

# **LaTeX Dersleri**

Zafer Acar

2022-03-01



# İçindekiler

## Önsöz

ix

## 1 Genel

1

1.1	LaTeX Nedir? . . . . .	1
1.2	Önemli Yapılar . . . . .	2
1.2.1	Komutlar . . . . .	2
1.2.2	Paketler . . . . .	2
1.2.3	Ortamlar . . . . .	3
1.2.4	Gruplar . . . . .	3
1.2.5	Boşluklar . . . . .	3
1.2.6	Özel amaçlı karakterler . . . . .	4
1.3	Kurulum . . . . .	4
1.3.1	GNU/Linux . . . . .	5
1.3.2	Mac OS . . . . .	5
1.3.3	Windows . . . . .	5
1.3.4	LaTeX Editörleri . . . . .	6
1.3.5	Çevrimiçi Editörler . . . . .	6
1.4	Tipik Bir Belge Yazımı . . . . .	6
1.5	Belge Sınıfları ve Seçenekleri . . . . .	7
1.6	Türkçe Dil Ayarları ve Çoklu Dil Kullanımı . . . . .	9
1.7	Heceleme . . . . .	10
1.8	Belgeye Başlık Oluşturma . . . . .	11
1.9	Bölümleme ve İçindekiler Tablosu . . . . .	11
1.10	Kitap Projesi Başlatma . . . . .	14
1.11	Varsayılan Sayfa Düzenini Değiştirme . . . . .	16
1.12	Satır ve Sayfa Kesme . . . . .	17
1.13	Paragraflar ve Cümle Sonları . . . . .	17
1.14	Aralıklar . . . . .	18
1.14.1	Dikey aralıklar . . . . .	18
1.14.2	Yatay aralıklar . . . . .	19
1.15	Metni Hizalamak ve Sütunlara Bölmek . . . . .	20
1.15.1	Hizalama . . . . .	20
1.15.2	Sütunlara bölmek . . . . .	20
1.16	Listeleme . . . . .	21
1.16.1	Temel listeler . . . . .	21
1.16.2	Listeleri özelleştirmek . . . . .	22

iii

## İçindekiler

1.17	Yazı tipleri . . . . .	23
1.17.1	Giriş . . . . .	23
1.17.2	Aile . . . . .	23
1.17.3	Biçem . . . . .	25
1.17.4	Boyut . . . . .	27
1.18	Renk kullanmak . . . . .	28
1.18.1	Yeni renkler oluşturmak . . . . .	30
<b>2</b>	<b>Matematiksel İfadeler</b>	<b>33</b>
2.1	Giriş . . . . .	33
2.1.1	Genel . . . . .	33
2.1.2	Matematik kipiyle Metin kipi arasındaki farklar . . . . .	35
2.1.3	Gruplandırma . . . . .	35
2.2	Parantezler, Gruplandırıcılar ve Oklar . . . . .	35
2.3	Yunan Harfleri . . . . .	37
2.4	Fonksiyonlar . . . . .	38
2.5	Yığın Simgeleri . . . . .	39
2.6	Matrisler . . . . .	41
2.7	Yazı Biçem ve Boyutları . . . . .	43
2.8	Boşluklar . . . . .	44
2.9	Denklem Ortamları . . . . .	45
2.10	Teorem ve Benzeri Ortamlar . . . . .	50
2.10.1	Temel Teoremler . . . . .	50
2.10.2	Sayaçlar . . . . .	51
2.10.3	Kanıtlar . . . . .	51
2.10.4	Teorem Stilleri . . . . .	52
<b>3</b>	<b>Özel Sayfalar</b>	<b>55</b>
3.1	Kaynakça . . . . .	55
3.1.1	Bütünleşik Kaynakça . . . . .	55
3.1.2	Kaynakçanın ayrı dosyada hazırlanması . . . . .	57
3.2	Dizin . . . . .	62
3.2.1	Çoklu Dizin . . . . .	64
3.2.2	Dizinin İçindekiler tablosuna yazılması . . . . .	65

## Şekil Listesi

1.1	Kaynak dosyanın olduğu dizinin düzenlenmesi . . . . .	15
1.2	Yazıtipi Aileleri . . . . .	24
1.3	Roman Yazıtipleri . . . . .	24
1.4	Sans Serif Yazıtipleri . . . . .	24
1.5	Typewriter Yazıtipleri . . . . .	25
1.6	Elyazısı . . . . .	25
1.7	Yazıtipi değiştiren paketler . . . . .	26
1.8	Yazıtipi Biçemleri . . . . .	26
1.9	Yazıtipi Boyutu Değiştiren Bildirimler . . . . .	27
1.10	Yazıtipleri Mutlak Boyutları . . . . .	28
1.11	<b>dvipsnames</b> seçeneğiyle gelen renkler . . . . .	29
3.1	Google Akademikten alıntı yapma . . . . .	58



## Tablo Listesi

1.1	LaTeX'de Belge Sınıfları . . . . .	7
1.2	LaTeX'de Belge Seçenekleri . . . . .	8
1.4	LaTeX'de Bölüm Seviyeleri . . . . .	11
1.6	LaTeX'de Ölçü Birimleri . . . . .	18
1.7	Renk Modelleri . . . . .	30
2.1	Gruplandırıcılar . . . . .	36
2.2	Oklar . . . . .	37
2.3	Yunan Harfleri . . . . .	38
2.4	Matematik Kipinde Yazı Biçemleri . . . . .	43
2.6	Teorem Stilleri . . . . .	52





## Önsöz



# 1 Genel

Bu bölümde LaTeX’le ilgili genel bilgilerden bahsedeceğiz.

## 1.1 LaTeX Nedir?

Önce TeX’le başlayalım. TeX, 1978’den itibaren Donald Knuth<sup>1</sup> tarafından belgelerin bilgisayarda dizilmesi için geliştirdiği bir dizgi sistemidir. LaTeX ise TeX’in kullanımını kolaylaştırmak için 1984 yılında Leslie Lamport<sup>2</sup> tarafından tasarlanmış bir makro pakettir.

LaTeX, genelde WYSIWYG (*Ne Görüyorsan Onu Alırsın*) editörleriyle karşılaştırılır. WYSIWYG, Microsoft Word, Libreoffice Writer gibi kelime işlemcilere ya da Adobe Indesign gibi programlara verilen genel bir isimdir. Hepsinin ortak özelliği, girdi ile çıktının aynı anda ve birlikte görünmesidir.

Bir metnin genel görünümü ve okunabilirliği, metnin nasıl hizalandığından ve kesildiğinden büyük ölçüde etkilenir. LaTeX, tüm paragraf için hizalamayı ve kesmeleri optimize eden son derece gelişmiş TeX algoritmalarını kullanır. Kelime işlemciler ve diğer programlar, satır başına çalıştıkları için oldukça yetersiz kalırlar. Bu, diğer şeylerin yanı sıra düzensiz aralıklara ve birçok kısa çizgiye sebep olur. Sonuçları görmeniz için Microsoft Word 2008 (Mac), Adobe InDesign CS4 ve LaTeX’le dizilmiş bir metni şuradan<sup>3</sup> inceleyebilirsiniz.

Sonuç, LaTeX’in diğer programların her ikisinden de üstün olduğunu açıkça gösterir: iki kat daha az tireleme kullanır ve yine de sözcük aralığındaki varyasyon, Word veya InDesign’dan belirgin şekilde daha azdır. LaTeX’te çok büyük sözcük aralığı içeren satırlar oluşmaz.

LaTeX’de girdi ve çıktı ekranı farklıdır ve çıktıyı görmek için girdinin derleme işleminden geçmesi gerekir. Ayrıca birçok şey için WYSIWYG editörlerinde olmayan yapılar vardır. Şimdi, bu yapıların ne oldukları ve ne işe yaradıklarını açıklayalım.

---

<sup>1</sup><https://www-cs-faculty.stanford.edu/~knuth/>

<sup>2</sup><http://www.lamport.org/>

<sup>3</sup><http://www.rtznet.nl/zink/comparison.pdf>

## 1.2 Önemli Yapılar

### 1.2.1 Komutlar

LaTeX komutları bir geribölü ( $\backslash$ ) işaretiyle başlar ve ya sadece harflerden ya da bir tane harf olmayan karakterden oluşurlar. Komut yazıldıktan sonra ya boşluk, ya bir sayı ya da harf olmayan bir karakter gelebilir.

Çoğu komut, zorunlu değişken alır. Bu zorunlu değişken komut adından sonra çengelli parantezler içine yazılır. Zorunlu değişken alan komutlar, zorunlu olmayan (isteğe bağlı) değişkenler de alabilir, bunlar da komut adından sonra gelen köşeli parantezler içine yazılırlar. Eğer değişkenler birden fazlaysa aralarına virgül koyularak ayrılır.

```
\:  
\LaTeX  
\item[...]  
\emph{...}  
\documentclass[...]{...}  
\subfloat[...][...]{...}  
\raisebox{...}[...][...]{...}  
\multicolumn{...}{...}{...}  
\bfseries ...
```

Fikir vermesi açısından yukarıda dokuz adet komut örneği verilmiştir. Birinci komut bir tane harf olmayan karakterden oluşan bir komuttur. İkincisi, değişkeni olmayan bir komuttur. Bazı harflerin büyük bazılarının da küçük olması komutların büyük-küçük harfe duyarlı olduğunu gösterir. Dokuzuncu komut ise bildirim şeklinde verilmiştir.

### 1.2.2 Paketler

LaTeX'de bazı özelliklerin (renkli yazmak, şekil eklemek vb.) kullanılabilmesi için kaynak dosyaya bazı paketlerin eklenmesi gerekir. Bu, `\usepackage` komutuyla yapılır. Bu komutun zorunlu değişkenine paket adı, zorunlu olmayan kısmına ise paket seçenekleri yazılır:

```
\usepackage[<seçenekler>]{<paket adı>}
```

Bu komutla paketin kaynak dosyaya eklenmesi TeX dağıtımıyla sisteminize kurulmuş olan paketin belgeye çağrılarak işe koşulması demektir.

### 1.2.3 Ortamlar

LaTeX’de ortamlar önemli bir yer tutar. Örneğin `document` bir ortamdır. Ortamları birden fazla ögeye uygulanan komutlar olarak düşünebiliriz.

Bir ortam `\begin` komutuyla başlayıp `\end` komutuyla biter. Her iki komutun zorunlu değişkeni ortamın adıdır:

```
\begin{<ortam adı>}
...
\end{<ortam adı>}
```

### 1.2.4 Gruplar

Gruplar, ortam benzeri yapılardır. Grup `\begingroup` komutuyla başlar ve `\endgroup` komutuyla biter. Grubun içinde kullanılan bir bildirim sadece gruba uygulanır.

### 1.2.5 Boşluklar

LaTeX’de belgenizin metnini oluştururken ister klavyedeki Space, ister Tab tuşu ile boşluk bırakın, bu boşluklar LaTeX tarafından bir karakter boşluk olarak algılanır. Arka arkaya çok sayıda boşluk bırakılsa da LaTeX bunu tek bir boşluk olarak algılar.

Bütün bir satırın boş bırakılması LaTeX tarafından paragraf başı olarak algılanır. Arka arkaya boş bırakılan çok sayıda boş satır LaTeX tarafından tek bir boş satır yani paragraf başı olarak algılanır.

İster bir boşluk, isterseniz de çok sayıda boşluk bırakın. İkisi de bir boşluk gibi işlem görür. Değişen bir şey yok.

Ayrıca boş bir satır yeni paragraf demektir, burada olduğu gibi.

---

İster bir boşluk, isterseniz de çok sayıda boşluk bırakın. İkisi de bir boşluk gibi işlem görür. Değişen bir şey yok. Ayrıca boş bir satır yeni paragraf demektir, burada olduğu gibi.

---

Komutlardan sonra gelen boşlukları LaTeX dikkate almaz. Komuttan sonra gerçekten bir boşluk bırakmak için, ya `{ }` ve ardından boşluk girilir ya da komut adından sonra özel bir boşluk komutu kullanılır.

```
\LaTeX boşluk yok.\\
\LaTeX{} boşluk var.\\
\LaTeX\ boşluk komutuyla boşluk.
```

---

$\LaTeX$  boşluk yok.  
 $\LaTeX$  boşluk var.  
 $\LaTeX$  boşluk komutuyla boşluk.

---

### 1.2.6 Özel amaçlı karakterler

Aşağıdaki karakterlerin herbiri  $\LaTeX$ 'de özel bir amaç için kullanılır. Dolayısıyla bu karakterleri doğrudan kullanmak istenmeyen sonuçlara yol açabilir.

# \$ % & { } ~ ^ \_ \

Bu karakterleri çıktıda elde etmek isterseniz, sondaki hariç, başına bir geribölü koymanız gerekir. Sondaki için, yani bir geribölü sembolü elde etmek içinse `\textbackslash` komutunu kullanabilirsiniz. Eğer `\\` komutunu vererseniz yeni bir satır başlatmış olursunuz.

Bu karakterlerden örneğin yüzde (%) karakteri kaynak dosyanızda yorum ya da açıklama yazmaya yarar. Bu sembolden sonra yazılanları  $\LaTeX$  dikkate almaz ve çıktıda görünmez.

Diğer karakterlerden örneğin (\$) nin matematik kipini açma ve kapatmaya yarar. (&) karakteri tablo ve benzeri yapılarda dikey hizalama yapmak için veya sütun ayracı olarak kullanılır. Çengelli parantezlerden zaten yeterince bahsettik. (#) karakteri yeni komutlar tanımlamakta kullanılır. Tilda (~) ise genişlemeyen bir boşluk yaratmak için kullanılır. (^) ve (\_) karakterleri de matematikte üst ve alt indis yazmak için kullanılır. Her birinin kullanımlarından yeri geldiğinde tekrar bahsedeceğiz.

## 1.3 Kurulum

$\LaTeX$ 'i kurmak için ilk olarak bir TeX dağıtımı edinmeniz gerekir. Dağıtımlar, dizgi sistemini ve  $\LaTeX$ 'de belge oluşturabilmek için gereken paketleri içerir.

İkinci ihtiyaç duyacağınız şey bir  $\LaTeX$  editörüdür. Edindiğiniz TeX dağıtımları genelde bir  $\LaTeX$  editörüyle birlikte gelir. Tabi editör kişisel bir tercihtir ve bir  $\LaTeX$  editörü yerine basit bir metin editörü kullanabilirsiniz. Ancak farklı işletim sistemleri için birçok iyi  $\LaTeX$  editörü vardır ve bunların kod vurgulama, otomatik tamamlama, otomatik belge oluşturma gibi  $\LaTeX$ 'e özgü işlevleri vardır. Dolayısıyla  $\LaTeX$ 'de yeniyken bir editör kullanmanızı tavsiye ederiz.

### 1.3.1 GNU/Linux

Linux sistemlere MiKTeX<sup>4</sup> ya da TeX Live<sup>5</sup> kurulabilir. MiKTeX'in indirme sayfasında Ubuntu, Mint, Debian, Fedora, CentOS ve openSUSE gibi Linux dağıtımlarında nasıl kurulacağı anlatılmıştır. TeX Live ise tüm popüler Linux dağıtımlarının depolarında mevcut olup, paket yöneticisi ya da komut satırı yardımıyla kurulabilir. Örneğin Ubuntu, Debian, Mint, Pardus gibi .deb uzantılı paketlerin kullanıldığı dağıtımlarda

```
sudo apt-get install texlive-base
```

komutuyla temel kurulum,

```
sudo apt-get install texlive-full
```

komutuyla da tam kurulum yapılır.

### 1.3.2 Mac OS

Mac OS kullanıcıları için iki seçenek mevcuttur: MiKTeX<sup>6</sup> ya da MacTeX<sup>7</sup>. MiKTeX kurulumu için .dmg uzantılı, MacTeX içinse .pkg uzantılı dosya indirilir ve standart kurulum yapılır.

### 1.3.3 Windows

Windows için aşağıdaki dağıtımlardan birini kurabilirsiniz.

- MiKTeX<sup>8</sup>
- TeX Live<sup>9</sup>
- proTeXt<sup>10</sup>

MiKTeX veya TeX Live dağıtımını kurarsanız sisteminize TeXworks<sup>11</sup> editörü de kurulur. proTeXt dağıtımını MiKTeX tabanlı bir dağıtım olup, tüm paketleri içerir ve beraberinde TeXstudio<sup>12</sup> editörüyle gelir.

---

<sup>4</sup><https://miktex.org/download>

<sup>5</sup><http://www.tug.org/texlive/>

<sup>6</sup><https://miktex.org/download>

<sup>7</sup><http://www.tug.org/mactex/>

<sup>8</sup><https://miktex.org/download>

<sup>9</sup><http://www.tug.org/texlive/>

<sup>10</sup><https://tug.org/protext/>

<sup>11</sup><https://www.tug.org/texworks/>

<sup>12</sup><https://texstudio.org/>

### 1.3.4 LaTeX Editörleri

Hangi editörü kullanacağınıza birkaç deneme yaptıktan sonra karar verebilirsiniz. Burada<sup>13</sup> en çok beğenilen editörler listelenmiş.

