

# Web Programlama


## 5. Hafta

21.10.2024



Mühendislik  
Fakültesi  
Bilgisayar Mühendisliği

Hazırlayan: Dr. Ercan Ezin

A large black and white QR code is positioned on the left side of the slide. It consists of a square grid of black modules on a white background, with three prominent square markers at the corners for alignment.

Yoklama için  
Qr okutun veya  
asagida yazılı  
linke gidin.

<https://128.pl/KQP6j>

# PRACTICE ASSIGNMENT

Kimler yaptı?

Sayfa Kaynağını görüntüle?  
Kim yaptı?

## Ercan Ezin

[Hakkımda](#)[Projeler](#)[İletişim](#)

### Hakkımda

Merhaba! Ben Ercan Ezin, bir yazılım geliştiricisiyim.

### Projeler

- Proje 1: Detaylar
- Proje 2: Detaylar
- Proje 3: Detaylar

### İletişim

Bana ulaşmak için: [ercanezin@example.com](mailto:ercanezin@example.com)

# GİRİŞ

GitHub, Kaydırma, Konumlandırma, Düzen Tasarımı, Responsive Design,

Not: Bu dersin içeriği kaynakçada belirtilen materyallerde derlenerek üretilmiştir.

# GIT VS GITHUB

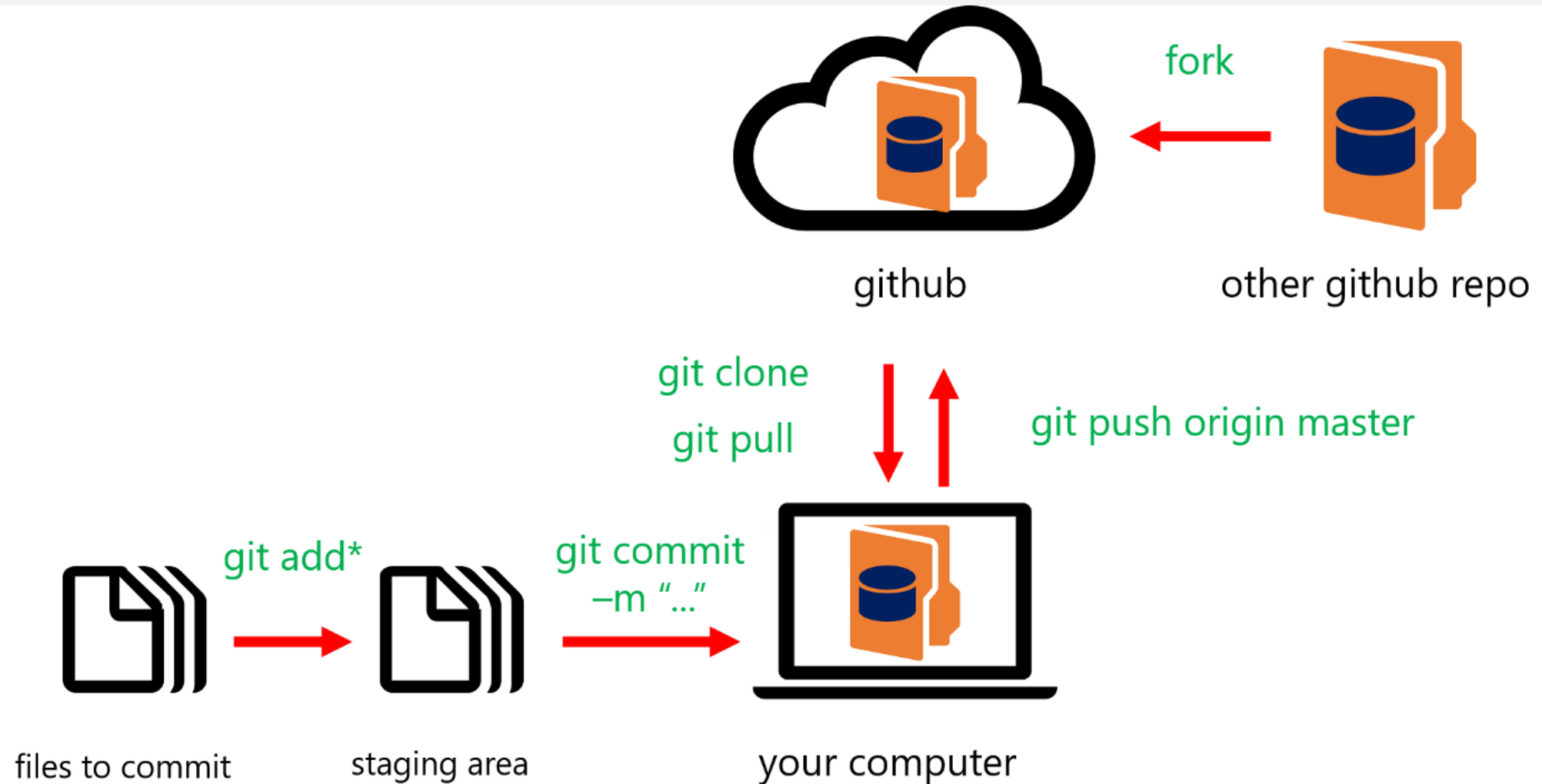


- Git bir sürüm kontrol sistemidir. Kaynak kodunuzu yönetmeye ve geçmişe dönük değişiklikleri zamana bağlı düzenli olarak kayıt altına alır.



- GitHub ise bulut-tabanlı olup Git sistemiyle sürümü kontrol edilen projelerin yer aldığı bir hosting firmasıdır.

# HOW GIT WORKS



# OPENING A GITHUB ACCOUNT

# OPENING A GITHUB ACCOUNT-1

## **GitHub'a Kaydolma Adımları**

- **GitHub Web Sitesini Ziyaret Edin:** Web tarayıcınızdan Adres çubuğuna <https://github.com> yazın ve Enter tuşuna basın.
- **Sign Up'a Tıklayın:** GitHub ana sayfasında, sağ üst köşede bulunan "Sign up" butonuna tıklayın.
- **E-posta Adresinizi Girin:** Açılan sayfada, üniversiteden almış olduğunuz edu.tr uzantılı e-posta adresinizi yazın ve "Continue" butonuna tıklayın.
- **Kullanıcı Adınızı Seçin:** Github kullanıcı adı uzun yıllar kullanacağınız ve projelerinizde diğer kullanıcıların göreceği bir metindir. O yüzden sizi iyi temsil eden benzersiz bir kullanıcı adı seçiniz ve "Continue" butonuna tıklayın.
