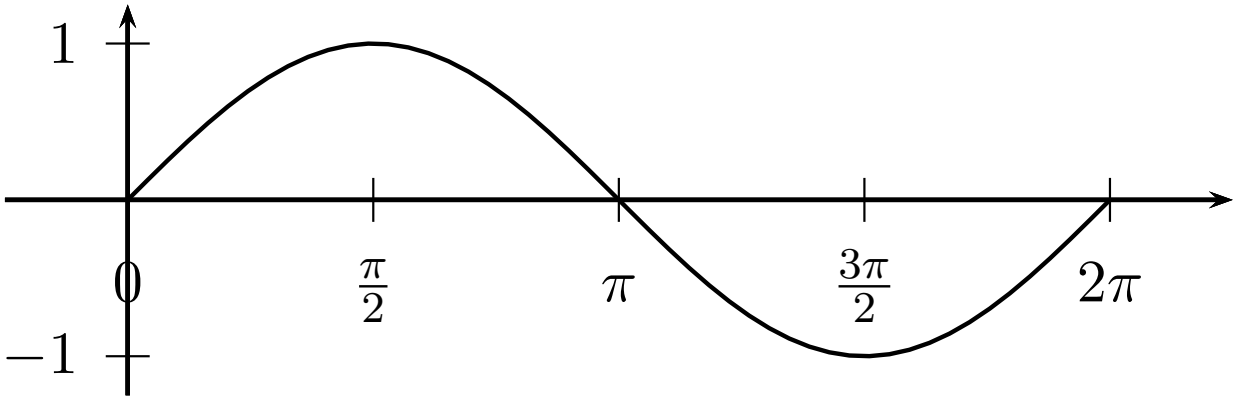
Yukarıdaki grafikten de görüldüğü üzere eksenler π cinsinden görülse de aslında bunlar (0,6) arasındaki reel sayılardır. Dolayısıyla sin fonksiyonunun (0,6) aralığındaki grafiği ile aynı grafiği elde etmiş olduk.

```
\begin{pspicture}(-0.25,-1.25)(2.5,1.25)
\psaxes[xtrigLabels,xunit=\psPi]{->}(-0.25,-1.25)
(2.25,1.25)
\psplot[algebraic]{0}{\psPiTwo}{SIN(x)}
\end{pspicture}
```



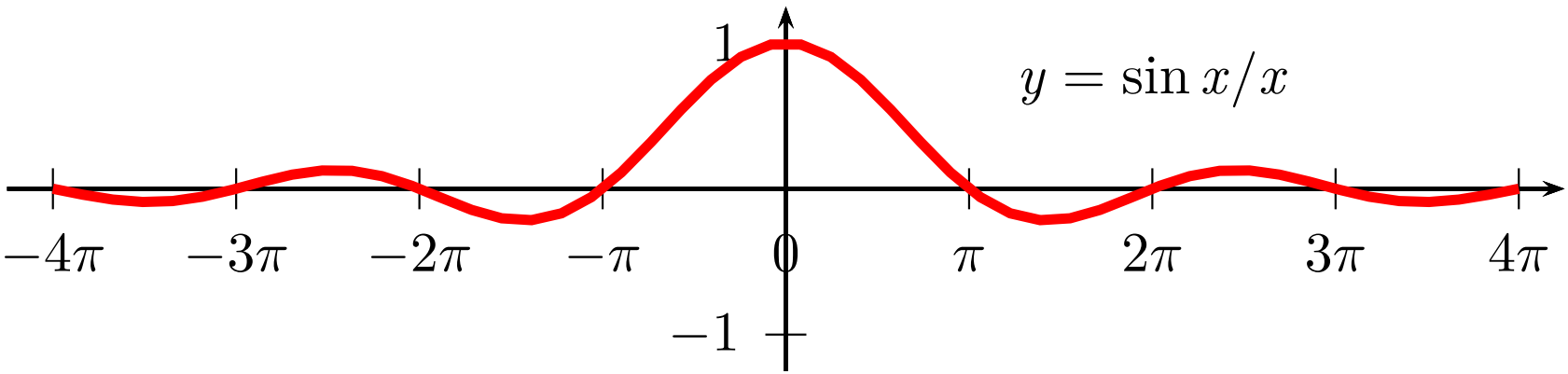
Yukarıdaki grafikten görüldüğü üzere sin fonksiyonunun $[0, 2\pi]$ aralığındaki grafiği elde edilmiştir. Bunun için xunit=\psPi seçeneği ile eksenler arası mesafenin π kadar olması sağlanmış daha sonra grafiğin bitim noktası \psPiTwo komutu ile 2π olarak belirtilmiştir.

```
\begin{pspicture}(-0.25,-1.25)(2.5,1.25)
\psaxes[xtrigLabels,xtrigLabelBase=2,dx=\pstPI2,
xunit=\psPi]{->}(-0.25,-1.25)(2.25,1.25)
\psplot[algebraic]{0}{\psPiTwo}{SIN(x)}
\end{pspicture}
```