Her LaTeX editöründe olan özelliklerin (otomatik kod tamamlama vb.) yanı sıra kullanıcı dostu arayüzü, yüzde yüze yakın Türkçe desteği, ücretsiz oluşu ve her üç sistemde de çalışabilmesinden dolayı TeXstudio<sup>14</sup>'yu tavsiye ediyoruz. Karar sizin.

### 1.3.5 Çevrimiçi Editörler

LaTeX'i hiçbir kurulum yapmadan çevrimiçi de kullanabilirsiniz. Aşağıda üç tanesi listelenmiştir.

- Overleaf<sup>15</sup>
- Papeeria<sup>16</sup>
- LaTeX Base<sup>17</sup>

En popüler olanı Overleaf olup, sayfasında beğenebileceğiniz binlerce şablon<sup>18</sup> ve LaTeX kullanımına yönelik anlatımlar<sup>19</sup> bulunur.

## 1.4 Tipik Bir Belge Yazımı

LaTeX'in varsayılan dosya uzantısı .tex'tir. Bu basit bir metin dosyası olup, LaTeX editörleriyle oluşturulup düzenlenebileceği gibi basit bir metin editörüyle de düzenlenebilir.

Bir belge hazırlamaya başlamak için verilecek ilk komut

```
\documentclass[...]{...}
```

olup, çengelli parantezler arasına oluşturmak istediğiniz belgenin sınıfı yazılır. Köşeli parantezlerin içine de isteğe bağlı bazı değişkenler yazılabilir. Eğer bu kısım boş bırakılırsa LaTeX varsayılan değerleri alacaktır. Bu komutun ardından sırasıyla `\begin{document}` ve `\end{document}` komutları verilerek belge ortamı oluşturulur. `\end{document}` komutuyla LaTeX'e belgenin bittiği söylenmiş olur ve LaTeX bu komuttan sonra girilenleri dikkate almaz.

---

<sup>13</sup><https://beebom.com/best-latex-editors/>

<sup>14</sup><https://texstudio.org/>

<sup>15</sup><https://www.overleaf.com/>

<sup>16</sup><https://papeeria.com/>

<sup>17</sup><https://latexbase.com/>

<sup>18</sup><https://www.overleaf.com/latex/templates>

<sup>19</sup><https://www.overleaf.com/learn>



## 1.5 Belge Sınıfları ve Seçenekleri

`\documentclass` komutuyla `\begin{document}` komutu arasına *sahanlık* denir. Sahanlık, belgenin ayarlarının yapıldığı kısımdır ve bu kısım çıktıda görünmez. `\begin{document}` ile `\end{document}` arasına da *gövde* denir. İçerik burada oluşturulur.

Aşağıda asgari bir LaTeX kaynak dosyası gösterilmiştir. `\documentclass` komutunun değişkeni olan `article`, belgenin makale olacağını belirtir.

```
\documentclass{article}
\begin{document}
  İşte ilk belgem.
\end{document}
```

Bu noktadan sonra örnek kaynak dosyayı LaTeX editörünüzünde oluşturup önceden oluşturduğunuz bir dizine kaydedin. Kaydederken dosya adında boşluk ve Türkçe karakter kullanmayın. Örneğin kaynak dosyanız `belge1.tex` olsun.

İkinci aşama kaynak dosyanın derlenmesidir. Derleme işlemi için LaTeX editörlerinde genelde araç çubuğunda oklar bulunur. Oka tıklandığında dosya derlenir ve sonuç, çıktı ekranında görünür.

Eğer metin editörü kullanıyorsanız derlemeyi uçbirimde (terminal, konsol,...) yapmanız gerekir. Derleme için uçbirim kaynak dosyanın olduğu dizinde açılıp

```
pdflatex belge1
```

komutu verilmelidir.

Derleme işleminden sonra kaynak dosyanızın olduğu dizinde `belge1.tex` ve `belge1.pdf` dosyalarının yanında yine `belge1` ile başlayan farklı uzantılara sahip dosyalar olacaktır. Bu dosyaların ne olduklarına ilerleyen yazılarda değinilecektir ancak dileyen okur (Oetiker et al., 2006, s. 13-14)'e bakabilir.

## 1.5 Belge Sınıfları ve Seçenekleri

Bölüm 1.4'de `\documentclass` komutunun zorunlu değişkeninin belge sınıfı olduğunu ve köşeli parantezler içine de seçeneklerin yazılacağından bahsetmiştik. Bu yazıda bunların neler olabileceklerinden bahsedelim.

Başka sınıflar olmakla birlikte LaTeX'de varsayılan olarak kullanılan beş belge sınıfı vardır.

**Tablo 1.1:** LaTeX'de Belge Sınıfları

Sınıf	Açıklama
<code>article</code>	Makale
<code>report</code>	Makaleden daha hacimli belgeler için kullanılır. Rapor, tez gibi

## 1 Genel

Sınıf	Açıklama
book	Kitap
letter	Mektup
beamer	Sunu

Bu beş sınıftan `article`, `report` ve `book` için kullanılabilecek seçenekler Tablo 1.2’de gösterilmiştir.

**Tablo 1.2:** LaTeX’de Belge Seçenekleri

Seçenek	Açıklama
<b>10pt, 11pt, 12pt</b>	Belge ana yazı büyüklüğü.
<b>a4paper, a5paper, letterpaper,...</b>	Kağıt boyutu.
<b>fleqn</b>	Formülleri ortada yazmak yerine, sola bitişik yazar.
<b>leqno</b>	Formül numaralarını sağ yerine sol tarafa koyar.
<b>titlepage, notitlepage</b>	Belge başlığını attıktan sonra yeni bir sayfa açıp açmayacağını belirler.
<b>onecolumn, twocolumn</b>	Belgenin tek sütun veya çift sütun dizileceğini belirtir.
<b>twoside, oneside</b>	Belgenin kağıdın hep tek tarafına mı yoksa iki tarafına mı basılacağını belirtir.
<b>landscape</b>	Belgeyi enine tutulmuş kağıda basılmak üzere hazırlar.
<b>openright, openany</b>	Belgede bölümleri hep sağ sayfalardan veya ilk gelen boş sayfadan başlatır.
<b>draft, final</b>	Belgeyi sırasıyla <i>taslak</i> ve <i>son</i> şeklinde hazırlar. <b>draft</b> seçilirse, sağ taraftan fırlamış olan satırlar kalın siyah bir çizgiyle işaretlenir.

Bu seçeneklerin her birinin kullanılabilirliği belge sınıfına göre farklılık gösterir. Aşağıdaki tabloda hangi seçeneğin hangi sınıf için varsayılan olduğu ve kullanılabilir olup olmadığı gösterilmiştir.

Seçenek	book	report	article
<b>10pt</b>	1	1	1
<b>letterpaper</b>	1	1	1
<b>oneside</b>	1/2	1	1
<b>twoside</b>	1	1/2	1/2

## 1.6 Türkçe Dil Ayarları ve Çoklu Dil Kullanımı

Seenek	book	report	article
<b>openany</b>	1/2	1	0
<b>openright</b>	1	1/2	0
<b>titlepage</b>	1	1	1/2
<b>final</b>	1	1	1

1: varsayılan 1/2: kullanılabilir 0:kullanılamaz

Örneğın belgeye

```
\documentclass[a4paper,12pt]{article}
```

komutuyla başlarsak LaTeX'e kağıt boyutu A4, ana yazı büyüklüğü 12 punto olan bir makale yazacağımızı bildirmiş oluruz.

Başka bir örnek

```
\documentclass[a5paper,11pt,twocolumn]{book}
```

olsun. Bu örnekte kağıt boyutu A5, ana yazı büyüklüğü 11 punto olan bir kitap yazacağımızı ve kitabın iki sütun olarak dizilmesini söyledik.

## 1.6 Türkçe Dil Ayarları ve Çoklu Dil Kullanımı

LaTeX'de Türkçe belgeler oluşturmak için öncelikle sahanlığa

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

komutlarının verilmesi gerekir.

T1 seçenekli **fontenc** paketi yazıtipi kodlamasıyla ilgili bir paket olup, hecelemenin doğru şekilde yapılmasını sağlar. Bir çok Avrupa dilinde de T1 seçeneğiyle kullanılır. turkish seçenekli **babel** paketi de Chapter, Table, Contents,... gibi isimlerin Türkçeleşmesi (Bölüm, Tablo, İçindekiler,...) içindir.

Yakın zamana kadar ö, ş, ç,... gibi Türkçe karakterlerin kullanılabilmesi için sahanlığa `\usepackage[utf8]{inputenc}` ya da `\usepackage[latin5]{inputenc}` komutlarından birinin verilmesi gerekiyordu. Bu paket (**inputenc**) girdi kodlamasını yöneten bir pakettir. Son güncellemelerle birlikte bu paketin kullanılma zorunluluğu ortadan kalkmıştır.

Aşağıda Türkçe asgari bir LaTeX kaynak dosyası örneği verilmiştir.

## 1 Genel

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\begin{document}
  İşte Türkçe ilk belgem.
\end{document}
```

Türkçe dışında ikinci bir dil kullanmak isterseniz, örneğin İngilizce, **babel** paketinin seçeneğini

```
\usepackage[english,turkish]{babel}
```

şeklinde değiştirmeniz gerekir. Burada etkin olan dil Türkçedir. İngilizceyi etkin hale getirmek için `\selectlanguage{english}` komutu kullanılır. Tekrar Türkçeye geçmek için de benzer şekilde `\selectlanguage{turkish}` komutu kullanılır.

Bir kelime ya da cümle gibi kısa metinler kullanılacaksa `\foreignlanguage` komutu kullanılabilir:

```
\foreignlanguage{<dil>}{<metin>}
```

Uzun metinler içinse diğer bir seçenek `otherlanguage` ortamıdır.

```
\begin{otherlanguage}{<dil>}
...
\end{otherlanguage}
```

Bu ortamın isimleri değiştirmeyen, örneğin, dil seçeneği İngilizce olmasına rağmen belgeye bir tablo eklediğinizde “Table” yerine yine “Tablo” adını yazan yıldızlı sürümü de (`otherlanguage*`) vardır.

## 1.7 Heceleme

Bazen tüm bu ayarlamalara rağmen LaTeX bazı kelimeleri doğru heceleyemeyebilir. Böyle durumlarda hecelemeyle ilgili yapmamız gerekir. Yanlış hecelenen kelimenin bölünebileceği yerler `\-` komutuyla gösterilir:

```
He\-\ce\-\le\-\me
```

Bu sadece ilgili kelimenin tireyle ayrıldığı yerde doğru hecelenmesini sağlar. Aynı kelime belgenin başka bir yerinde yine yanlış hecelenebilir. Bunun yerine `\begin{document}` komutundan sonra `\hyphenation` komutuyla hece yerleri tire (-) işaretiyle gösterilmiş olan kelime listesi oluşturulursa belgenin tamamına bu kural uygulanmış olur. Örneğin

```
\hyphenation{He-ce-le-me FORTRAN}
```

komutuyla “Heceleme” kelimesinin nereden bölüneceği, “FORTRAN”, “Fortran” ya da “fortran” kelimelerinin bölünmeyeceği LaTeX’e söylenmiş olur.

## 1.8 Belgeye Başlık Oluşturma

LaTeX’de belgeye başlık oluşturmak için `\title` komutu kullanılır. Yazar adı `\author` komutuyla girilir. Birden fazla yazar varsa yazar adları arasına `\and` komutu girilir.

İsteğe bağlı olarak tarih için `\date` komutu kullanılır. Eğer `\date` komutu kullanılmazsa LaTeX belgenizi derlediğiniz günün tarihini basar. Tarihin basılmasını istemiyorsanız, bu komutu tarih yazılmadan `\date{}` şeklinde kullanmanız gerekir.

Son olarak, başlığın belgenize yazılması için `\begin{document}` komutundan sonra başlığı oluşturmak istediğiniz yere `\maketitle` komutunu girersiniz. Belge başlığını attıktan sonra yeni bir sayfanın açılıp açılmayacağı Bölüm 1.5’de belirttiğimiz gibi belgenin sınıfına bağlı olarak belirlenir.

Ayrıca `\title`, `\author` ve `\date` komutları `\thanks` komutunu içerebilir. Bu komutun değişkeni bir e-posta adresi, iş adresi veya bir teşekkür metni olabilir.

```
\documentclass[a4paper,12pt]{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\title{Belge Başlığı}
\author{Yazar 1\thanks{A Üniversitesi} \and Yazar 2\thanks{B Üniversitesi}}
\date{XX.XX.XXXX}
\begin{document}
\maketitle

İçerik...
\end{document}
```

## 1.9 Bölümlendirme ve İçindekiler Tablosu

LaTeX’de belgenizi bölümlere ayırmak için 7 seviye bulunmaktadır.

**Tablo 1.4:** LaTeX’de Bölüm Seviyeleri

Komut	Seviye	Açıklama
<code>\part{...}</code>	-1( book ve report ) o ( article)	letter hariç
<code>\chapter{...}</code>	0	sadece book ve report
<code>\section{...}</code>	1	letter hariç
<code>\subsection{...}</code>	2	letter hariç

## 1 Genel

Komut	Seviye	Açıklama
<code>\subsubsection{...}</code>	3	letter hariç
<code>\paragraph{...}</code>	4	letter hariç
<code>\subparagraph{...}</code>	5	letter hariç

Türkçe dil paketi ekli belgelerde `\part` komutu “Kısım”, `\chapter` komutu “Bölüm” olarak yazılır. Kısımlar I, II, III,... şeklinde bölümler ise 1, 2, 3,... şeklinde numaralandırılır. `\section` komutu book ve report sınıflarında `\chapter` komutunu takip ederek 1.1, 1.2,... diğer sınıflarda 1, 2, 3,... şeklinde numaralandırılır. `\subsection` komutu da `\section` komutunu takip ederek numaralandırılır.

Türkçe dil paketi ekli belgelerde `\part` komutu “Kısım”, `\chapter` komutu “Bölüm” olarak yazılır. Kısımlar I, II, III,... şeklinde bölümler ise 1, 2, 3,... şeklinde numaralandırılır. `\section` komutu book ve report sınıflarında `\chapter` komutunu takip ederek 1.1, 1.2,... diğer sınıflarda 1, 2, 3,... şeklinde numaralandırılır. `\subsection` komutu da `\section` komutunu takip ederek numaralandırılır.

İçindekiler tablosu için LaTeX’e `\tableofcontents` komutu verilir. Bu komutun yazıldığı yerde İçindekiler tablosu oluşturulur. İçindekiler tablosunun doğru dizilmesi için kaynak dosyanızı en az iki kere derlemeniz gerekir.

LaTeX’de article sınıfında 4 ve 5’inci seviye başlıklara, book ve report sınıflarında ise bunlara ek 3’üncü seviye başlıklara numara verilmez ve numara verilmeyen başlıklar İçindekiler tablosuna yazılmaz. Bu seviyelerdeki başlıklara numara verilmesini ve İçindekiler tablosuna yazılması için iki adet `\setcounter` komutu

```
\setcounter{secnumdepth}{<seviye>}
\setcounter{tocdepth}{<seviye>}
```

şeklinde kullanılır. Birinci komuttaki `<seviye>` değişkeninde kaçınıcı seviyeye kadar olan başlıkların numaralandırılacağını, ikinci komuttaki `<seviye>` değişkeninde de kaçınıcı seviyeye kadar olan başlıkların İçindekiler tablosuna yazılacağını sayıyla belirtirsiniz. Örneğin book ve report sınıflarında

```
\setcounter{secnumdepth}{3}
\setcounter{tocdepth}{3}
```

komutlarıyla `\subsubsection` komutuna kadar olan başlıklara hem numara verir hem de İçindekiler tablosuna yazdırırsınız. Komutların çalışması için ya sahanlıkta ya da `\tableofcontents` komutundan önce verilmelidir.

Uzun başlıkların İçindekiler tablosunda daha kısa yazılması istenirse bölüm komutlarının zorunlu olmayan değişkenine başlıkların kısa şekli yazılır:

```
\section[Kısa Başlık]{Uzunuuuuuuuuuuuuuuuun Başlık}
```

Bölüm komutlarının birde yıldızlı sürümleri vardır:

```

\part*{...}
\chapter*{...}
\section*{...}
\subsection*{...}
\subsubsection*{...}
\paragraph*{...}
\subparagraph*{...}

```

Komutlar bu şekilde verildiğinde başlığa numara verilmez ve İçindekiler tablosuna yazılmaz.

İçindekiler tablosunu LaTeX otomatik oluştursa da elle eklemeler yapılabilir, hatta Kaynakça gibi özel sayfalarda bu eklemeler gereklidir. Bunun için `\addcontentsline` komutu kullanılır.