- **Şifre Belirleyin:** Güçlü bir şifre oluşturun ve "Continue" butonuna tıklayın.
- **Profil Bilgilerini Doldurun:** İsteğe bağlı olarak, adınızı ve diğer bilgileri girin. "Continue" butonuna tıklayın.



# OPENING A GITHUB ACCOUNT-2

- **E-posta Doğrulaması:** GitHub, verdiğiniz e-posta adresine bir doğrulama e-postası gönderir. E-posta kutunuzu kontrol edin ve gelen e-postadaki "Verify email address" butonuna tıklayarak e-posta adresinizi doğrulayın.
- **Hesap Ayarları:** Doğrulama işleminden sonra, GitHub size bazı sorular soracaktır (örneğin, GitHub'ı nasıl kullanacağınız hakkında). Bu adımları tamamlayın ve "Continue" butonuna tıklayın.
- **Plan Seçimi:** GitHub, ücretsiz veya ücretli plan seçenekleri sunar. Öğrencilere edu.tr uzantılı email ile premium hesap ücretsiz sunulmaktadır. Şimdilik ücretsiz planı seçin ve "Continue" butonuna tıklayın.
- **Hesabınız Hazır:** Kaydolma işleminiz tamamlandı! Artık GitHub hesabınıza giriş yapabilir ve projelerinizi yönetmeye başlayabilirsiniz.

Videolu anlatım:

<https://www.youtube.com/watch?v=-2LZ96VST9c>

Öğrenci hesabı için:

<https://education.github.com/pack/join>



# BASIC GIT COMMANDS

# BASIC GIT COMMANDS-1

## **Temel Git Komutları. [3]**

Git kullanmak için geliştiriciler, kodu kopyalamak, oluşturmak, değiştirmek ve birleştirmek için belirli komutlar kullanır. Bu komutlar doğrudan komut satırından veya GitHub Desktop ya da Git Kraken gibi bir uygulama kullanılarak çalıştırılabilir. İşte Git kullanımı için bazı yaygın komutlar:

**git init:** Yepyeni bir Git deposunu başlatır ve mevcut bir dizini izlemeye başlar. Mevcut dizin içinde, sürüm kontrolü için gereken iç veri yapısını barındıran gizli bir alt klasör ekler.

**git clone:** Uzakta zaten var olan bir projenin yerel kopyasını oluşturur. Klon, projenin tüm dosyalarını, geçmişini ve dallarını içerir.