Yukarıdaki grafikten görüldüğü üzere xtrigLabelBase=2 seçeneği ile x eksenindeki π değerleri 2'ye bölünmüş olarak yazılmıştır. Grafiğin bitim noktasının aynı kalması için reel değerdekine benzer şekilde dx=dx=\pstPI2 seçeneği ile x eksenindeki etiketlerin arasının $\pi/2$ olması sağlanmıştır.

```
\begin{center}
{\psset{xunit=.5cm}
\begin{pspicture}(-4.25,-1.25)(4.25,1.25)
\psaxes[xtrigLabels,xunit=\psPi]{->}(0,0)(-4.5,-
1.25)
(4.25,1.25)
\psplot[algebraic,linecolor=red,linewidth=2pt]
{-\psPiFour} {\psPiFour}{SINC(x)}
\rput(\psPiTwo,0.75){$y=\sin x/x$}
\end{pspicture}}
\end{center}
```



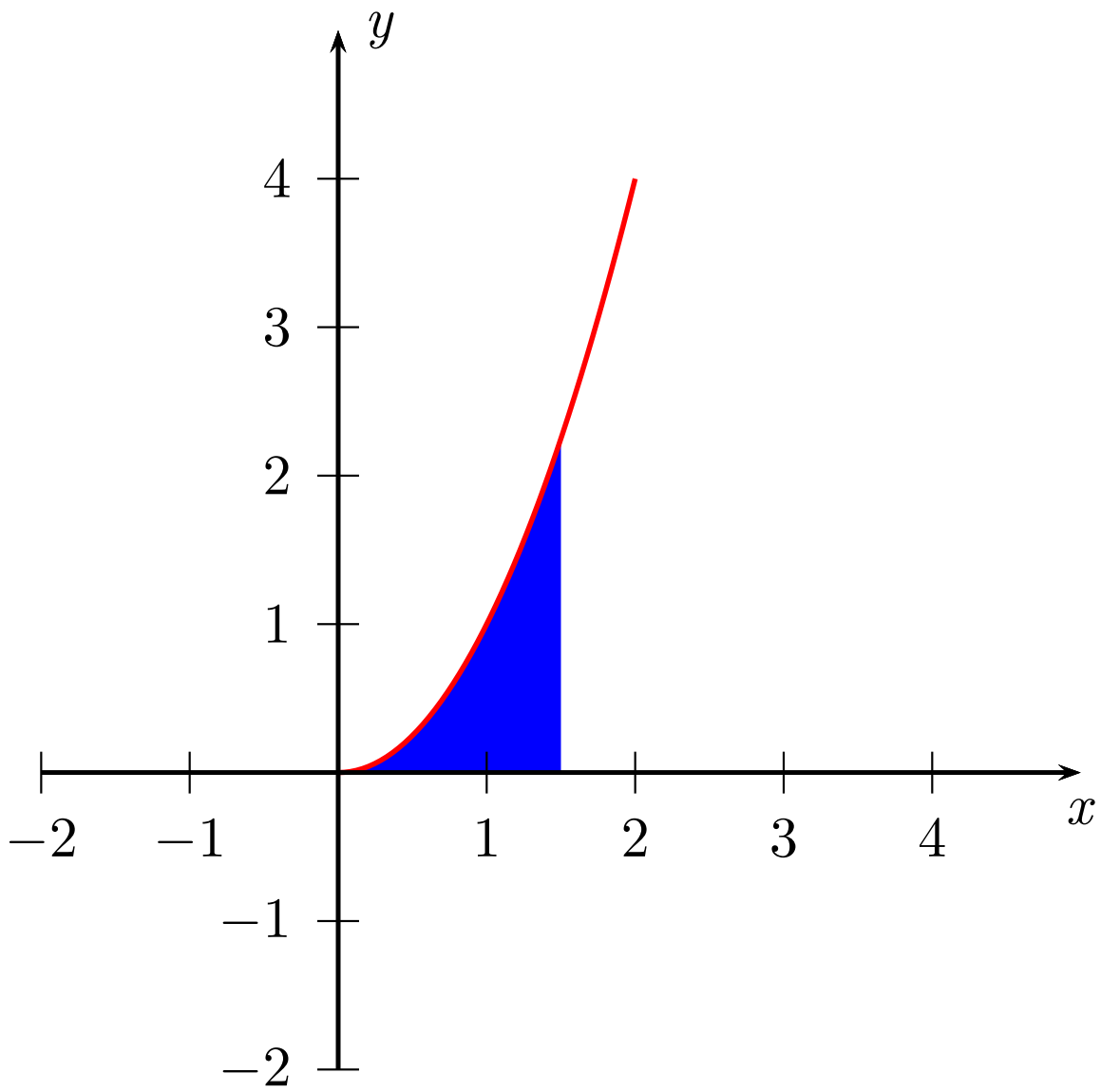
Eğriler arasını doldurma