```
\addcontentsline{toc}{<giriş formatı>}{<giriş metni>}
```

Burada `toc`, bilginin yazılacağı İçindekiler tablosunun dosya uzantısıdır. Bütünlük olarak oluşturduğunuz kaynakçanın İçindekiler tablosuna yazılması için `\begin{thebibliography}` komutunun peşine `book` ve `report` sınıflarında

```
\addcontentsline{toc}{chapter}{Kaynakça}
```

`article` sınıfında ise

```
\addcontentsline{toc}{section}{Kaynaklar}
```

komutunun verilmesi gerekir.

“Kaynakça” ya da “Kaynaklar” isimleri yerine farklı isimler kullanılabilir elbette. Ancak `thebibliography` ortamının oluşturulduğu yerlerde LaTeX bu isimleri yazdıracağından tutarlı olması açısından bu isimler önerilmiştir.

```

\documentclass[a4paper,12pt]{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\title{\LaTeX'de Bölümlendirme ve İçindekiler Tablosu Oluşturma}
\author{\TeX dizgi}
\begin{document}
\maketitle
\tableofcontents
\section{Birinci Seviye Başlık}
İçerik...
\subsection{İkinci Seviye Başlık}
İçerik...
\subsubsection{Üçüncü Seviye Başlık}
İçerik...
\section[Kısa Başlık]{Uzunuuuuuuun Başlık}

```

## 1 Genel

```
İçerik...
\section*{Numarasız Başlık}
\addcontentsline{toc}{section}{Numarasız Başlık}
İçerik...
\end{document}
```

### 1.10 Kitap Projesi Başlatma

LaTeX’de kitap yazmaya başlamak için belge sınıfı `book` seçilir. Bunun dışında kitapların *Baş*, *Gövde* ve *Son* kısımları olur. Bu kısımların nerede başlayıp nerede bittikleri aşağıdaki komutlarla LaTeX’e bildirilir:

- `\frontmatter` (baş) komutu `\begin{document}` komutundan hemen sonra verilir. Bu komut, baş taraftaki İçindekiler, Önsöz gibi kısımların sayfa numaralandırmasını Roma rakamıyla yapar. Ayrıca bu kısımda bölüm komutları (\*) işareti olmadan verildiğinde (örneğin `\chapter{Önsöz}`) bunlara numara verilmez ancak İçindekiler tablosuna yazılırlar.
- `\mainmatter` (gövde) komutu kitabın ilk bölüm başlığından hemen önce verilmelidir. Buradan itibaren sayfa numaralandırmasını yeniden başlatıp Arap rakamlarına geçer.
- `\appendix` (ekler) komutu kitabınızın eklerindeki bölümleri harflerle numaralandırır (Ek A, Ek B, ... ).
- `\backmatter` (son) komutu kitabınızda her şey bittikten sonra verilir fakat bilinen belge sınıflarında görünürde hiçbir etkisi yoktur.

Kitap gibi büyük hacimli belgelerle çalışırken kaynak dosyanızı parçalara ayırmak gerekebilir. LaTeX bunun için size iki komutla yardımcı olur: `\input` ve `\include`. İkisi arasındaki fark `\include` komutuyla eklediğiniz metin yeni bir sayfadan başlayarak dizilir.

Bu komutların zorunlu değişkeni eklemek istediğiniz dosyanın adıdır. Örneğin kaynak dosyanızla aynı dizinde yer alan `dosya1.tex` dosyasını eklemek için

```
\include{dosya1}
```

komutunu kullanırsınız. Eğer dosya uzantısı `.tex` değilse (örneğin `.txt` olsun) o zaman dosya adını uzantısıyla yazmanız gerekir:

```
\include{dosya1.txt}
```

Ayrıca, hangi dosyaların eklenebileceğini LaTeX’e bildiren bir komut vardır: `\includeonly`. Bu komut, sadece sahanlığa yazılabilir. Komutun zorunlu değişkeninde eklenebilecek dosyalar aralarına virgül koyularak (ve boşluk bırakılmadan) listelenir:

```
\includeonly{dosya1,dosya2,dosya3,...}
```

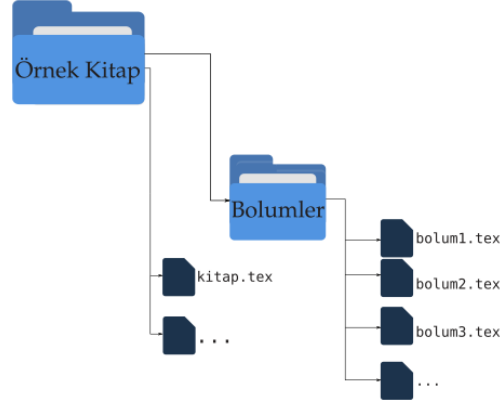


### 1.10 Kitap Projesi Başlatma

Böyle bir liste oluşturulduktan sonra bu listede olmayan bir dosya artık `\include` komutuyla kaynak dosyaya eklenemez.

`\input` komutu sahanlıkta da kullanılabilir. Örneğin, sahanlığınızı tek bir dosyaya yazıp, bu dosyayı bu komutla sahanlığa ekleyebilirsiniz.

Daha düzenli çalışmak adına kaynak dosyanızın olduğu dizini de düzenleyebilirsiniz.



Şekil 1.1: Kaynak dosyanın olduğu dizinin düzenlenmesi

Bu şekilde bir düzenleme yaptığınızda `\input` ya da `\include` komutlarıyla dosya eklemek istediğinizde dosyanın bulunduğu dizini de göstermeniz gerekir.

Burada kaynak dosya `kitap.tex`'dir. Bu kaynak dosyaya `bolum1.tex` dosyasını eklemek istediğinizde komutu

```
\input{Bolumler/bolum1}
```

şeklinde verirsiniz. Bu sayede kaynak dosyanızın olduğu dizinde (Örnek Kitap) sadece `kitap` ile başlayan dosyalar olur. Diğer dosyalar alt dizinde (Bolumler) yer alır.

Dikkat edilirse “Örnek Kitap” dışında, “Bolumler” alt dizini ve tüm dosya adları Türkçe karakter ya da boşluk içermez.

```
\documentclass[a4paper,12pt]{book}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\title{Örnek Kitap}
\author{\TeX dizgi}
\begin{document}
\frontmatter
\maketitle
\tableofcontents
\input{Bolumler/onsoz}
```

## 1 Genel

```
\mainmatter
\input{Bolumler/bolum1}
\input{Bolumler/bolum2}
\appendix
\input{Bolumler/ek1}
\backmatter
\end{document}
```

### 1.11 Varsayılan Sayfa Düzenini Değiştirme

LaTeX’de varsayılan kağıt boyutunun letterpaper olduğunu Bölün 1.5’de ifade etmiştik. Ayrıca aynı yazıda başka bir kağıt boyutunun nasıl seçileceğine de yer vermiştik. Şimdi ise hem sayfamızın kenar boşluklarının nasıl ayarlanacağından hem de ön tanımlı olmayan, tamamen keyfi bir sayfa boyutunun nasıl belirleneceğinden bahsedelim.

Bu tür sayfa düzenlemeleri için LaTeX’de **geometry**<sup>20</sup> paketi kullanılır. Öncelikle paketi

```
\usepackage{geometry}
```

komutuyla sahanlığa ekleyin. Ardından paket seçeneklerinde aşağıdaki tanımlamalarla sayfanın düzenini değiştirebilirsiniz:

Tanım	Değer
top	üst boşluk
bottom	alt boşluk
left	sol boşluk
right	sağ boşluk
paperwidth	sayfa genişliği
paperheight	sayfa yüksekliği

Örneğin sahanlıkta

```
\usepackage[paperwidth=175mm,paperheight=255mm,top=2cm,bottom=2cm,
left=2.5cm,right=2.5cm]{geometry}
```

komutunu verdiğinizde boyutu  $175 \times 255$  mm, üst ve alt boşluğu 2 cm, sol ve sağ boşluğu 2.5 cm olan bir sayfa düzeni oluşturursunuz. Dilerseniz `paperwidth` ve `paperheight` tanımlamalarını yerine, örneğin `a4paper` yazarak sadece kenar boşluklarla ilgili tanımlamaları yapabilirsiniz.

<sup>20</sup><http://ftp.cc.uoc.gr/mirrors/CTAN/macros/latex/contrib/geometry/geometry.pdf>

LaTeX’de milimetre (mm) ve santimetre (cm) dışında inç (in), punto (pt), em ve ex gibi ölçü birimleri de vardır. Bunlara ileride değinilecektir. Ayrıca yine ileride daha ayrıntılı sayfa düzeni oluşturmaktan da bahsedeceğiz. Bu aşamada bu kadarı yeterli olacaktır.

## 1.12 Satır ve Sayfa Kesme

LaTeX, kelimeler arası boşlukları otomatik ayarlayarak satırları iki yana yaslayarak dizer. Bir satırı kesip yeni bir satıra geçmek için `\\` veya `\newline` komutları kullanılır.

Birinci komut `\\*` şeklinde verildiğinde satırdan sonra sayfa kesilmesini önler.

Benzer şeyi `\linebreak` komutu da yapar. Fakat bu komut ile satır kesilirse LaTeX kalan yarım satırı iki yana yaslar. `\nolinebreak` komutu ise satırın kesilmesini önler.

Birçok kelimeyi birlikte aynı satırda tutmak gerekirse `\mbox` komutu kullanılır:

```
\mbox{<metin>}
```

Buradaki `<metin>` içindeki kelimeler her durumda birleşik kalırlar. Benzer şeyi `\fbox` komutu da metin etrafına çizgi çizerek yapar.

Sayfayı kesip yeni bir sayfaya geçmek için `\newpage` ya da `\pagebreak` komutları kullanılır. `\nopagebreak` komutu sayfa kesilmesini önler. `\newpage` ile `\pagebreak` komutları arasında da `\newline` ile `\linebreak` komutlarındakine benzer bir fark vardır.

## 1.13 Paragraflar ve Cümle Sonları

Boş bir satırın yeni bir paragraf açtığını Bölüm 1.2.5’de belirtmiştik. Aynı şey, `\par` komutuyla da yapılabilir. Ancak bu komut yeni bir paragraf açmaktan ziyade farklı amaçlar için kullanılır (yeri geldiğinde değinilecektir). Nitekim kaynak dosyanızın okunabilirliği açısından paragrafları ayırmak için boş bir satır bırakmak daha kullanışlıdır.

LaTeX’de varsayılan olarak `\chapter` ve `\section` gibi bölümlendirme komutlarından sonra oluşturulan ilk paragraf girintisiz, sonrakiler girintili olur. Bu, paragraf başlarında `\indent` ya da `\noindent` komutlarıyla tek seferliğine değiştirilebilir. Birinci komut girinti oluşturur, ikincisi ise girintiyi kaldırır.

LaTeX, okumayı kolaylaştırmak için cümle sonlarında fazladan boşluklar bırakır. Bunu yaparken de her cümlenin nokta, soru işareti veya ünlem işaretiyle bittiğini varsayar. Kısaltmalarda büyük harflerden sonra nokta geldiğinden, büyük harften sonra nokta

## 1 Genel

koyulursa LaTeX bunu cümle sonu saymaz. Eğer bir büyük harften sonra nokta koyuyorsanız ve burası cümle sonuysa LaTeX'in burayı cümle sonu sayması için büyük harften sonraki noktanın önüne \@ koymanız gerekir.

LaTeX'in noktadan sonra fazladan boşluk *koyulmasını* isterseniz \frenchspacing komutunu kullanırsınız. Bu komutu kullandıysanız, artık noktadan önce \@ koymanıza gerek yoktur. Daha sonra tekrar cümle sonlarında fazladan boşluk kullanmak istenirse de \nonfrenchspacing komutu kullanılır.

Unvan kısaltmasından sonra unvanın ait olduğu kelimeyle birlikte kalması ve fazladan boşluk bırakılmaması için tilda (~) işareti kullanılabilir. Bu işaret hem genişlemeyen bir boşluk bırakır hem de satırın orada kesilmesini önler.

### 1.14 Aralıklar

LaTeX'de hem dikey hem de yatay aralıklar otomatik olarak ayarlanır. Fazladan aralıklar bırakmak için komutlar kullanılır.

Aralık bırakırken kullanabileceğimiz ölçü birimleri Tablo 1.6'de gösterilmiştir.

**Tablo 1.6:** LaTeX'de Ölçü Birimleri

Birim	Değer
mm	milimetre $\approx 1/25$ inç
cm	santimetre = 10 mm
in	inç = 25.4 mm
pt	punto $\approx 1/72$ inç
em	Kullanılan yazı tipinde 'M' harfinin genişliği
ex	Kullanılan yazı tipinde 'x' harfinin yüksekliği

#### 1.14.1 Dikey aralıklar

Dikey aralık birkaç komutla bırakılabilir. Bunlardan biri \vspace olup, komut iki boş satır arasında

`\vspace{<uzunluk>}`

şeklinde verilir. Komut bu şekilde verildiğinde komutun zorunlu değişkeninde birimiyle belirtilen uzunluk kadar dikey aralık bırakılır. Eğer bir sayfanın başında veya sonunda aralık bırakılmak istenirse, komut \vspace\* şeklinde yıldızlı vermelidir. Bu komutun aralığa ilave yapan \addvspace sürümü de vardır.

Bir paragrafın iki satırı arasında veya bir tablonun satırları arasında ilave aralık açmak için

`\[<uzunluk>]`

komutu kullanılır.

`A\[1ex]`  
B

Bu komutlarda belirtilen uzunluklar negatif de olabilir.

`A\[−2ex]`  
B

Sınırsız bir dikey aralık oluşturmak için `\vfill` komutu kullanılır. Bu komuttan sonra gelen her şey sayfanın altına yaslanır.

Ön tanımlı gelen `\smallskip`, `\medskip` ve `\bigskip` komutları sırasıyla küçük, orta ve büyük aralıklar bırakır.

### 1.14.2 Yatay aralıklar

Ön tanımlı yatay aralıklar

`\ , \: \; \quad \qquad \!`

komutlarıyla verilir. Bu komutlar sırasıyla bir sözcük arası, 3/18 em, 4/18 em, 5/18 em, 1 em, 2 em, −3/18 em uzunlukta yatay aralık bırakır.

Belli bir uzunlukta yatay aralık bırakmak için `\hspace` komutu kullanılır. Yine dikey aralıkta olduğu gibi yatay aralık negatif de olabilir. Eğer aralık satır başına veya sonuna rasgelse dahi bu aralığı korumak istiyorsanız, yıldızlı `\hspace*` komutu kullanırsınız.

`A\hspace{2cm}B\`  
`A \hspace{2cm} B`

Komutlardan önce veya sonra boşluk bırakmak farklı sonuçlar üretir.

Sınırsız bir yatay aralık oluşturmak için `\hfill` komutu kullanılır. Bu komuttan sonra gelen her şey satırın sonuna yaslanır. Hem satır sonuna yaslamak hem de aralığı noktalarla doldurmak isterseniz `\dotfill` komutunu kullanırsınız. Satır sonuna yaslayıp aralığa çizgi çekmek isterseniz de `\hrulefill` komutunu kullanırsınız.

`A\hfill B\`  
`A\dotfill B\`  
`A\hrulefill B`

## 1.15 Metni Hizalamak ve Sütunlara Bölmek

### 1.15.1 Hizalama

LaTeX’de metni sola hizalamak için `flushleft`, sağa hizalamak için `flushright` ve ortalı hizalamak için `center` ortamları kullanılır.

```
\begin{flushleft}
  burası sola hizalı
\end{flushleft}
\begin{flushright}
  sağa hizalı
\end{flushright}
\begin{center}
  ve ortalı
\end{center}
```

### 1.15.2 Sütunlara bölmek

LaTeX’de belgenin tamamının iki sütun dizilmesi için `\documentclass` komutunun seçeneğine `twocolumn` yazılabileceğinden Bölüm 1.5’de bahsettik. Bu, tüm belgenin iki sütun dizilmesini sağlar. Bazı sayfaları iki, bazılarınıysa tek sütun dizmek istiyorsanız `\twocolumn` ve `\onecolumn` komutlarını kullanmanız gerekir. `\twocolumn` komutunun verildiği sayfadan sonraki sayfalar iki, `\onecolumn` komutunun verildiği sayfadan sonraki sayfalar tek sütun dizilir.

Eğer metni daha fazla sütuna bölmek ve sütunları istediğiniz yerden başlatmak gibi daha fazla seçenek istiyorsanız, `multicols` ortamını kullanmanız gerekir. Bu ortamı kullanabilmek için

```
\usepackage{multicol}
```

komutuyla **multicol**<sup>21</sup> paketini eklemelisiniz.

```
\begin{multicols}{<sütun sayısı>}

\end{multicols}
```

Burada, `<sütun sayısı>` değişkeninde oluşturulmak istenen sütun adedi sayıyla belirtilir.

Bu ortamda sütun genişlikleri eşit olup, sütunlar arası boşluk `\columnsep`, sütunlar arasındaki çizginin kalınlığı `\columnseprule` ve sütunlar arasındaki çizginin rengi `\columnseprulecolor` komutlarında saklıdır. Bu değişkenler `\setlength` ya da `\def` komutları kullanılarak değiştirilebilir.

---

<sup>21</sup><http://ftp.ntua.gr/mirror/ctan/macros/latex/required/tools/multicol.pdf>

```
\setlength{\columnsep}{1cm}
\setlength{\columnseprule}{1pt}
\def\columnseprulecolor{\color{blue}}
```

Yukarıdaki birinci komutla sütunlar arasındaki boşluk 1 cm, çizgi kalınlığı 1 pt ve çizgi rengi mavi olarak düzenlenir. Bu komutlar ya sahanlığa ya da ortamı kullanmadan önce gövdeye yazılmalıdır.

Şimdiye kadar renk kullanımından bahsetmedik ancak çizgi rengini değiştirmek için verilen komutun kullanılabilmesi için sahanlığa `\usepackage{color}` komutuyla `color` paketinin eklenmesi gerekir.

Ortam isteğe bağlı bir değişken de alabilir. Bu, çengelli parantezlerden sonra köşeli parantezler içine yazılır. Köşeli parantezler içinde yazılanlar bölünmeden ve çok sütunlu metnin üstünde dizilir.

```
\begin{multicols}{2}
[\section{Başlık}
Burası sütunlara bölünmez.]
Burası sütunlara bölünür.
\end{multicols}
```

Sütünü kesmek için `\columnbreak` komutu kullanılır. Komutun verildiği yerde sütun kesilir, ardından kesme noktasından önceki paragraflar tüm kullanılabilir alanı doldurmak için eşit olarak dağıtılır. Dolayısıyla bazen beklenen sonucu vermeyebilir.

Varsayılan `multicols` ortamında sütunların her biri aynı miktarda metin içerecek şekilde dengelenmiştir. Bu, ortamın yıldızlı sürümü (`multicols*`) kullanılarak değiştirilebilir.

## 1.16 Listeleme

### 1.16.1 Temel listeler

LaTeX’de listeleme için değişik ortamlar vardır. Bu ortamlar tek başına kullanılabilceği gibi birlikte de kullanılabilirler. Her ortamda maddeler `\item` komutuyla belirtilir.

Bir listeyi numaralı şekilde dizmek için `enumerate` ortamı kullanılır.

```
\begin{enumerate}
\item madde 1
\begin{enumerate}
\item alt madde 1
\begin{enumerate}
\item en alt madde 1
```

## 1 Genel

```
\end{enumerate}
\item alt madde 2
\end{enumerate}
\item madde 2
\end{enumerate}
```

- 
1. madde 1
    - (a) alt madde 1
      - i. en alt madde 1
    - (b) alt madde 2
  2. madde 2
- 

Numarasız, özel işaretli listeler için `itemize` ortamı kullanılır ve bu ortamda madde işareti değiştirilebilir.

```
\begin{itemize}
\item madde 1
\item madde 2
\item[$\circ$] madde 3
\item[+] madde 4
\end{itemize}
```

Açıklamalı bir liste içinse `description` ortamı kullanılır. Bu ortamda köşeli parantez içine alınan anahtar kelimeler kalın dizilir.

```
\begin{description}
\item[Nokta] Boyutu olmayan
\item[Çember] Bir noktaya eşit
uzaklıktaki noktaların geometrik yeri
\end{description}
```

### 1.16.2 Listeleri özelleştirmek

Listelerin özelleştirmek için **enumerate**<sup>22</sup> paketi kullanılabilir. Paketi `\usepackage{enumerate}` komutuyla ekledikten sonra `enumerate` ortamını başlatan komutun peşine köşeli parantezler içinde madde işaretlerinin tipi belirtilebilir:

---

<sup>22</sup><http://ftp.ntua.gr/mirror/ctan/macros/latex/required/tools/enumerate.pdf>



```
\begin{enumerate}[I.]
\item bir
\item iki
\item üç
\end{enumerate}
```

Bunun dışında çok daha fazla özelleştirmeye izin veren **enumitem**<sup>23</sup> paketi vardır. Bu paketi kullanarak yapılabilecek listelere de ilerde değineceğiz. Dileyen okur paket belgesini inceleyip listelerini özelleştirebilir.

## 1.17 Yazıtipleri

### 1.17.1 Giriş

Yazıtipi konusu *kodlama*, *aile*, *biçem* ve *boyut* olmak üzere dört alt başlıkta incelenebilir. Kodlama çok teknik bir konu olup amacımız dışındadır, ancak sadece şunu belirtelim ki kodlama işini LaTeX’de Bölüm 1.6’de bahsettiğimiz **fontenc** paketi üstlenir. Bu paketi belgenize eklemiş olduğunuzu varsayarak devam edeceğiz.

Bu yazıda anlatacağımız şeylerden bazıları bir bakıma LaTeX’in felsefesine aykırı olacak. Nitekim LaTeX, `\documentclass` komutunda belirtilen ana yazıtipi boyutuna göre, dipnot ya da başlık gibi ana yazıtipi boyutundan farklı dizilen şeylerin boyutunu olabilecek en güzel ve doğru şekilde ayarlar. O yüzden bu konudaki klasik uyarıyı biz de yineleyelim:

**Dikkat!** *B eL* genizin içinde ne kadar çok yazı tipi kullanırsanız, *eseriniz* daha GÜZEL ve *okunaklı olacaktır*.

### 1.17.2 Aile

Yazıtipleri Roman ya da Serif, Sans Serif ve Typewriter olmak üzere üç ailede toplanabilir. Roman ailesi tırnaklı ya da süslü diyebileceğimiz yazıtiplerini, Sans Serif ailesi tırnaksız ya da süssüz yazıtiplerini ve Typewriter ailesi de daktilo yazıtiplerini barındırır.

LaTeX’de her belge sınıfı varsayılan yazıtipi ailesiyle gelir. **beamer** sınıfının varsayılan ailesi Sans Serif olup, diğer sınıfların varsayılan ailesi Roman’dır.

<sup>23</sup><http://ftp.cc.uoc.gr/mirrors/CTAN/macros/latex/contrib/enumitem/enumitem.pdf>

## 1 Genel

Varsayılan aile `\familydefault` komutunda saklı olup, `\renewcommand` komutuyla değiştirilebilir.

```
\renewcommand{\familydefault}{\rmdefault}
\renewcommand{\familydefault}{\sfdefault}
\renewcommand{\familydefault}{\ttdefault}
```

Birinci komut sahanlığa yazılırsa, belge sınıfından bağımsız olarak varsayılan aile Roman, ikincisi yazılırsa Sans Serif, üçüncüsü yazılırsa Typewriter olur.

Eğer belgenin tamamının değilde bazı kelime ya da cümlelerin farklı aileden yazılması istenirse –ki genelde böyle kullanılır– aşağıdaki komut ya da bildirimler kullanılır.

Komut	Bildirim	Aile
<code>\textrm</code>	<code>\rmfamily</code>	Roman (Serif)
<code>\textsf</code>	<code>\sffamily</code>	Sans Serif
<code>\texttt</code>	<code>\ttfamily</code>	Typewriter

Şekil 1.2: Yazıtipi Aileleri

LaTeX’de varsayılan yazıtipi Computer Modern olup, ek bir pakete ihtiyaç duymadan kullanılabilecek yazıtipleri aşağıda gösterilmiştir.

Kısaltma	Yazı Tipi
cmr	Computer Modern Roman (varsayılan)
lmr	Latin Modern Roman
pbk	Bookman
bch	Charter
pnc	New Century Schoolbook
ppl	Palatino
ptm	Times
put	Utopia
ccr	Computer Concrete
fi4	Inconsolata

Şekil 1.3: Roman Yazıtipleri

Kısaltma	Yazı Tipi
cmss	Computer Modern Sans Serif (varsayılan)
lmss	Latin Modern Sans Serif
pag	Avant Garde
phv	Helvetica
fvs	Bitstream Vera Sans

Şekil 1.4: Sans Serif Yazıtipleri

Kısaltma	Yazı Tipi
cmtt	Computer Modern Typewriter (varsayılan)
lmtt	Latin Modern Typewriter
pcr	Courier

Şekil 1.5: Typewriter Yazıtipleri

Kısaltma	Yazı Tipi
pzc	<i>Zapf Chancery</i>

Şekil 1.6: Elyazısı

Varsayılan yazıtipleri `\rmdefault`, `\sfdefault` ve `\ttdefault` komutlarında saklı olup, `\renewcommand` komutuyla değiştirilebilirler.

```
\renewcommand{\rmdefault}{<kısaltma>}
```

Burada `<kısaltma>`, tablolarda belirtilen kısaltmalardır. Örneğin

```
\renewcommand{\rmdefault}{put}
```

komutu sahanlığa yazıldığında, eğer varsayılan aile Roman ise belgenizin ana yazıtipi Utopia olur.

Eğer tüm belgenin değil, bazı kelime ya da cümlelerin farklı yazıtipinde yazılması istenirse `\fontfamily` komutuyla `\selectfont` komutu birlikte aşağıdaki şekilde kullanılır.

```
{\fontfamily{pbk}\selectfont Bookman yazıtipi.} Ana yazıtipi.
```

Varsayılan yazıtipi paket ekleyerek de değiştirilebilir. Bu hem pratiktir hem de bazı paketler matematiksel ifadelerin yazıtipine de etki eder. Bu paketlerin bazıları tabloda gösterilmiştir.

Bunların dışında beğenebileceğiniz birçok yazıtipini LaTeX Yazıtipi Kataloğu<sup>24</sup>'nda bulabilirsiniz.

### 1.17.3 Biçem

Metin içinde kelimeleri bazen italik bazen de kalın dizmek isteyebilirsiniz. Bu değişimler aşağıdaki tablodaki komut ya da bildirimlerle yapılır.

<sup>24</sup><https://tug.org/FontCatalogue/>

## 1 Genel

Paket	Roman	Matematik	Sans Serif	Typewriter
	CM Roman	$f(x) = x^2$	CM Sans	CM Typewriter
lmodern	LM Roman	$f(x) = x^2$	LM Sans	LM Typewriter
mathpazo	Palatino	$f(x) = x^2$		
mathptmx	Times	$f(x) = x^2$		
times	Times		Helvetica	Courier
helvet			Helvetica	
courier				Courier
bookman	Bookman		Avant Garde	Courier
newcent	New Century Schoolbook		Avant Garde	Courier
charter	Charter			
fourier	Utopia	$f(x) = x^2$		
eulervm		$f(x) = x^2$		

Şekil 1.7: Yazıtipi değiştiren paketler

Komut	Bildirim	Çıktı
<code>\textnormal</code>	<code>\normalfont</code>	Normal (varsayılan)
<code>\emph</code>	<code>\em</code>	Vurgulu
<code>\textit</code>	<code>\itshape</code>	İtalik
<code>\textsc</code>	<code>\scshape</code>	BAŞ HARFLER BÜYÜK
<code>\textsl</code>	<code>\slshape</code>	Eğik
<code>\textup</code>	<code>\upshape</code>	Dik (varsayılan)
<code>\textbf</code>	<code>\bfseries</code>	Kalın
<code>\textmd</code>	<code>\mdseries</code>	Orta Kalın (varsayılan)

Şekil 1.8: Yazıtipi Biçemleri

İzleyen kelime `\textit{italik}`  
harflerle yazılmıştır.  
Metnin geri kalan kısmı  
normaldir.

İzleyen ifade `{\slshape {\bfseries eğik kalındır}}`.

İzleyen ifade `\textit{\textbf{italik kalın}}`, ama bu  
`\textsc{\textit{büyük küçük harf değil}}`.

Eğer vurgulu metin içinde bazı kelimeler tekrar vurgulanırsa bu kelimeler normale döner.

```
{\em Vurgulu metinde tekrar
vurgu yapılırsa {\em normale}
döner.}
```

LaTeX’de vurgu yukarıdaki gibi yapılsa da altını çizerek vurgu yapmak isteyen olabilir. Kuyruklu harflerin altı çizildiğinde varsayılan satır aralığı değiştiğinden vurguyu bu

şekilde yapmamanız daha doğrudur. Ancak illa altını çizmek isterseniz `\underline` komutunu kullanabilirsiniz.

#### 1.17.4 Boyut

Yazıtipi boyutunu değiştirmek için aşağıdaki bildirimler kullanılır.

Bildirim	Çıktı
<code>\tiny</code>	Örnek
<code>\scriptsize</code>	Örnek
<code>\footnotesize</code>	Örnek
<code>\small</code>	Örnek
<code>\normalsize</code>	Örnek
<code>\large</code>	Örnek
<code>\Large</code>	Örnek
<code>\LARGE</code>	Örnek
<code>\huge</code>	Örnek
<code>\Huge</code>	Örnek

Şekil 1.9: Yazıtipi Boyutu Değiştiren Bildirimler

`{\Large Büyük}` ve `{\scriptsize küçük}` harfler.

Bu bildirimlerin aynı zamanda satır aralığını da değiştirdiğine dikkat edilmelidir. Aşağıdaki iki örnekte, `\par` (paragraf) komutunun verdiğiniz yere bağlı olarak farklı sonuçlar ürettiği gösterilmiştir. Doğru kullanım ikincisidir.

```
{\large
Sokrates: Platon
yalan söyleyecek
aşağıdaki cümlede.}\par
```

```
{\large Platon: Sokrates
doğruyu söyledi
önceki cümlede.\par}
```

## 1 Genel

	10pt (varsayılan)	11pt seçeneği	12pt seçeneği
<code>\tiny</code>	5pt	6pt	6pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\small</code>	9pt	10pt	11pt
<code>\normalsize</code>	10pt	11pt	12pt
<code>\large</code>	12pt	12pt	14pt
<code>\Large</code>	14pt	14pt	17pt
<code>\LARGE</code>	17pt	17pt	20pt
<code>\huge</code>	20pt	20pt	25pt
<code>\Huge</code>	25pt	25pt	25pt

Şekil 1.10: Yazıtipleri Mutlak Boyutları

Bu bildirimlerin etkisi belge ana yazıtipi boyutuna bağlıdır. Mutlak boyutlar aşağıdaki tabloda gösterilmiştir.

Bağımsız bir yazıtipi boyutu elde etmek için `\fontsize` ile `\selectfont` komutları birlikte kullanılır.

```
{\fontsize{<boyut>}{<aralık>}\selectfont <metin>}
```

Buradaki `<boyut>` yazıtipi boyutu, `<aralık>` ise satır aralığıdır. İkisinin de ölçü birimi punto (pt) olup, temel kural, aralığın boyutun 1.2 katı olmasıdır.

```
{\fontsize{30}{36}\selectfont
```

Yazı tipi boyutu 30 punto,  
satır aralığı 36 punto.}

Ana yazıtipi boyutu `\normalsize` komutunda saklı olup, `\renewcommand` komutuyla değiştirilebilir.

```
\renewcommand{\normalsize}{\fontsize{30}{36}\selectfont}
```

Yukarıdaki komutu sahanlığa yazarsanız belgenizin ana yazıtipi boyutu 30 pt, satır aralığı ise 36 pt olur.

## 1.18 Renk Kullanmak

LaTeX’de renkler `color`<sup>25</sup> ya da `xcolor`<sup>26</sup> paketleriyle kullanılır. İkinci paket daha güçlüdür.

```
\usepackage{xcolor}
```

<sup>25</sup><http://ftp.cc.uoc.gr/mirrors/CTAN/macros/latex/required/graphics/color.pdf>

<sup>26</sup><http://ftp.ntua.gr/mirror/ctan/macros/latex/contrib/xcolor/xcolor.pdf>

komutuyla paketi ekledikten sonra `\color` ya da `\textcolor` komutları aşağıdaki şekillerde kullanılır.