pspicture ortamında çizilen bir eğrinin altında kalan alanı ya da iki eğrinin arasında alanı doldurmak için

\pscustom[opsiyon]{...}

komutu kullanılır. Burada {...} kısmına arası doldurulmak istenen eğriler ile ilgili komutlar girilir. Şimdi bu komutun nasıl kullanıldığına dair bir iki örnek verelim:

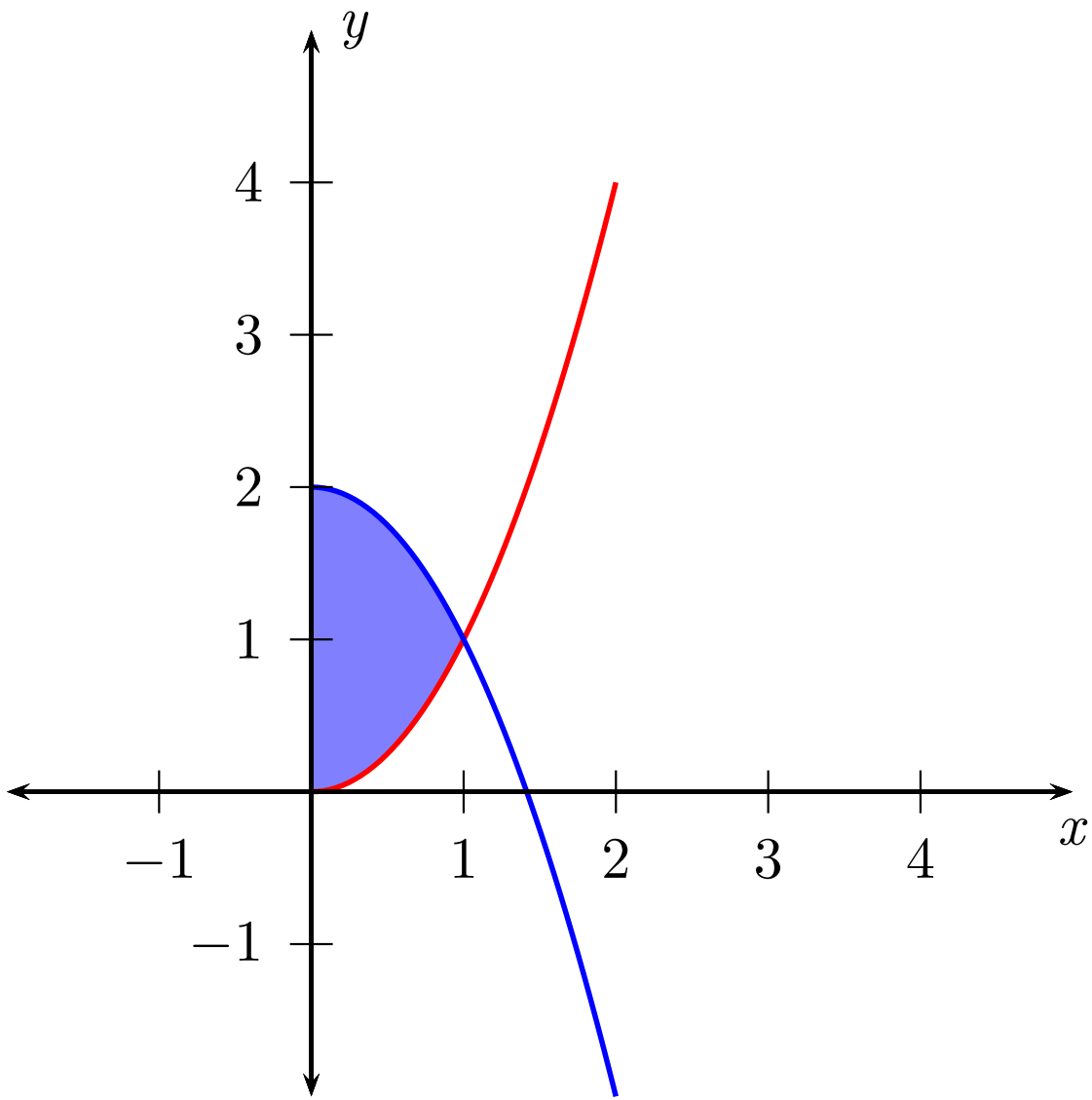
```
\begin{center}
\begin{pspicture}(-3,-5)(6,6)
\pscustom[fillstyle=solid,fillcolor=blue,linestyle=none,
algebraic]{
\psplot{0}{1.5}{x^2}
\psline(0,0)(*0 {0})
\psline(*1.5 {1.5^2})(1.5,0)}
\psaxes[showorigin=false]{->}(0,0)(-2,-2)(5,5)
\psplot[linewidth=1pt,algebraic,linecolor=red]{0}{2}{x^2}
\end{pspicture}
\end{center}
```