```
\color{red}{Kırmızı}\\
{\color{blue} Tamamı mavi}\\
\textcolor{pink}{Pembe}
```

Kullanılabilecek renk adları black, white, gray, darkgray, lightgray, brown, red, green, blue, cyan, magenta, lime, olive, orange, pink, purple, teal, violet ve yellow'dur.

Eğer `\documentclass` komutunun seçeneğine `dvipsnames` yazılırsa aşağıdaki renkler de kullanılabilir duruma gelir.

Apricot		Aquamarine		Bittersweet		Black		Blue	
BlueGreen		BlueViolet		BrickRed		Brown		BurntOrange	
CadetBlue		CarnationPink		Cerulean		CornflowerBlue		Cyan	
Dandelion		DarkOrchid		Emerald		ForestGreen		Fuchsia	
Goldenrod		Gray		Green		GreenYellow		JungleGreen	
Lavender		LimeGreen		Magenta		Mahogany		Maroon	
Melon		MidnightBlue		Mulberry		NavyBlue		OliveGreen	
Orange		OrangeRed		Orchid		Peach		Periwinkle	
PineGreen		Plum		ProcessBlue		Purple		RawSienna	
Red		RedOrange		RedViolet		Rhodamine		RoyalBlue	
RoyalPurple		RubineRed		Salmon		SeaGreen		Sepia	
SkyBlue		SpringGreen		Tan		TealBlue		Thistle	
Turquoise		Violet		VioletRed		White		WildStrawberry	
Yellow		YellowGreen		YellowOrange					

Şekil 1.11: `dvipsnames` seçeneğiyle gelen renkler

Bunlar da yeterli gelmiyorsa seçeneklere bir de `svgnames` yazın ve buradaki<sup>27</sup> yüzden fazla rengi kullanılabilir duruma getirin.

```
\documentclass[svgnames,dvipsnames]{article}
```

<sup>27</sup><https://www.latextemplates.com/svgnames-colors>

### 1.18.1 Yeni renkler oluşturmak

LaTeX'in size hazır olarak verdiği renkler yeterli gelmezse kendi renklerinizi oluşturabilirsiniz. Bu, iki şekilde yapılabilir.

Hazır renkler istenilen oranlarda karıştırılarak yeni renkler elde edilebilir. Aşağıda karışımın nasıl hazırlanabileceği gösterilmiştir.

```
\color{blue!80}{\%80 mavi \%20 beyaz} \\
\color{red!40!blue}{\%40 kırmızı \%60 mavi} \\
{\color{yellow!20!green!75!black} \%20x0.75=15 sarı
\%(100-20)x0.75=60 yeşil \%100-75=25 siyah}
```

İkinci yol `\definecolor` komutuyla renk modelleri kullanmaktır.

**Tablo 1.7:** Renk Modelleri

Model	Açıklama
gray	Grinin tonları (0-1). 0 (siyah), 1 (beyaz). Yani 0.95 çok açık gri, 0.30 koyu gri olur.
rgb	Red, Green, Blue (0-1). Üç renk (kırmızı, yeşil, mavi) 0 ile 1 arasında bir sayıyla temsil edilir.
RGB	Red, Green, Blue (0-255). Üç renk (kırmızı, yeşil, mavi) 0 ile 255 arasında bir sayıyla temsil edilir.
HTML	Red, Green, Blue (OO-FF). Kodlar RRGGBB şeklinde verilir.
cmk	Cyan, Magenta, Yellow, Black (0-1). Dört renk (camgöbeği, eflatun, sarı, siyah) 0 ile 1 arasında bir sayıyla temsil edilir.

Komut

```
\definecolor{<isim>}{<renk modeli>}{<kod>}
```

şeklinde verilir. Burada `<isim>`, sizin daha sonra kullanmak üzere vereceğiniz bir isimdir. Kodlar için HueSnap<sup>28</sup> adresinden yararlanabilirsiniz. Örneğin

```
\definecolor{renk1}{gray}{0.50}
\definecolor{renk2}{rgb}{1,0.7,0.3}
\definecolor{renk3}{RGB}{125,32,200}
\definecolor{renk4}{HTML}{CC33CC}
\definecolor{renk5}{cmk}{0,0.7,1,0.5}
```

<sup>28</sup><https://www.huesnap.com/color>



komutlarıyla renk1, renk2, renk3, renk4 ve renk5 adında beş adet renk tanımlanmış olur. Bu komutlar ya sahanlığa ya da bu renkleri kullanacağınız satırdan önce gövdeye yazılmalıdır.

Yeni oluşturduğunuz rengi tek seferlik kullanacaksanız

```
\color[<model>]{<kod>}
\textcolor[<model>]{<kod>}{<metin>}
```

komutlarını kullanabilirsiniz.

```
{\color[RGB]{102,0,51} Bol}\
\textcolor[RGB]{0,76,153}{Çeşit bol}
```

Metnin arka planını renklendirmek için \colorbox komutu kullanılır. Benzer \fcolorbox komutu aynı şeyi çizgi çizerek yapar.

```
\colorbox{<arka plan rengi>}{<metin>}
\colorbox[<renk modeli>]{<arka plan rengi>}{<metin>}
\fcolorbox{<çizgi rengi>}{<arka plan rengi>}{<metin>}
\fcolorbox[<renk modeli>]{<çizgi rengi>}{<arka plan rengi>}{<metin>}
```

Komutlar yukarıda gösterildiği şekilde kullanılabilir.

```
\colorbox{Cyan}{Metin}
\colorbox[rgb]{0.4,0.4,0.5}{Metin}
\fcolorbox{red}{yellow}{Metin}
\fcolorbox[RGB]{0,0,0}{255,204,255}{Metin}
```

Sayfaları renklendirmek için ise \pagecolor komutu kullanılır. Komutun zorunlu değişkeninde renk belirtilir. Tekrar normale (beyaz) dönmek için \nopagecolor komutu kullanılır. Eğer bu işe yaramazsa \pagecolor{white} komutunu kullanabilirsiniz.



## 2 Matematiksel İfadeler

Matematik formüllerini dizmek, kuşkusuz, LaTeX'in en güçlü olduğu konulardan biridir. Çok fazla matematiksel gösterimin varlığından dolayı da büyük bir konudur. Bu bölümde ileri bir matematik kitabını dizmek için gereken birçok şey anlatılacaktır ancak işin sınırları göz önüne alındığında başka kaynaklara da başvurmanız gerekebilir.

### 2.1 Giriş

Belgenizde yalnızca birkaç basit matematiksel formül kullanacaksanız herhangi bir pakete gerek olmadan yazabilirsiniz. Ancak çok sayıda karmaşık formül içeren bilimsel bir belge yazma niyetindeyseniz temel AMS<sup>1</sup> paketlerini kullanmanız gerekir. Bu paketler **amsmath**<sup>2</sup>, **amssymb**<sup>3</sup> ve **amsfonts**<sup>4</sup> dir.

```
\usepackage{amsmath,amssymb,amsfonts}
```

Yukarıdaki komutu sahanlığa yazarak paketleri belgenize ekleyiniz. Bundan sonra bu paketleri eklediğinizi varsayarak devam edeceğiz ve bunların dışında bir pakete ihtiyaç duyarsak ayrıca belirteceğiz.

#### 2.1.1 Genel

Belgenizin metnini oluştururken LaTeX'in metnin ne zaman matematiksel olduğunu bilmesi gerekir. Bunun nedeni, LaTeX'in matematiksel ifadeleri normal metinden farklı bir şekilde dizmesidir. Bu nedenle matematiksel ifadeler, normal metinden farklı olarak bazı ortamlarda girilirler.

Matematik özel ortamlar gerektirdiğinden, doğal olarak standart şekilde kullanabileceğiniz uygun ortam adları vardır. Bununla birlikte, diğer ortamların çoğundan farklı olarak, formülünüzü bildirmek için bazı kullanışlı kısaltmalar vardır. LaTeX'de bu ortamlar ya da kısaltmalar kullanılarak formüller iki türlü dizilir:

---

<sup>1</sup><https://www.ams.org/publications/authors/tex/amslatex>

<sup>2</sup><http://ftp.ntua.gr/mirror/ctan/macros/latex/required/amsmath/amsmath.pdf>

<sup>3</sup><https://texdoc.net/texmf-dist/doc/fonts/amsfonts/amssymb.pdf>

<sup>4</sup><http://ftp.ntua.gr/mirror/ctan/fonts/amsfonts/doc/amsfonts.pdf>

## 2 Matematiksel İfadeler

- Formüller satırın içinde, yani bildirildiği metnin gövdesi içine yazılır:  $\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$ . Görüldüğü gibi LaTeX, paragraf yapısını bozmamak için sembolleri olabildiğince sıkıştırır ve gerek görürse alttakileri yana kaydırır.
- Formüller ayrı bir satırda tek başlarına tüm detaylarıyla sergilenir:

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}.$$

Formülün satır içerisinde dizilmesi için ya  $\$...\$$  arasına arasına, ya  $\backslash(...\backslash)$  arasına ya da  $\backslash\begin{math}$  ile  $\backslash\end{math}$  arasına, yani  $\text{math}$  ortamında yazılması gerekir. Üçü de aynı sonucu verir.

Formülün sergilenmesi içinse ya  $\backslash[... \backslash]$  arasına ya  $\text{displaymath}$  ortamında ya da  $\text{equation}$  ortamında yazılması gerekir.  $\text{equation}$  ortamında yazılan formülü LaTeX otomatik numaralandırır. Numara verilmesini istemezseniz ortamı  $\text{equation*}$  şeklinde kullanmanız gerekir.

TeX'in eski sürümlerinde formüller, sergilenmeleri için  $\$... \$$  arasına yazılırdı. Bu kullanım hala geçerlidir ancak bazı sorunlara yol açabildiğinden (örneğin belge seçeneğine  $\text{fleqn}$  yazıldığında) kullanımı önerilmez.

```
 $\$x\in\mathbb{R}\$$  için  $|x|<1$  ise  
 $\begin{equation*}$   
     $-1<x<1$   
 $\end{equation*}$   
olur.
```

$x \in \mathbb{R}$  için  $|x| < 1$  ise

$$-1 < x < 1$$

olur.

Numara verilen bir formülü  $\backslash\text{label}$  komutuyla etiketleyip,  $\backslash\text{ref}$  ya da  $\backslash\text{eqref}$  komutuyla formüle atıf yapılabilir. Atıf  $\backslash\text{eqref}$  komutuyla yapılırsa formülün numarası parantez içinde yazılır.

```
 $\begin{equation}$   
 $\backslash\text{label}\{eq:euler\}$   
     $e^{i\pi}+1=0$   
 $\end{equation}$   
Euler'in  $\backslash\text{eqref}\{eq:euler\}$  formülüne göre\dots
```

$$e^{i\pi} + 1 = 0 \tag{2.1}$$

Euler'in (2.1) formülüne göre...

### 2.1.2 Matematik Kipiyle Metin Kipi Arasındaki Farklar

Matematiksel ifadeleri girerken düz metin girişinden farklı olarak dikkat edilmesi gereken bazı noktalar vardır:

1. Boşlukların ve satır kesmelerinin çoğunun önemi yoktur, çünkü tüm boşluklar ya matematiksel ifadelerden mantıksal olarak türetilir, ya da özel komutlarla belirtilmesi gerekir.
2. Boş satırlara izin verilmez.
3. Aksanlı harfler kullanılmaz.
4. Her harf matematiksel bir değişken olarak kabul edilir ve italik dizilir. Eğer düz yazıyla ve normal aralıklarla bir metin girilecekse `\textrm{}` ya da `\text{}` komutları kullanılmalıdır. Bu komutlarla metin kipine geçiş yapılmış olur ve metin artık düz ve normal aralıklarla dizilir. İtalik ve normal aralıklarla metin girilecekse `\textit{}` komutu kullanılabilir. Ayrıca bu komutlarla aksanlı harfler de kullanılabilir.

Aşağıdaki örnek matematik kipi ile metin kipi arasındaki farkları gösterir.

`$x^2-1=0 \text{ ise } x=\pm 1$.`

$$x^2 - 1 = 0 \text{ ise } x = \pm 1.$$

### 2.1.3 Gruplandırma

Formülleri dizerken dikkat edilmesi gereken noktalardan biri komutların çoğunun kendisinden sonra ilk gelen karaktere etki etmesidir. Bu yüzden bir komutun çok sayıda karaktere etki etmesi istenirse bu karakterler iki çengelli parantez `{...}` arasına yazılarak gruplandırılmalıdır.

`$a^{x+y} \neq a^{x+y}$`

$$a^x + y \neq a^{x+y}$$

## 2.2 Parantezler, Gruplandırıcılar ve Oklar

LaTeX’de her türlü parantez ve gruplandırıcı kullanılabilir. Yuvarlak ve köşeli parantezler klavyedeki yerlerinden, çengelli parantez ise `\{` ve `\}` komutları kullanılarak girilir.

`$(a,b,c) \neq \{a,b,c\}$`



Tablo 2.2: Oklar

$\leftarrow$ <code>\leftarrow</code> ya da <code>\gets</code>	$\longleftarrow$ <code>\longleftarrow</code>	$\uparrow$ <code>\uparrow</code>	$\Leftarrow$ <code>\Leftarrow</code>
$\Leftrightarrow$ <code>\Leftrightarrow</code>	$\Uparrow$ <code>\Uparrow</code>	$\rightarrow$ <code>\rightarrow</code> ya da <code>\to</code>	$\longrightarrow$ <code>\longrightarrow</code>
$\downarrow$ <code>\downarrow</code>	$\Rightarrow$ <code>\Rightarrow</code>	$\Longrightarrow$ <code>\Longrightarrow</code>	$\Downarrow$ <code>\Downarrow</code>
$\leftrightarrow$ <code>\leftrightarrow</code>	$\longleftrightarrow$ <code>\longleftrightarrow</code>	$\Uparrow$ <code>\Uparrow</code>	$\Leftrightarrow$ <code>\Leftrightarrow</code>
$\Leftrightarrow$ <code>\Leftrightarrow</code>	$\Updownarrow$ <code>\Updownarrow</code>	$\mapsto$ <code>\mapsto</code>	$\longmapsto$ <code>\longmapsto</code>
$\nearrow$ <code>\nearrow</code>	$\hookrightarrow$ <code>\hookrightarrow</code>	$\hookrightarrow$ <code>\hookrightarrow</code>	$\searrow$ <code>\searrow</code>
$\leftharpoonup$ <code>\leftharpoonup</code>	$\rightharpoonup$ <code>\rightharpoonup</code>	$\swarrow$ <code>\swarrow</code>	$\leftharpoondown$ <code>\leftharpoondown</code>
$\rightarrow$ <code>\rightarrow</code>	$\nwarrow$ <code>\nwarrow</code>	$\rightleftharpoons$ <code>\rightleftharpoons</code>	$\leadsto$ <code>\leadsto</code>
$\rightharpoondown$ <code>\rightharpoondown</code>			

```
\(
\downarrow
\big\downarrow
\Big\downarrow
\bigg\downarrow
\Bigg\downarrow
\)
```



Bunların dışında altlarına ya da üstlerine matematiksel ifadeler yazılabilen `\xleftarrow` ve `\xrightarrow` komutları vardır.

```
\(
\xleftarrow{a}
\xrightarrow[X]{a+b}
\)
```

$$\xleftarrow[a]{a+b}$$

## 2.3 Yunan Harfleri

Yunan harfleri matematikte yaygın olarak kullanılır. Bu harfler ters eğik çizgiden sonra harfin adı yazılarak elde edilir. Eğer ilk harf küçük ise küçük, büyükse de büyük harf

## 2 Matematiksel İfadeler

elde edilir. Bazı büyük Yunanca harfler Latin harfleri gibi görüldüğünden (örneğin, büyük harf Alpha ve Beta yalnızca sırasıyla “A” ve “B”dir) ayrıca tanımlanmamışlardır. Küçük harf epsilon, theta, kappa, phi, pi, rho ve sigma iki farklı sürümde sunulmaktadır. Alternatif sürümü, harf adının önüne “var” eklenerek oluşturulur.

**Tablo 2.3:** Yunan Harfleri

$\alpha$ \alpha	$\theta$ \theta	$o$ o	$\tau$ \tau
$\beta$ \beta	$\vartheta$ \vartheta	$\pi$ \pi	$\upsilon$ \upsilon
$\gamma$ \gamma	$\iota$ \iota	$\varpi$ \varpi	$\phi$ \phi
$\delta$ \delta	$\kappa$ \kappa	$\rho$ \rho	$\varphi$ \varphi
$\epsilon$ \epsilon	$\lambda$ \lambda	$\varrho$ \varrho	$\chi$ \chi
$\varepsilon$ \varepsilon	$\mu$ \mu	$\sigma$ \sigma	$\psi$ \psi
$\zeta$ \zeta	$\nu$ \nu	$\varsigma$ \varsigma	$\omega$ \omega
$\eta$ \eta	$\xi$ \xi		
$\Gamma$ \Gamma	$\Lambda$ \Lambda	$\Sigma$ \Sigma	$\Psi$ \Psi
$\Delta$ \Delta	$\Xi$ \Xi	$\Upsilon$ \Upsilon	$\Omega$ \Omega
$\Theta$ \Theta	$\Pi$ \Pi	$\Phi$ \Phi	

`\forall \epsilon > 0` için

$\forall \epsilon > 0$  için

## 2.4 Fonksiyonlar

LaTeX’de fonksiyonlar aşağıdaki komutlarla dizilirler.

`\arccos \cos \csc \exp \ker \limsup \min \sinh \arcsin \cosh \deg \gcd`  
`\lg \ln \Pr \sup \arctan \cot \det \hom \lim \log \sec \tan \arg`  
`\coth \dim \inf \liminf \max \sin \tanh`

Matematik kipinde fonksiyonlar diğer değişkenler gibi italik değil düz yazılırlar ve boşluklar otomatik ayarlanır.

`\sin x`, `\exp x`, `\log x`,  
`\det A`, `\min_{x \in A} f(x)`

$\sin x$ ,  $\exp x$ ,  $\log x$ ,  $\det A$ ,  $\min_{x \in A} f(x)$

Bunların dışında bir fonksiyon tanımlamak için `\DeclareMathOperator` komutu kullanılır.

`\DeclareMathOperator{\obeb}{obeb}`

Yukarıdaki komuttan sonra artık kullanabileceğiniz bir “obeb” fonksiyonu olur.



`\obeb(12,16)=4`

$\text{obeb}(12, 16) = 4$

Bu komutun sınır değerleri sağ taraf yerine alta dizen yıldızlı sürümü vardır: `\DeclareMathOperator*`. Örneğin `\DeclareMathOperator*\Max\Max` komutunu sahanlıkta verdikten sonra belgede kullanırsanız şöyle bir çıktı alırsınız:

```
\begin{equation*}
\Max_{x \in A} f(x)
\end{equation*}
```

$$\Max_{x \in A} f(x)$$

Modülo fonksiyonu içinse `\mod` ya da `\pmod` komutları verilir. İkinci komut fonksiyonu parantez içinde yazar.

`$a \equiv b \pmod p$ ise $p \mid a-b$'dir.`

$a \equiv b \pmod p$  ise  $p \mid a - b$ 'dir.

Limit için `\lim` komutu aşağıdaki şekilde verilir.

```
\lim_{<değişken> \to <değişken>}
```

Buradaki `\to` komutu  $\rightarrow$  üretir ve  $\infty$  için `\infty` komutu verilir.

```
\[
\lim_{x \to 0}
\frac{\sin x}{x} = 1 \quad \quad
\lim_{n \to +\infty} f_n = \delta
\]
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1 \quad \quad \lim_{n \rightarrow +\infty} f_n = \delta$$

## 2.5 Yığın Simgeleri

Matematikte bazen bir ifadenin altına ya da üstüne başka ifadeler yazmak gerekebilir. Bunlar yığın simgeleri olarak adlandırılırlar.

LaTeX'de aşağıdaki

```
\overset{<birinci değişken>}{<ikinci değişken>}
```

komutu birinci değişkendeki sembolü, normal boyda yazılan ikincinin üzerine yazar. `\underset` komutu ise alta yazar.

## 2 Matematiksel İfadeler

```
\[
\overset{R}{\sim}
\]
```

$$\overset{R}{\sim}$$

`\overline` ve `\underline` komutları bir ifadenin üstüne veya altına yatay bir çizgi çekerler.

```
\[
\overline{x+y}
\]
```

$$\overline{x+y}$$

`\overbrace` ve `\underbrace` komutları bir ifadenin üstüne veya altına yatay bir çengel atarlar.

```
\[
\underbrace{1+2+\dots+n}_{=} =
\frac{n(n+1)}{2}
+(n+1)
\]
```

$$\underbrace{1+2+\dots+n}_{=\frac{n(n+1)}{2}}+(n+1)$$

`\overleftarrow` komutu ifadenin üstüne sola, `\overrightarrow` ise sağa bir ok çizer. Bu komutlar vektörleri göstermek için kullanılabilir. Vektörler için `\vec` komutu da kullanılır.

```
\[
\overrightarrow{AB} \quad \vec{a}
\]
```

$$\overrightarrow{AB} \quad \vec{a}$$

`\stackrel` komutu `\overset` gibi davranır.

```
\[
\int \stackrel{!}{f_N(x)} = 1
\]
```

$$\int \stackrel{!}{f_N(x)} = 1$$

## 2.6 Matrisler

Temel matrisler `matrix` ortamında girilir. Bu ortamda elemanlar otomatik ortalanır ve sütunlar normal bir tablo gibi dizilir. Her sütun & karakteriyle ayrılır ve alt satıra geçmek için `\` komutu verilir.

```
\[
\begin{matrix}
a & b & c \\
d & e & f \\
g & h & i
\end{matrix}
\]
```

$$\begin{matrix} a & b & c \\ d & e & f \\ g & h & i \end{matrix}$$

Çeşitli matrisler dizmek için `matrix` ortamının farklı sürümleri kullanılır: `pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix` ve `Vmatrix`. Bu ortamlar sırasıyla yuvarlak, köşeli, çengelli, dikey çubuklu ve çift dikey çubuklu matrisler oluşturur.

```
\[
\begin{pmatrix}
1 & 2 \\
3 & 4
\end{pmatrix}
\]
```

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

```
\[
\begin{bmatrix}
1 & 2 \\
3 & 4
\end{bmatrix}
\]
```

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

## 2 Matematiksel İfadeler

```
\[
A=
\begin{bmatrix}
x_{11} & x_{12} & \dots \\
x_{21} & x_{22} & \dots \\
\vdots & \vdots & \ddots
\end{bmatrix}
```

$$A = \begin{bmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

Küçük bir matris yazmak için `smallmatrix` ortamı kullanılır. Bu matriste parantezler elle eklenmelidir.

```
Şimdi yazacağımız matris
\big(
\begin{smallmatrix}
a & b \\
c & d
\end{smallmatrix}
\big)
küçük bir matristir.
```

Şimdi yazacağımız matris  $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$  küçük bir matristir.

Bazı durumlarda, hizalamayı elle yapmak ve sütunlar veya satırlar arasına çizgi çekmek istenebilir. Bu durumda `tabular` ortamının matematik sürümü olan `array` ortamını kullanılmalıdır.

```
\[
\left(\begin{array}{r|r}
-1&2\\
3&-4
\end{array}\right)
```

$$\left(\begin{array}{c|c} -1 & 2 \\ \hline 3 & -4 \end{array}\right)$$

## 2.7 Yazı Biçem ve Boyutları

Matematiksel ifadeleri dizerken bazen yazının biçimini ya da boyutunu değiştirmek isteyebilirsiniz.

LaTeX’de matematik kipindeki yazıların biçimleri aşağıdaki komutlar kullanılarak değiştirilir.

**Tablo 2.4:** Matematik Kipinde Yazı Biçemleri

Komut	Görünüm
<code>\mathnormal{ABC def 123}</code>	$ABCdef123$
<code>\mathrm{ABC def 123}</code>	$ABCdef123$
<code>\mathit{ABC def 123}</code>	$ABCdef123$
<code>\mathbf{ABC def 123}</code>	$\mathbf{ABCdef123}$
<code>\mathtt{ABC def 123}</code>	$ABCdef123$
<code>\mathsf{ABC def 123}</code>	$ABCdef123$
<code>\mathfrak{ABC def 123}</code>	$\mathfrak{ABCdef123}$
<code>\mathbb{ABC}</code>	$\mathbb{ABC}$
<code>\mathcal{ABC}</code>	$\mathcal{ABC}$
<code>\mathscr{ABC}</code>	$\mathscr{ABC}$

Son satırdaki komutun kullanılabilmesi için

```
\usepackage{mathrsfs}
```

komutuyla **mathrsfs** paketi eklenmiş olmalıdır.

Bu komutlarla girilen ifadelerdeki boşluklar yine dikkate alınmaz ve yine aksanlı harfler girilemez.

Matematik kipindeki bir ifadenin hem kalın hem de italik yazılması için `\boldsymbol` komutu kullanılmalıdır.

```
\[
\mu, M \quad \mathbf{\mu}, \mathbf{M} \quad \mu, M
\boldsymbol{\mu}, \boldsymbol{M}
\]
```

$$\mu, M \quad \mathbf{\mu}, \mathbf{M} \quad \mu, M$$

Matematik kipindeki yazının boyutunu elle ayarlayabileceğiniz dört komut vardır: `\displaystyle`, `\textstyle`, `\scriptstyle` ve `\scriptscriptstyle`. `\textstyle` komutu ifadeyi normal metin boyutunda dizer, `\displaystyle` komutu ise ifadeyi ayrı satırda sergilenir gibi büyük dizer. `\scriptstyle` ve `\scriptscriptstyle` komutları da normal metin boyutundan küçük dizerler.

## 2 Matematiksel İfadeler

```
\[
\sum_{k=0}^n z^k \quad \textstyle\sum_{k=0}^n z^k
\]

$$\sum_{k=0}^n z^k$$


$$\sum_{k=0}^n z^k$$


$$\sum_{k=0}^n z^k$$

```

$$\sum_{k=0}^n z^k$$

$$\sum_{k=0}^n z^k$$

Kesirler dizilirken  $\{\displaystyle\frac{\dots}{\dots}\}$  ve  $\{\textstyle\frac{\dots}{\dots}\}$  komutları yerine onların kısaltmaları olan `\dfrac` ve `\tfrac` komutları kullanılabilir. Aynı şey `\binom` komutu için de geçerlidir.

```

$$\frac{1}{n} \log x$$


$$\frac{1}{n} \log x$$


$$\binom{n}{3}$$


$$\dbinom{n}{3}$$

\[
\frac{1}{n} \log x \quad \frac{1}{n} \log x \quad \binom{n}{3} \quad \binom{n}{3}
\]

$$\frac{1}{n} \log x$$


$$\frac{1}{n} \log x$$


$$\binom{n}{3}$$


$$\binom{n}{3}$$

\]
```

$$\frac{1}{n} \log x$$

$$\frac{1}{n} \log x$$

## 2.8 Boşluklar

Bazen LaTeX formülleri dizerken olması gerektiği gibi boşluk bırakma konusunda yetersiz kalabilir. Bu durumda boşluklar elle oluşturulur. Boşluklar için kullanılabilecek komutlar tablodaki gibidir.

Negatif	<code>\!</code>
İnce	<code>\,</code>
Orta	<code>\:</code>
Kalın	<code>\;</code>

Sözcük arası	<code>\</code>
Bir quad	<code>\quad</code>
İki quad	<code>\qquad</code>

Örneğin `\`, komutunun bıraktığı ince boşluk bazı formüllerde çok kullanışlıdır.

```
\[
\int_a^b f(x) dx \quad \sqrt{2} a
\quad \sqrt{\log x}
\]
```

```
\[
\int_a^b f(x) \, dx \quad \sqrt{2} a
\quad \sqrt{\log x}
\]
```

$$\int_a^b f(x) dx \quad \sqrt{2} a \quad \sqrt{\log x}$$

$$\int_a^b f(x) \, dx \quad \sqrt{2} a \quad \sqrt{\log x}$$

Negatif aralık bırakan `\!` komutu da fazla aralıklı ifadeleri birbirine yaklaştırmak için kullanılır.

```
\[
x^2/2 \quad a/\sin b
\]
```

```
\[
x^2\!/2 \quad a/\!\sin b
\]
```

$$x^2/2 \quad a/\sin b$$

$$x^2/2 \quad a/\sin b$$

## 2.9 Denklem Ortamları

Bir satıra sığmayacak kadar uzun bir formülü ya da birden çok satırdan oluşan bir denklemi veya denklem sistemlerini hizalayıp dizmek için LaTeX'de çeşitli ortamlar kullanılır.

`multline` ortamı bir formülü hizalanmamış bir kaç satıra ayırır.

## 2 Matematiksel İfadeler

```
\begin{multline}
f=a+b+c \\
+i+j+k+l \\
+x+y+z
\end{multline}
```

$$f = a + b + c + i + j + k + l + x + y + z \quad (2.2)$$

Bu ortamda ilk satır sola, son satır sağa ve kalanlar ortalı hizalanır. Denklemin numarası da son satırın sağına yazılır. Dekleme numara verilmesi istenmiyorsa ortam `multline*` şeklinde kullanılmalıdır.

`split` ortamı denklemi dikey hizalanmış birden çok satırda dizer.

```
\begin{equation}
\begin{split}
a&= b+c-d \\
&= e-f \\
&= g
\end{split}
\end{equation}
```

$$\begin{aligned} a &= b + c - d \\ &= e - f \\ &= g \end{aligned} \quad (2.3)$$

Hizalama & karakteriyle yapılır (genelde = işaretinden hemen önce kullanılır). Ortam mutlaka formülün numaralandırılmasından sorumlu ya da numara vermeyen başka bir matematik ortamında kullanılması gerekir.

`gather` ortamı birden fazla formülü birlikte gruplandırır, ortalar ve her birini ayrı bir satırda numaralandırır. Yine `gather*` ortamı, aynı türden numaralandırılmamış formüller üretir.

```
\begin{gather}
a=b+c \\
V+F-S=2
\end{gather}
```



$$a = b + c \quad (2.4)$$

$$V + F - S = 2 \quad (2.5)$$

`align` ortamı, iki veya daha fazla satırdan oluşan bir denklemi her bir satırı hizalı ve numaralı şekilde dizmek için kullanılır. Hizalama aynı şekilde `&` karakteriyle yapılır. Ortam yıldızlı (`align*`) şekilde kullanılırsa hiçbir satır numaralandırılmaz.

```
\begin{align}
  a& = b+c+d \\
  e& = f \\
  x-1& = y+z
\end{align}
```

$$a = b + c + d \quad (2.6)$$

$$e = f \quad (2.7)$$

$$x - 1 = y + z \quad (2.8)$$

`align` ortamı aynı zamanda birden fazla özerk formül dizisini birleştirmek için de kullanışlıdır. Bu durumda, `&` karakteri konumuna bağlı olarak hizalama ve ayırıcı olmak üzere iki farklı işlev üstlenir.

```
\begin{align*}
  a &=b & c&=d & e&=f \\
  u &=v & w&=x & y&=z
\end{align*}
```

$$\begin{array}{ccc} a = b & c = d & e = f \\ u = v & w = x & y = z \end{array}$$

`alignat` ortamı `align` ortamına benzer fakat sütun sayısını belirten bir değişken alır (Bir satırda kullanılan `&` sayısının bir fazlasının yarısı sütun sayısını vermelidir} ve denklemler arasındaki yatay boşluğun kontrolünü sağlar. Eğer boşluk komutlarından biri kullanılmazsa denklem sistemleri arasında boşluk bırakılmaz (örnekte boşluk komutu olarak `\qquad` kullanılmıştır).

```
\begin{alignat*}{3}
  a&=b\qquad & c&=d\qquad & e&=f \\
  u&=v\qquad & w&=x\qquad & y&=z
\end{alignat*}
```

## 2 Matematiksel İfadeler

$$\begin{array}{ccc} a = b & c = d & e = f \\ u = v & w = x & y = z \end{array}$$

`flalign` ortamı `align` ortamına benzer ancak ilk denklem sistemini sola ve son denklem sistemini sağa yaslar.

`aligned` ortamı yine `align` ortamına benzer fakat başka bir matematik ortamında kullanılması gerekir.

```
\[
\left.
\begin{aligned}
a&= b+1 \\
a&= 2b
\end{aligned}
\right\}
\quad a = 2 \text{ ve } b = 1.
\]
```

$$\left. \begin{array}{l} a = b + 1 \\ a = 2b \end{array} \right\} \quad a = 2 \text{ ve } b = 1.$$

`cases` ortamı parçalı fonksiyonları dizmek için kullanışlıdır. Ortamın içine yazılan denklemlerin solunda uygun boyda bir çengelli parantez açar. Sütunlar sola yaslıdır. Ortamın başka bir matematik ortamında kullanılması gerekir.

```
\[
n!=
\begin{cases}
1 & \text{\$n=0\$ ise} \\
n(n-1)! & \text{\$n\ge 1\$ ise}
\end{cases}
\]
```

$$n! = \begin{cases} 1 & n = 0 \text{ ise} \\ n(n-1)! & n \geq 1 \text{ ise} \end{cases}$$

Her satıra numara veren bir ortamda bazı satırların numarasız olması istenirse bu satırların sonuna `\notag` ya da `\nonumber` komutları verilir. `\tag` komutuyla ise keyfi bir numara ya da işaret yazılabilir.

```
\begin{align}
x&=y\\
z&=y+1 \notag\\
w&=3 \tag{*}
\end{align}
```

$$x = y \tag{2.9}$$

$$z = y + 1$$

$$w = 3 \tag{*}$$

Numaralı formüllere etiket yine `\label` komutuyla koyulur ve `\eqref` komutuyla atıf yapılır.

```
\begin{align}
a &= b+c \\
c &= d \label{eq:cd}\\
e &= f+g
\end{align}
Yukarıdaki \eqref{eq:cd}
formülü\dots
```

$$a = b + c \tag{2.10}$$

$$c = d \tag{2.11}$$

$$e = f + g \tag{2.12}$$

Yukarıdaki (2.11) formülü...