y
x
4
3
2
1
−1
−2
−2
−1
1
2
3
4

Yukarıdaki komut ile $y = x^2$ parabolü ile $y = 0$ doğrusu yani x ekseninin arasında $(0, 0)$'dan $(1.5, 0)$'a kadar kalan alan doldurulmuştur. Bunun için ilk olarak \psplot{0}{1.5}{x^2} komutu, $y = x^2$ parabolü $(1.5, 0)$ aralığında bir üst sınır olarak belirlenirken, diğer sınırları (0,0) ile bu noktanın fonksiyondaki değerine karşılık gelen kısmı olan (*0 {0}) noktasını belirten \psline(0,0)(*0 {0}) komutu ve fonksiyonun (*1 {1.5^2}) değeri ile bu değere karşılık gelen (1.5,0) noktayı belirten \psline(*1 {1.5^2})(1.5,0)} komutu ile belirlenmiştir. Burada dikkat edilmesi gereken husus, sınırlar için ilk koordinat nokta ve fonksiyonun bu noktadaki değer ile belirlenirken diğer koordinat fonksiyonun değeri ve bu değeri veren nokta şeklinde belirlenmektedir. Eğer sınır koordinatları bu şekilde verilmezse uygun alan doldurulamaz.

Şimdi iki eğri arasında kalan alanın nasıl doldurulduğunu görelim:

```
\begin{center}
\begin{pspicture}(-3,-5)(6,6)
\pscustom[fillstyle=solid,fillcolor=blue!50,
linestyle=none,algebraic]{
\psplot{0}{1}{x^2}
\psplot{1}{0}{2-x^2}}
\psaxes[showorigin=false]{->}(0,0)(-2,-2)(5,5)
\psplot[linewidth=2pt,algebraic,linecolor=red]{0}{2}{x^2}
\end{pspicture}
\end{center}
```


y
x
4
3
2
1
−1
−1
1
2
3
4

Son örnekte görüleceği üzere boyanmak istenen bölgenin sınırları \psplot{0}{1}{x^2} ve \plot{1}{0}{2-x^2} komutları ile belirlenmiştir. Burada doldurulacak olan bölgeyi belirlemek için yine ilk komut ile fonksiyon 0'dan 1'e çizdiriliyor iken ikinci komut ile fonksiyon 1'dan 0'e çizdirilmektedir.

LaTeX'in Özelleştirilmesi

LaTeX yeni komutlar, ortamlar ve paketler tanımlanmasına olanak sağladığı gibi kendi içinde bulunan komut ve ortamlarında yeniden tanımlanmasına da izin verir. Önceki kısımlar yeni bir ortamın nasıl tanımlanacağını görmüştük. Şimdi yeni bir komut ve var olan bir komutun yeniden nasıl tanımlanacağını görelim.

Yeni Komut Tanımlama

LaTeX'de yeni bir komut tanımlamak için
\newcommand{\ad}[sayı]{tanım}

komutu kullanılır. Bu komut preamble kısmına yazılmalıdır. Burada ad kısmına komuta vermek istediğimiz ad, tanım kısmına ise bu komutun nasıl tanımlandığı yani işlevinin ne olduğu yazılır. sayı kısmı ise komutun argümanını belirtir ve bu 0'dan 9'a kadar bir değer alır. Eğer bir değer girilmemişse 0 demektir. Şimdi bir iki örnek ile bunu açıklayalım.

Preamble kısmına \newcommand{\ld}{2021-2022 Güz Dönemi \LaTeX{} dersi} ile yeni bir komut tanımlayalım. Eğer kaynak dosya içerisinde bu komut aşağıdaki gibi kullanılırsa yandaki sonuç elde edilir.

```
\ld{} boyunca \LaTeX{}
programının ne olduğu ne
amaçla kullanıldığı ve
matematiksel ifadelerin
nasıl yazıldığı
öğretilmiştir.
```

2021-2022 Güz Dönemi LaTeX dersi boyunca LaTeX programının ne olduğu ne amaçla kullanıldığı ve matematiksel ifadelerin nasıl yazıldığı öğretilmiştir.

Görüldüğü üzere tanımlamış olduğumuz bu komut sayesinde uzun bir metni yazmak yerine ona atadığımız kısa bir komutu kullanarak aynı şeyi yazdırabiliyoruz. Bu da sıklıkla kullanılan uzun ifadelerin yazımında oldukça büyük bir kolaylık sağlamaktadır. Yukarıdaki örnekte görüldüğü üzer \ld komutunda sonra boşluk bırakmamıza rağmen çıktıda komuttan sonra gelen ilk kelime birleşik olarak yazılmıştır. Bu durum \LaTeX komutundan sonra boşluk olmaması gibi komuttan hemen sonra {} eklenerek çözülebilir.

Yeni komut tanımlama bazı matematiksel sembollerin yazılmasında da oldukça kullanışlıdır. En basit şekilde reel sayılar, kompleks sayılar ve doğal sayılar kümesini göstermek için kullanılan süslü notasyonları elde etmek için sırasıyla \mathbb{R}, \mathbb{C} ve \mathbb{N} komutları kullanılmalıdır. Fakat preamble'a eklenecek olan

```
\newcommand{\R}{\mathbb{R}}
\newcommand{\C}{\mathbb{C}}
\newcommand{\N}{\mathbb{N}}
```

komutları ile bu semboller daha pratik olarak yazdırılabilir. Örneğin:

```
$\R$, $\C$, $\N$
```

$\mathbb{R}$, $\mathbb{C}$, $\mathbb{N}$

Şimdi yeni tanımlanan komutta argümanın nasıl çalıştığını görelim. Preamble kısmına

```
\newcommand{\yil}[1]{\emph{#1}Güz Dönemi \LaTeX{} dersi}
```

ekleyelim. Eğer kaynak dosyada aşağıdaki şekilde kullanılırsa elde edilecek sonuçlar yandaki gibi olacaktır.

```
\yil{2022-2023}{} ve
\yil{2023-2024}{}
bölümümüz yüksek
lisans dersi olarak
verilecektir.
```

2022-2023 Güz Dönemi LaTeX dersi ve *2023-2024* Güz Dönemi LaTeX dersi bölümümüz yüksek lisans dersi olarak verilecektir.

Görüldüğü üzere komutun argümanına yazılan ifade \emph komutu ile vurgulanmıştır. Burada dikkat edilmesi gereken noktalardan biri tanımlanacak komutun adının Türkçe karakter içermemesidir.