Aynı ortamda yer alan formüllerin 1, 2,... yerine 1.a, 1.b,... biçiminde numaralandırılması için ortamın `subequations` ortamının içine yazılması gerekir.

```
\begin{subequations}
\label{eq:sub}
\begin{align}
a &= b+c \\
c &= d \label{eq:subb} \\
e &= f+g
\end{align}
\end{subequations}
Formül~\eqref{eq:sub} ve
altformül~\eqref{eq:subb}\dots
```

## 2.10 Teorem ve Benzeri Ortamlar

Gerçek bir matematik kitabı karıştırdıysanız “Teorem 2.1”, “Sonuç 2.1.1” gibi ifadelerle başlayan paragraflara rastlamış olmalısınız. Bunlar metnin geri kalanından ayrılmış ve yanında sıralı sayılarla etiketli paragraflardır. Bu, matematikte teoremler için yaygın olarak kullanılır, ancak her şey için kullanılabilir.

LaTeX, herhangi bir teorem benzeri bildirimi kolayca tanımlamanıza izin veren bir komut sunar: `\newtheorem`.

### 2.10.1 Temel Teoremler

Öncelikle sahanlığa

```
\usepackage{amsthm}
```

komutuyla `amsthm` paketini ekleyiniz.

En basit kullanım

```
\newtheorem{<ad>}{<başlık>}
```

komutunun sahanlığa verilmesidir. İlk değişken olan `<ad>`, referans olarak kullanacağınız addır, ikinci değişken `<başlık>` ise LaTeX’in her kullandığınızda yazdıracağı çıktıdır. `<ad>` değişkeni aksanlı bir harf içermemelidir. Örneğin

```
\newtheorem{tanım}{Tanım}
```

komutunu sahanlığa verdiğinizde `tanım` ortamını LaTeX’e tanıtmış olur ve bu ortamı kullandığımızda şöyle bir çıktı alırsınız:

```
\begin{tanım}
  İşte yeni bir tanım.
\end{tanım}
```

**Tanım 2.1.** İşte yeni bir tanım.

Kullanılan teoreme (yukarıdaki örnekte Tanım) özel bir ad vermek ya da not düşmek isteyebilirsiniz. Bu, ortam komutundan sonra köşeli parantezler içinde belirtilebilir:

```
\begin{tanım}[Gauss]
  Gauss'un tanımı.
\end{tanım}
```

**Tanım 2.2** (Gaus). Gauss’un tanımı.

### 2.10.2 Sayaçlar

Sayaçlar, belge sınıfına göre varsayılan değerleri kullanır. Örneğin `book` sınıfında bir teorem kullanıldığında “Teorem 2.3” (kitabın 2’inci bölümünde yer alan 3’üncü teorem), `article` sınıfında bir teorem kullanıldığında “Teorem 3” (makaledeki 3’üncü teorem) benzeri çıktılar alınır. Varsayılan ayarları değiştirmek için sayacın takip etmesi istenilen bölüm seviyesi (`chapter`, `section` gibi) belirtilebilir:

```
\newtheorem{<ad>}{<başlık>}[<sayaç>]
```

Örneğin `article` sınıfında sahanlığa

```
\newtheorem{teorem}{Teorem}[section]
```

komutunu verdiğinizde teoreminiz `\section` başlık seviyesinin numarasına göre numara alır.

Varsayılan olarak, her teorem kendi sayacını kullanır. Bununla birlikte, benzer teoremlerin (örneğin Teoremler, Lemmalar ve Sonuçlar) bir sayacı paylaşması yaygındır. Bu durumda, sonraki teoremleri şöyle tanımlayın:

```
\newtheorem{<ad>}[<sayaç>]{<başlık>}
```

Burada `<sayaç>` kullanılacak olan sayacın adıdır. Genelde ana teorem adı olur. Örneğin sahanlıkta

```
\newtheorem{lemma}[teorem]{Lemma}
```

tanımlamasını yaparsanız (Bu komutu verebilmek için önceden `teorem` ortamı tanımlanmış olmalıdır) artık Lemma’lar Teorem’lerle aynı sayacı kullanacaktır.

`\newtheorem` komutu en fazla bir tane isteğe bağlı değişken içerebilir. Ayrıca komut `\newtheorem*` şekilde kullanılırsa numara verilmeyen bir teorem tanımlanmış olur.

### 2.10.3 Kanıtlar

Bir teoremin kanıtı `proof` ortamında yazılır. Genel kullanım şöyledir:

```
\begin{proof}
İşte benim kanıtım.
\end{proof}
```

Bu ortamı kullandığınızda en sona kanıtın bittiği anlamında bir kare (QED adıyla da bilinir) ekler ve Türkçe dil paketi ekli belgelerde “Kanıt” adını yazar. Bu ad, `\proofname` komutunda saklı olup istenirse değiştirilebilir:

```
\renewcommand{\proofname}{İspat}
```

## 2 Matematiksel İfadeler

Bu komuttan sonra kullanılan tüm proof ortamlarında artık “Kanit” yerine “İspat” yazar.

Ayrıca kanıtı tek seferliğine elle adlandırmak isterseniz, kendi adınızı köşeli parantezler içine yazabilirsiniz:

```
\begin{proof}[İspat]
  İşte diğer ispatım.
\end{proof}
```

Kanıtın sonunu bildiren  $\square$  işareti bazen son satırda yalnız kalırsa, `\qedhere` komutuyla onu doğru yere oturtabilirsiniz:

```
\begin{proof}
  Sadeleştirme yapılırsa  $[E=mc^2 \quad \text{\color{blue}\text{\texttt{\textbackslash qedhere}}}]$ 
\end{proof}
```

Özel bir QED sembolü kullanmak isterseniz `\qedsymbol` komutunu yeniden tanımlayabilirsiniz.

```
\renewcommand{\qedsymbol}{\text{\color{blue}\text{\texttt{\textbackslash blacksquare}}}}
\begin{proof}
  Siyah kare.
\end{proof}
```

Eğer sembolü gizlemek isterseniz `\renewcommand` komutunun son değişkenini boş bırakmanız yeterli olur.

### 2.10.4 Teorem Stilleri

Teorem stilleri `\theoremstyle` komutuyla değiştirilir. Bu komut, `\newtheorem` komutu kullanarak tanımlanan ortamların çıktısını değiştirme olanağını verir.

```
\theoremstyle{<stil adı>}
```

Buradaki `<stil adı>` kullanmak istediğiniz stildir. Bu komuttan sonra tanımlanmış tüm teoremler bu stili kullanacaktır. LaTeX’de önceden tanımlanmış stiller aşağıdaki-lerdir:

**Tablo 2.6:** Teorem Stilleri

Stil Adı	Açıklama	Görünüm
plain	Teoremler, lemmalar, önermeler vb. için kullanılır (varsayılan)	Başlık düz ve kalın, gövde metni vurgulu
definition	Tanımlar ve örnekler için kullanılır	Başlık düz ve kalın, gövde metni düz

Stil Adı	Açıklama	Görünüm
remark	Açıklamalar ve notlar için kullanılır	Başlık vurgulu, gövde metni düz

Örneğin sahanlığa

```
\theoremstyle{remark}
\newtheorem{notum}{Not}
```

komutlarını verip notum ortamını kullandığınızda başlık vurgulu, gövde metni düz olacaktır.

#### 2.10.4.1 Özel Stiller

Kendi stilinizi tanımlamak için `\newtheoremstyle` komutunu kullanabilirsiniz:

```
\newtheoremstyle{<stil adı>}% kullanılacak stilin adı
{<üst boşluk>}% teoremin üstünde bırakılacak boşluk. Örn:3pt.
{<alt boşluk>}% teoremin altında bırakılacak boşluk. Örn:3pt.
{<gövde yazı tipi>}% teorem gövdesinde kullanılacak yazı tipi.
%Örn:\normalfont, \itshape...
{<girinti>}% Paragraf girintisi ölçüsü. Örn:0pt
{<başlık yazı tipi>}% teorem başlık yazı tipi.
%Örn:\sffamily,\bfseries
{<noktalama>}% başlıktan sonraki noktalama.
%Noktalama istenmezse boşluk bırakılabilir. Örn:\;
{<boşluk>}% başlıktan sonraki boşluk. Örn:0.25em
{<manuel başlık>}% Elle başlık belirtilir.
```

Boş bırakılan herhangi bir değişken olursa varsayılan değerler alınır. Son satırdaki `<manuel başlık>` değişkeni `\thmname`, `\thmnumber` ve `\thmnote` komutlarıyla biçimlendirilir. Birinci komut başlığı, ikinci komut numarayı, üçüncüsü ise notu biçimlendirmek içindir.

Not değişkeni her zaman isteğe bağlıdır, ancak başlık oluşturulurken `\thmnote` komutuyla belirtilmezse varsayılan olarak görünmeyecektir.

```
\newtheoremstyle{benimstilim}
{5pt}%
{5pt}%
{\normalfont}%
{} %
{\bfseries\sffamily}%
{\;}%
{0.25em}%
```

## 2 Matematiksel İfadeler

```
{\thmname{#1} \thmnumber{#2}\thmnote{---#3}.}%  
\theoremstyle{benimstilim}%  
\newtheorem{sonuc}{Sonuç}%
```



## 3 Özel Sayfalar

Bu bölümde bir kitaptaki özel sayfalar olan kaynakça ve dizinden bahsedeceğiz.

### 3.1 Kaynakça

#### 3.1.1 Bütünleşik Kaynakça

LaTeX’de kaynakça oluşturmanın bir yolu, kaynakçayı kaynak dosyanızın (.tex uzantılı ana dosyanız) içindeki bir ortamda hazırlamaktır. Kullanacağımız ortam thebibliography ortamıdır.

```
\begin{thebibliography}{<sayı>}
```

```
\end{thebibliography}
```

Ortam komutundaki <sayı> değişkeni, kaynağın etiketi veya etiket girilmediği takdirde verilen sıra numarasının kaç karakter uzunluğunda olacağını belirtir. Örneğin ortam

```
\begin{thebibliography}{9}
```

```
\end{thebibliography}
```

şeklinde oluşturulursa etiket veya etiket girilmediği takdirde verilen sıra numarası için bir karakter uzunluğunda yer ayrılması gerektiği ve toplamda bu ortama en fazla dokuz adet kaynak girileceği belirtilmiş olur. Eğer dokuzdan fazla kaynak girilecekse, 99 kaynağa kadar izin veren “45” gibi iki basamaklı bir sayı girilebilir.

Ortama her kaynak \bibitem komutuyla eklenir ve komuttan sonra kaynağı tanımlayıcı bilgiler girilir. Bu bilgiler girilirken biçim elle oluşturulur.

```
\bibitem[<etiket>]{<anahtar>}
```

Komutun zorunlu değişkeni olan <anahtar>, ileride kaynağa atıf yapmak için kullanacağınız bir tanımlayıcıdır ve her kaynak için benzersiz olmalıdır. Genelde akılda kolay kalması için yazarın soyadı ve yayın yılı olacak şekilde düzenlenir.

Zorunlu olmayan <etiket> değişkeni girilmediği takdirde kaynağın önüne köşeli parantezler içinde kaynağın sıra numarası yazdırılır.

### 3 Özel Sayfalar

Kaynağın sıra numarasının köşeli parantezler içinde yazılması istenmezse aşağıdaki komutlarla değişiklik yapılabilir.

```
\makeatletter
\renewcommand{\@biblabel}[1]{\textbf{#1.}}
\makeatother
```

Bu komut verilirse sıra numaraları parantezsiz, kalın ve ardında nokta olacak şekilde yazılır.

Ortam genelde `\end{document}` komutundan hemen önce oluşturulur ve ortamın oluşturulduğu yerde LaTeX, book ve report sınıflarında eğer Türkçe dil paketi eklenmişse “Kaynakça”, article sınıfında ise “Kaynaklar” ismini ve ardından kaynakları yazdırır.

Kaynaklardan herhangi birine atıf `\cite` komutuyla yapılır.

```
\cite[<seçenekler>]{<anahtar>}
```

Komutun zorunlu değişkeni olan `<anahtar>`, atıf yapılmak istenen kaynağın `\bibitem` komutundaki zorunlu değişkenidir. İsteğe bağlı `<seçenekler>` değişkeninde ise sayfa numarası, bölüm numarası gibi fazladan vurgulanmak istenen bilgiler girilebilir.

Atıf yapılan yerde kaynağın etiketi ya da etiket girilmediği takdirde sıra numarası köşeli parantez içinde yazdırılır. Eğer fazladan yapılan vurgu varsa, bu, etiket ya da sıra numarasının devamında virgülden sonra yazdırılır.

Aynı yerde birden fazla kaynağa atıf yapılacaksa atıf yapılacak kaynakların anahtarları aralarına virgül koyularak yazılır:

```
\cite{<anahtar1>,<anahtar2>,<anahtar3>}
```

Aşağıda kaynak dosya örneği verilmiştir.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{url}
\title{\LaTeX'de Kaynakça Yönetimi 1: Bütünleşik Kaynakça}
\author{Zafer Acar}
```

```
\begin{document}
\maketitle
WYSIWYG editörleri yerine, \TeX/\LaTeX{} \cite{lampport94}
dizgi sistemini kullanmaya başlayın. Görüldüğü gibi kaynakça oluşturmak
ve atıf yapmak oldukça kolaydır.
```

Ali Nesin, `\cite[s.~47]{nesin07}`'de pokerin matematiğini anlatıyor.

İki kaynağa birden atıf `\cite{lamport94,nesin07}` şeklinde yapılır.

```
\begin{thebibliography}{9}
\bibitem{lamport94}
Leslie Lamport,
\textit{\LaTeX: a document preparation system}, Addison Wesley,
Massachusetts, 2nd edition, 1994.

\bibitem[N]{nesin07} Ali Nesin, \textit{Matematik ve Oyun},
Nesin Yayıncılık, 2007.
\end{thebibliography}
\end{document}
```

#### 3.1.2 Kaynakçanın ayrı dosyada hazırlanması

Kaynakçayı ayrı bir dosyada hazırlayıp TeX dağıtımlarıyla hazır olarak gelen BiBTeX programıyla yazdırabiliriz.

Bu yöntemde kaynakça, uzantısı `.bib` olan ayrı bir dosyada hazırlanır. Bu dosya basit bir metin dosyası olup, metin editörü ya da LaTeX editörü kullanılarak oluşturulabilir, düzenlenebilir. Ayrıca Mendeley<sup>1</sup> ya da Jabref<sup>2</sup> gibi akademik referans düzenleme programlarından da yararlanılabilir.

Bu yöntemin önemli avantajları vardır:

1. Biçimlendirme otomatik yapılır. Eğer çalışmanızı yayımlayacak dergi ya da yayınevi kaynakçayı farklı bir formatta isterse her kaynağı tek tek elle biçimlendirmek zorunda kalmazsınız. Basit bir komut işinizi görür.
2. Dosyayı bir kere oluşturur ve sonra başka çalışmalarda kullanabilirsiniz.
3. Google Akademik<sup>3</sup>, Google Kitaplar<sup>4</sup> ve DergiPark<sup>5</sup> gibi platformlardan kullandığınız kaynakların BiBTeX kodunu çekebilirsiniz (bkz. Şekil 3.1).
4. Yukarıda da bahsettiğimiz gibi Mendeley ve Jabref gibi akademik atıf düzenleme programlarını kullanarak kaynakların BiBTeX kodunu oluşturabilir, düzenleyebilirsiniz.

##### 3.1.2.1 Dosyanın hazırlanması

Aşağıda `.bib` uzantılı bir dosya örneği gösterilmiştir.

---

<sup>1</sup>[https://www.mendeley.com/?interaction\\_required=true](https://www.mendeley.com/?interaction_required=true)

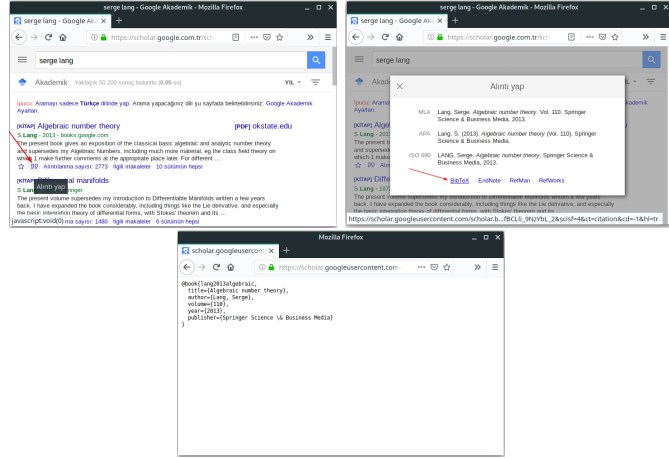
<sup>2</sup><https://www.jabref.org/>

<sup>3</sup><https://scholar.google.com.tr/>

<sup>4</sup><https://books.google.com.tr/>

<sup>5</sup><http://www.dergipark.org.tr/tr/>

### 3 Özel Sayfalar



Şekil 3.1: Google Akademikten alıntı yapma