Eğer birden fazla argümanlı bir komut tanımlamak istiyorsak komut içindeki sayıyı argüman sayısı ile değiştirmemiz gerekir. Örneğin:

\newcommand{\yd}[2]{\emph{#1} \textbf{#2} Dönemi \LaTeX{} dersi}

şeklinde iki argümanlı bir komut tanımlayalım. Bu komutu aşağıdaki gibi kullanırsak yandaki çıktıyı verecektir:

```
\yd{2022-2023}{Güz}{} ve
\yil{2023-2024}{Bahar}{}
bölümümüz yüksek
lisans dersi olarak
verilecektir.
```

2022-2023 **Güz** LaTeX dersi ve *2023-2024* **Bahar** LaTeX dersi bölümümüz yüksek lisans dersi olarak verilecektir.

Bu örnekte görüldüğü üzere ikinci argüman komuta ikinci bir küme parantezi ile eklenmiştir ve komutta #2'nin yazıldığı yere argümana girilen ifade eklenmektedir.

Argümanlı komut tanımlama matematiksel ifadelerin yazımında da oldukça büyük bir kolaylık sağlamaktadır. Buna basit bir örnek verelim:

```
\newcommand{\set}[1]{\left{ #1\right}}
```

komutu ile artık argümana yazacağımı her ifade küme parantezleri arasına yazılacaktır. Örneğin:

```
Doğal sayılar kümesi
$\N=\set{1,2,\cdots}$
şeklindedir.
```

Doğal sayılar kümesi
$\mathbb{N} = \{1, 2, \cdots\}$ şeklindedir.

Mevcut Komutu Düzenleme

LaTeX'in kendi komutları ile aynı ada sahip yeni bir komut tanımlayamayız. Fakat mevcut komutları kendi istediğimiz şekilde yeniden tanımlamamız mümkündür. Bunun için,

\renewcommand{\ad}[sayı]{tanım}

şeklinde newcommand komutunun kullanımına oldukça benzerdir.

Bu komut genellikle LaTeX'de tanımlanan içindekiler, tablolar, şekiller gibi macroların isimlerini değiştirmek için kullanılır. Bunun için:

\renewcommand{\makro adı}{vermek istenilen ad}

komutu preamble'a eklenmelidir. Örneğin:

```
\renewcommand{\figurename}{görseller}
```

komutu kullanılırsa artık babel paketi kullanmıyorsak figure adı turkish opsiyonu ile babel paketi kullanıyorsak da şekil adı artık görseller olarak değişecektir.

Yukarıda verilen komut babel paketi ile farklı bir dil yüklendiğinde çalışmaz. Eğer turkish opsiyonu ile babel paketi yüklenmişse

```
\addto\captionsturkish{
\renewcommand{\macro adı}{vermek istenilen ad}}
```

komutu kullanılmalıdır.

Aşağıda verilen tabloda yukarıdaki komutta kullanılabilecek makro adları ve karşılıkları verilmiştir:

Makro adı	Karşılığı
\abstractname	Abstract ortamı
\chaptername	Bölüm (chapter)
\contentsname	İçindekiler tablosu
\listfigurename	Şekiller listesi
\listtablename	Tablolar listesi
\figurename	Şekil adı
\tablename	Tablo adı
\apendixname	Ekler
\indexname	Dizin
\refname	Kaynaklar (article sınıfında)
\bibname	Kaynaklar (book ve report sınıfında)

Slayt Hazırlamak

Matematiksel ifadeleri içeren bir sunum dosyası hazırlamak, matematik ifadelerin yazımı ve diziliminin kullanılan sunum hazırlama programında kolayca gerçekleştirilememesinden dolayı bazen oldukça zor olmaktadır. LaTeX standart bir dokumanda matematiksel ifadelerin yazımında ve diziminde sağladığı büyük kolaylığı sunum hazırlamada da sağlamaktadır. Bu kısımda LaTeX kullanarak bir sunum dosyasının nasıl hazırlanacağına dair bazı temel bilgiler verilecektir. Bilinmelidir ki burada bahsedilmeyen daha pek çok özellik mevcuttur. Bunlar ihtiyaç duyulması halinde gerekli paketlerin içerikleri veya başka kaynaklar okunarak öğrenilebilir.

Beamer

- LaTeX ile sunum dosyası hazırlamak için beamer dokuman sınıfı kullanılacaktır.
- Hazırlanacak her slayt sayfası \begin{frame}...\end{frame} arasına girilmelidir.
- Bir frame içindeki girdileri sayfanın üstüne hizalamak için \begin{frame}[t], aşağısına hizalamak için ise \begin{frame}[b] şeklinde opsiyonlar eklenmelidir. Bunun varsayılanı sayfanın ortası olduğundan [c] opsiyonun kullanılması bir değişiklik yapmaz. Eğer bunu her frame için değil de bütün sunumdaki frameler için yapmak istiyorsak tek tek yapmak yerine \documentclass[t]{beamer} şeklinde dokuman sınıfından önce belirmeliyiz.
- Bir frame'e \frametitle{başlık} komutu ile başlık atabiliriz. Bunun diğer biri yolu ise \begin{frame}{başlık} şeklinde yazmaktır.

- Bir sunumun başlangıç sayfasının sunum ve sunan kişi hakkında gerekli bilgilerin olduğu sayfa olması gerekmektedir. Bu sayfa için ilk oluşturulacak ilk frame içine aşağıdaki komutlar ile ilgili bilgiler girilmelidir. Yani:

```
\begin{frame}
\title{Sunumun başlığı}
\author{Sunumun yazarları (her yazar \and ile
ayrılır)}
\institute{Yazarların çalıştığı kurum}
\date{Sunumun tarihi veya sunumun yapıldığı
organizasyon adı}
\titlepage
\end{frame}
```

Burada son komut olan \titlepage, \maketitle komutuna benzer şekilde yazılan bu bilgilerin yazdırılmasını sağlar.

- Başlangıç sayfası yapmanın diğer bir yolu ise yukarıda yazılan komutlar ile verilen bilgilerin preamble'a yani \begin{document}'dan önce yazılarak ilk frame'i

```
\begin{frame}
\maketitle
\end{frame}
```

olarak yazmak. Başlangıç sayfası bu iki durumda da birebir aynı şekilde türetilir. Buradaki tek fark ikinci durumda sunumla ile ilgili bilgiler (başlık, yazar adı, vs) kullanılan temaya bağlı olarak sayfaların alt ya da üst kısmında yazdırılmasına karşın ilk durumda böyle bir durum söz konusu değildir.

Yukarıda da bahsedildiği üzere ikinci durumda oluşturulan başlangıç sayfasındaki bilgiler beamer da kullanılan temaya bağlı olarak her sayfanın altında veya üstünde bir bilgi olarak sunulur. Fakat, bazen verilen bu bilgiler çok uzun olduğundan bu kısımlara sığmaz bu gibi durumlar yukarıdaki komutlardan hemen sonra[kısa ad] şeklinde bu ifadelerin kısa bir adı yazılabilir. Örneğin \title[kısa ad]{tam ad} gibi.

- beamer da kullanılabilecek pek çok tema vardır. İstenilen tema \usetheme{tema adı} komutu ile yüklenir. Kullanılan her temaya göre başlıkların dizilimi, sayfa alt üst bilgisi, ortam renkleri gibi pek çok özellik değişmektedir. https://mpetroff.net/files/beamer-theme-matrix/ internet sayfasından bu temalar incelenebilir.
- Kullanılan her temanın kendine özgü renkleri mevcuttur. Eğer kullanılan temanın kendine özgü sitilinden memnun fakat kullanılmış olan renkleri beğenmediysek bunları \usecolortheme{renk teması} komutu ile değiştirebiliriz. Burada kullanılabilecek renk temaları yine yukarıda verilen internet sayfasında sunulmuştur.

- beamer'daki temaların renklerini kendi tanımlamış olduğumuz renkler ile de değiştirmek mümkündür. Bunun için ilk olarak \definecolor{ad}{rgb}{x,y,z} komutu ile rgb kodları ile tanımlanmış renge istediğimiz bir ad veririz. Daha sonra \usecolortheme[named=ad]{structure} komutu ile tanımlamış olduğumuz bu rengi temanın rengi olarak ayarlayabiliriz. LaTeX'de kullanılabilecek renkleri ve bunların rgb kodlarını http://latexcolor.com internet sayfasından görebiliriz.

Böylece beamer'da bir sunum dosyasın bazı temel ayarlarının nasıl yapıldığını görmüş olduk. Şimdi sunum içeriğinin hazırlanması için kullanılacak bazı komut ver ortamlardan bahsedebiliriz.