```
@book{lang13,
  title={Algebraic number theory},
  author={Lang, Serge},
  volume={110},
  year={2013},
  publisher={Springer Science & Business Media},
}

@article{lampert78,
  title={Time, clocks, and the ordering of events in a
    distributed system},
  author={Lampert, Leslie},
  journal={Communications of the ACM},
  volume={21},
  number={7},
  pages={558--565},
  year={1978},
  publisher={ACM},
}

@manual{oetiker06,
  author = {Oetiker, Tobias and Partl, Hubert and Hyna, Irene
    and Schlegl, Elisabeth},
  title = {İnce bir {\LaTeXe} Elkitabı veya, 116 dakikada
    {\LaTeXe}},
  note = {Türkçesi: Bekir Karaoğlu},
  url = {http://ftp.ntua.gr/mirror/ctan/info/lshort/turkish/
    lshort-tr.pdf},
  year = {2006},
}
```

```
}
```

Bu dosyada Serge Lang’a ait bir kitap (`@book`), Leslie Lamport’a ait bir makale (`@article`) ve LaTeX için bir teknik kılavuz (`@manual`) vardır.

Her kaynağın ilk olarak `@` işaretiyle türü belirtilir. Yukarıdakilere ek olarak rapor için `@report`, tez için `@thesis`, çevrimiçi kaynaklar için `@online` kullanılır. Bunların dışındaki birçok türe LaTeX editörlerinin menü çubuğuklarında bulunan “Kaynakça (Bibliography)” menüsünden ulaşılabilir.

İlk girdi (`lang13`, `lamport78`, `oetiker06`) kaynağa atıf yapmak için kullanılan anahtardır. Sonrasında gelenler de tahmin edilebileceği gibi başlık (`title`), yazar (`author`), yayıncı (`publisher`), yıl (`year`), dergi (`journal`), cilt (`volume`)... gibi kaynağı tanımlayan bilgilerdir. Bu tanımlamaların her biri eşittir işaretiyle sonra iki çengelli parantez arasında yapılır (çift tırnak da kullanılabilir) ve her tanımlama (sonuncusu olsa dahi) virgülle ayrılır.

Yazar adı ya

```
author={Adı Soyadı}
```

ya da

```
author={Soyadı, Adı}
```

şeklinde girilmelidir ve birden fazla yazar varsa yazarlar yukarıdaki yazımdan dolayı virgülle değil `and` ile ayrılmalıdır. Yazarları ayırmak için virgül kullanırsanız yüksek ihtimalle LaTeX, yazarların adları ve soyadlarını karıştıracaktır.

Bir diğer önemli nokta özel kelimeleri yazmak için kullanılan komutları ve aksanlı harfleri iki çengelli parantez içinde yazmaktır. Örneğin “â” için `{\^a}` yazılmalıdır. Genel olarak sorun yaşanan karakterleri iki çengelli parantez içine yazmak gerekir.

Her tür için zorunlu olarak belirtilmesi gereken bilgiler ve isteğe bağlı bilgiler vardır. Bunların ne olduklarını tahmin etmek zor değildir. Bu konuda editörden de yararlanabilirsiniz. Örneğin, `.bib` uzantılı dosyayı açıp editörde “Kaynakça → Tez” yolunu izlerseniz aşağıdaki listeyi yazdıracaktır.

```
@thesis{ID,
  author = {author},
  title = {title},
  type = {type},
  institution = {institution},
  date = {date},
  OPTsubtitle = {subtitle},
  OPTtitleaddon = {titleaddon},
  OPTlanguage = {language},
  OPTnote = {note},
  OPTlocation = {location},
```

### 3 Özel Sayfalar

```
OPTmonth = {month},
OPTisbn = {isbn},
OPTchapter = {chapter},
OPTpages = {pages},
OPTpagetotal = {pagetotal},
OPTaddendum = {addendum},
OPTpubstate = {pubstate},
OPTdoi = {doi},
OPTeprint = {eprint},
OPTeprintclass = {eprintclass},
OPTeprinttype = {eprinttype},
OPTurl = {url},
OPTurldate = {urldate},
}
```

Görüldüğü gibi ilk altı satır zorunlu, OPT ile başlayanlar isteğe bağlıdır. İsteğe bağlı olanlardan belirtmek istediklerinizin başında bulunan OPT'yi silip tanımlamayı yapabilirsiniz.

Editörden yararlanmanın diğer bir yolu “Kaynakça → Kaynakça Kaydı Ekle...” yolunu izlemektir. Bu yolu izlediğinizde aşağıdaki pencere açılır (örnek TeXstudio editörüne aittir).

ID	Alan	Değer
1	ID	
2	author	
3	title	
4	date	
5	editor	
6	editora	
7	editorb	
8	editorc	
9	translator	
10	annotator	
11	commentator	
12	introduction	
13	foreword	
14	afterword	
15	subtitle	
16	titleaddon	
17	maintitle	
18	mainsubtitle	
19	maintitleaddon	

Pencerenin solunda kaydı eklemek istediğiniz dosyayı ve ortada kayıt türünü belirtir, sağda da kaynağın bilgilerini girersiniz. Zorunlu bilgiler en üstte yer alan kalın yazılmış

olanlardır.

### 3.1.2.2 Kaynakçanın yazdırılması

Kaynakçayı yazdırmak için BiBTeX'i kullanacağız. BiBTeX'in LaTeX'le standart olarak geldiğini ifade etmiştik. Dolayısıyla bu programı kullanmak için ek bir şey yapmanız gerekmez.

Oluşturulan .bib uzantılı dosya \bibliography komutuyla içeri aktarılır, \bibliographystyle komutuyla da kullanılacak biçim belirtilir.

```
\bibliographystyle{<biçim>}
\bibliography{<dosya>}
```

Burada yer alan <dosya> uzantısının belirtilmesine gerek yoktur. Dosyanın kaynakca.bib olduğunu varsayarak, komut \bibliography{kaynakca} şeklinde verilir. Kullanılabilecek biçimler abbrv, acm, alpha, apalike, ieetr, plain, siam ve unsrt'dir. Biçimlerin nasıl çıktılar verdiklerini görmek için şuraya<sup>6</sup> bakabilirsiniz.

Atıf, bütünleşik kaynakçada olduğu gibi \cite komutuyla yapılır fakat bütünleşik kaynakçadan farklı olarak atıf yapılmayan kaynaklar yazdırılmaz. Bazı kaynakların bu kuraldan ayrı tutulması istenirse \nocite komutu, değişkenine kaynağın anahtarı yazılarak \bibliography komutundan önce verilmelidir.

```
\nocite{<anahtar>}
```

Eğer tüm kaynakların bu kuraldan ayrı tutulması isteniyorsa komut \nocite{\*} şeklinde verilmelidir.

Kaynakçanın belgeye yazılması için kaynak dosyanın derlenip, BiBTeX programının çalıştırılması ve ardından dosyanın en az iki kere daha derlenmesi gerekir. BiBTeX programı, editörde "Araçlar → Kaynakça" yoluyla çalıştırılır (klavye kısa yolu F8). Aynı şey, uçbirimde sırasıyla

```
pdflatex kaynakdosya
bibtex kaynakdosya
pdflatex kaynakdosya
pdflatex kaynakdosya
```

komutları çalıştırılarak yapılabilir.

Aşağıda kaynak dosya örneği verilmiştir. Bu dosyayı derleyebilmeniz için içeriği yukarıda verilen kaynakca.bib dosyasının bu dosyayla aynı dizinde olması gerektiğini unutmayınız.

---

<sup>6</sup>[https://tr.overleaf.com/learn/latex/Bibtex\\_bibliography\\_styles](https://tr.overleaf.com/learn/latex/Bibtex_bibliography_styles)

### 3 Özel Sayfalar

```
\documentclass[10pt,a4paper]{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{dtk-logos} % \BibTeX komutu için...
\title{Kaynakça Yönetimi 2: \BibTeX}
\author{Zafer Acar}
\begin{document}
\maketitle
Lang'ın kitabı \cite{lang13}, Lamport'un makalesi \cite{lamport78}
ve \LaTeX{} için Türkçe kaynak \cite{Oetiker06} \dots

\bibliographystyle{siam}
\bibliography{kaynakca}
\end{document}
```

## 3.2 Dizin

Bilimsel bir yapıtta bulunması gereken *dizin* ya da diğer adıyla *indeks*, bir yapıtın kişi, konu, yer adı vb. bakımından içindekileri yer numarasıyla belirten ve yapıtın arkasında yer alan alfabetik listedir.

LaTeX'de dizin oluşturabilmek için sahanlığa

```
\usepackage{makeidx}
\makeindex
```

komutları girilir. Birinci komut dizin için gerekli olan **makeidx**<sup>7</sup> paketini çağırır, ikinci komut ise dizinleme komutlarını etkinleştirir.

Dizinde gösterilmek istenen madde, `\index` komutunun değişkeni olarak girilir:

```
\index{<madde>}
```

Dizin maddesi girme örnekleri aşağıda gösterilmiştir.

---

<sup>7</sup><http://ftp.ntua.gr/mirror/ctan/macros/latex/base/makeindx.pdf>



Örnek	Dizin Maddesi	Açıklama
<code>\index{kuvvet}</code>	kuvvet, 7	Düz madde
<code>\index{magnetik!kuvvet}</code>	magnetik, 10	‘kuvvet’in alt maddesi
<code>\index{magnetik see{kuvvet}}</code>	magnetik, <i>bkz.</i> kuvvet	Atıflı madde
<code>\index{Türev@textsl{Türev}}</code>	<i>Türev</i> , 2	Biçimlendirilmiş madde
<code>\index{Limit@textbf{Limit}}</code>	<b>Limit</b> , 7	Biçimlendirilmiş madde
<code>\index{Açı textbf}</code>	Açı, <b>3</b>	Biçimlendirilmiş sayfa numarası
<code>\index{İvme textit}</code>	İvme, 5	Biçimlendirilmiş sayfa numarası
<code>\index{ecole@'ecole}</code>	école, 4	Aksanlı madde girişi

LaTeX, kaynak dosyanızı derlediğinizde bu dizin maddelerini sayfa numaralarıyla birlikte, kaynak dosyayla adı aynı fakat uzantısı `.idx` olan bir dosyaya kaydeder (bu dosyaya *ham dosya* denir). Bu dosyanın `makeindex` programından geçirilmesi gerekir. Bu editörde “Araçlar → Dizin” yoluyla yapılır. Aynı şey uçbirimde,

```
makeindex kaynakdosya
```

komutu girilerek yapılabilir. Dosya tekrar derlendiğinde sıralanmış dizin belgeye yazılır. Bunun için dizinin yazılması istenen yere `\printindex` komutu verilir. Bu genelde, belgenin en sonunda `\end{document}` komutundan hemen öncedir. Komutun girildiği yere LaTeX, Türkçe dil paketi ekli belgelerde “Dizin” başlığını oluşturur ve belgede `\index` komutuyla eklenmiş maddeleri sırayla dizer.

Program, ham dosyayı işleyip dizin maddelerini abece sırasına göre dizer ve `.ind` uzantılı bir dosyaya aktarır. Ancak, Türkçe aksanlı harflerle başlayan kelimeler doğru yerde yazılmazlar. Bu harflerin doğru yere yazılması için `.ind` uzantılı dosyanın metin editörüyle açılarak elle düzenlenmesi gerekir. Ardından kaynak dosya derlenir. Elle düzeltilmeden sonra tekrar `makeindex` programı çalıştırılırsa `.ind` uzantılı dosya tekrar oluşturulacağı için elle yapılan düzeltmeler bozulur. O yüzden düzeltme en son yapılmalıdır.

Aksanlı harflerle başlayan kelimelerin doğru yerde yazılmaları için aksanlı madde girme komutundan faydalanılabilir. Örneğin “çekik” kelimesinin peşine `\index{czekiç@çekiç}` komutu verilirse bu kelime doğru yerde dizilecektir. Burada yapılan sorun yaratan “ç” harfi yerine “cz” yazılmasıdır.

### 3.2.1 Çoklu Dizin

Birden fazla dizin oluşturmak isterseniz (örneğin biri *normal dizin* diğeri de *simgeler dizini*) **index**<sup>8</sup> paketini kullanabilirsiniz. Her dizin paket eklendikten ve `\makeindex` komutu sahanlıkta verildikten sonra `\newindex` komutuyla tanıtılır.

```
\usepackage{index}
\makeindex
\newindex{normal}{ndx}{nnd}{Normal Dizin}
\newindex{simge}{sdx}{snd}{Simgeler Dizini}
```

Komutun dört değişkeni vardır. Bunlar sırasıyla, dizin adı (örnekte `normal` ve `simge`), oluşturulacak ham dosyanın uzantısı (örnekte `.ndx` ve `.sdx`), `makeindex` tarafından ham dosyanın işlenmesiyle oluşturulan dosyanın uzantısı (örnekte `.nnd` ve `.snd`) ve son olarak `\printindex` komutuyla yazdırılacak başlıktır (örnekte “Normal Dizin” ve “Simgeler Dizini”). Buradaki uzantılar varsayılan `.idx` ve `.ind` uzantılardan farklı olmalıdır.

Ardından bir kelime ya da simgeyi dizine eklemek için, eklemek istenilen dizine göre `\index` komutu köşeli parantezler içinde dizin adı belirtilerek kullanılır.

```
\index[normal]{kuvvet}
\index[simge]{F@$\vec{F}$}
```

Birinci komut, “kuvvet” kelimesini normal dizine, ikinci komut  $\vec{F}$  simgesini simgeler dizinine ekler.

Belge derlendikten sonra iki tane `\makeindex` komutu uçbirimde,

```
makeindex kaynakdosya.ndx -o kaynakdosya.nnd
makeindex kaynakdosya.sdx -o kaynakdosya.snd
```

şeklinde verilir. Belgenizde dizinlerin yazılması istenen yere de

```
\printindex[normal]
\printindex[simge]
```

komutları verilir. Ardından belge tekrar derlenerek dizinler yazdırılır.

Çoklu dizin için diğer bir seçenek **multind**<sup>9</sup> paketini kullanmaktır. Görece `index` paketine göre daha pratiktir. Sahanlığa

```
\usepackage{multind}
\makeindex{normal}
\makeindex{simge}
```

<sup>8</sup><http://ftp.cc.uoc.gr/mirrors/CTAN/macros/latex/contrib/index/index.pdf>

<sup>9</sup><https://www.ctan.org/pkg/multind>

komutlarıyla `normal` ve `simge` adında iki dizin tanımlanır. Yine dizine yazılması istenen maddeler `\index` komutundan önce çengelli parantezler içinde dizin adı belirtilerek girilir.

```
\index{normal}{kuvvet}
\index{simge}{F@$\vec{F}$}
```

Bu defa `makeindex` programı uçbirimde

```
makeindex normal
makeindex simge
```

komutlarıyla çalıştırılır. Yine `\printindex` komutları dizinlerin eklenmesi istenen yere

```
\printindex{normal}{Normal Dizin}
\printindex{simge}{Simgeler Dizini}
```

şeklinde verilir.

### 3.2.2 Dizinin İçindekiler tablosuna yazılması

Dizini İçindekiler tablosuna yazmak için `\printindex` komutunun peşine `book` ve `report` sınıflarında `\addcontentsline` komutu,

```
\addcontentsline{toc}{chapter}{\indexname}
```

şeklinde, `article` sınıfında ise

```
\addcontentsline{toc}{section}{\indexname}
```

şeklinde verilmelidir. Çoklu dizin oluşturulmuşsa, `book` ve `report` sınıflarında

```
\printindex{normal}{Normal Dizin}
\addcontentsline{toc}{chapter}{Normal Dizin}
\printindex{simge}{Simgeler Dizini}
\addcontentsline{toc}{chapter}{Simgeler Dizini}
```

şeklinde, `article` sınıfında ise komutlardaki `chapter` yazan yere `section` yazarak verilmelidir.



## Kaynakça

Oetiker, T., Partl, H., Hyna, I., and Schlegl, E. (2006). *İnce bir LaTeX2e Elkitabı veya, 116 dakikada LaTeX2e*. <http://ftp.ntua.gr/mirror/ctan/info/lshort/turkish/lshort-tr.pdf>. Türkçesi: Bekir Karaoğlu.