- LaTeX'de kullanılan her ortam beamer'da da kullanılabilir. Ayrıca \newtheorem komutu ile bir ortam tanımlama işlemi beamer için de mümkündür. Örneğin: hatırlarsanız \newtheorem{tnm}{Tanım} komutu ile bir tanım ortamı oluşturmuştuk. Benzer şeyi beamer içinde yaparak bir tanım ortamı oluşturabilir ve böylece istenilen bir frame içerisinde gerekli tanımları verebiliriz.
- Beamer'da ortam olarak kullanılabilecek bir diğer komut block ortamıdır. Bu \begin{block}...\end{block} şeklinde yazılıp bu ortama yazılan şeyler ayrıca vurgulanmış olur. Ayrıca block ortamına başlık atmakta mümkündür. Bunun için \begin{block}{başlık adı} şeklinde yazılması gerekir.
- Sunum dosyasından bir kelime ya da cümleyi vurgulamak için \alert{...} komutu kullanılabilir. Bu komut içeriğine yazılan ifade kullanılan temaya göre değişiklik göstermek üzere farklı bir renk ile vurgulanır.

- İstenilen bir frame'i birden fazla sütuna ayırmak mümkündür. Bunun için \begin{columns}...\end{columns} ortamı kullanılır. Bu ortam içerisine girilen her \column{genişlik} komutu sayfayı bir sütuna ayırır. Örneğin 3 adet \column{genişlik} komutu ile sayfa üç sütuna ayrılmış olur. Burada genişlik kısmı bu sütunların genişliğini vurgulamaktadır. Bu komutlardan sonra girilecek her ifade bir sonraki komuta kadar bir sütun olarak dizilir. Bunun bir diğer yolu ise her bit sütuna yazdırılacak ifadeyi \begin{column}{genişlik}...\end{column} arasına yazmaktır.
- Bir frame'i istenilen sayıda sütuna ayırmanın bir diğer yolu da multicol paketi ile gelen \begin{multicols}{sayı}...\end{multicols} komutudur. Eğer frame'de sütunlara ayrılmak istenilen ifadeler bu komut içerisine yazılmalıdır. Böylece bu girdiler sayı kısmında belirtilen değer kadar sütuna ayrılarak yazdırılır.

- Hazırlanan bir sunumda bazen sayfadaki her girdinin bir seferde gözükmesi istenmeyebilir. Eğer bazı girdilerin sırayla gelmesi isteniyorsa bunun için \pause komutu kullanılmalıdır. Bu komut, ilk gözükmesi istenen girdiden sonra konulursa bu komuttan sonra \pause komutu kendinden sonraki girdileri yeni bir slaytmış gibi gösterir. Tabii bu durum oluşturulan sunum dosyasının sayfa sayısını artırır. Eğer sunumun sayfa sayısını gerçek değerine indirgemek istersek \pause komutlarını tek tek kaldırmak yerine documentclass'a handout opsiyonu ekleyerek dosyayı standart hale getirebiliriz.
- Klasik bir LaTeX dosyasının hazırlanmasında olduğu gibi beamer ile hazırlanan bir sunum dosyasına da section, subsection,... komutları ile başlıklar eklenebilir. Bu başlıklar sunumun akışını nasıl ilerlediğini göstermektedir. Yani kullanılan temaya bağlı olarak frame'in hangi başlığın içeriğinde olduğu görülebilmektedir.

Kaynaklar

[1] Knuth, D. E., & Bibby, D. The texbook (Vol. 15). Reading: Addison-Wesley, 1984.

[2] Lamport, L. LaTeX: a document preparation system, Addison Wesley, Massachusetts, 2nd edition, 1994.

[3] Kopka, H., & Daly, P. W. A Guide to LATEX 2, Document Preparation for Beginners and Advanced Users, 1999.

[4] Oetiker, T., Partl, H., Hyna, I., Schlegl, E. İnce bir LATEX2ε Elkitabı (Türkçe Çeviri: Bekir Karaoğlu), 2006.

[5] Kottwitz, S. LaTeX beginner's guide. Packt Publishing Ltd, 2011.

[6] Kottwitz, S. LaTeX Cookbook. Packt Publishing Ltd, 2015.

[7] https://www.ctan.org