# BASIC GIT COMMANDS-2

**git add:** Bir değişikliği sahneye(staging area) alır. Git, bir geliştiricinin kod tabanındaki değişiklikleri izler, ancak değişikliklerin proje tarihine dahil edilmesi için bunları sahneye almak ve anlık yedeğini(snapshot) almak gerekir. Bu komut, o iki adımlı sürecin ilk kısmı olan sahneleme işlemini gerçekleştirir. Sahneye alınan herhangi bir değişiklik, bir sonraki anlık yedeğinin bir parçası olacak ve projenin tarihine dahil edilecektir. Ayrı ayrı staging ve commit işlemleri, geliştiricilere projelerinin tarihini kontrol etme imkanı sunar ve kodlama ile çalışma şekillerini değiştirmez.

**git commit:**Anlık yedeği proje tarihine kaydeder ve değişiklik izleme sürecini tamamlar. Kısacası, bir commit yapmak fotoğraf çekmeye benzer. git add ile sahneye alınan her şey, git commit ile anlık görüntünün bir parçası haline gelir.

# BASIC GIT COMMANDS-3

**git status:** Değişikliklerin durumunu izlenmemiş, değiştirilmiş veya sahneye alınmış olarak gösterir.

**git branch:** Yerel olarak üzerinde çalışılan dalları gösterir.

**git merge:** Geliştirme hatlarını birleştirir. Bu komut genellikle iki farklı dalda yapılan değişiklikleri birleştirmek için kullanılır. Örneğin, bir geliştirici, bir özellik dalındaki değişiklikleri ana dal ile birleştirmek istediğinde birleştirme işlemi yapar.

**git pull:** Yerel geliştirme dosyalarını, uzak kardeşinden gelen güncellemelerle günceller. Geliştiriciler, bir takım arkadaşı uzak bir depoda commitler yaptıysa ve bu değişiklikleri yerel ortamlarında yansıtmak istiyorlarsa bu komutu kullanır.

**git push:** Yerel olarak bir depoya yapılan tüm commitleri uzak depoya günceller.

# UPLOADING YOUR PORTFOLIO-1

## **Yeni Repository Oluşturma**

- GitHub'a giriş yaptıktan sonra, sağ üst köşedeki "+" butonuna tıklayın ve "New Repository" seçin.
- Repository adınızı şu şekilde ayarlayın:  
github-kullanıcı-adi.github.io
- Repository'yi "Public" yani herkese açık olarak ayarlayın ve "Add a README file" kutusunu işaretleyin.
- "Create repository" butonuna tıklayın.

# UPLOADING YOUR PORTFOLIO-2

## **README.md Dosyasını Özelleştirme**

- Repository'nize gidin ve README.md dosyasına tıklayın.
- Dosyayı düzenlemek için kalem ikonuna tıklayın.
- Dosya içeriğini Markdown formatına göre düzenleyin.

Türkçe Markdown Rehberi:

<https://medium.com/deep-learning-turkiye/t%C3%BCrk%C3%A7e-markdown-rehberi-61779d2e2a96>

# UPLOADING YOUR PORTFOLIO-3

## Örnek README.md dosya içeriği:

```
# Simba The Cat  
Kişisel portföyüme hoş geldiniz! Ben bir yazılım geliştiricisiyim.
```

```
## Hakkımda  
Cloud computing ve teknoloji topluluklarında deneyimim var.
```

```
## Yetenekler  
- Cloud Computing  
- GitHub & Git  
- Microsoft Power Platform
```

```
## İletişim  
[LinkedIn](https://www.linkedin.com/username)  
veya [GitHub](https://github.com/username)  
üzerinden bana ulaşabilirsiniz.
```

## Simba The Cat

Kişisel portföyüme hoş geldiniz! Ben bir yazılım geliştiricisiyim.

### Hakkımda

Cloud computing ve teknoloji topluluklarında deneyimim var.

### Yetenekler

- Cloud Computing
- GitHub & Git
- Microsoft Power Platform

### İletişim

[LinkedIn](#) veya [GitHub](#) üzerinden bana ulaşabilirsiniz.



# UPLOADING YOUR PORTFOLIO-4

## GitHub Pages'i Etkinleştirme

- Repository'nizde, üstteki "Settings" sekmesine tıklayın.
  - GitHub Pages bölümüne kadar aşağı kaydırın.
  - "Source" altında main (veya master) seçeneğini seçin ve "Save" butonuna tıklayın.
- Sonuç:** <https://github-kullanici-adiniz.github.io/> formatında canlı bir URL alacaksınız. Bu url ile yüklediğiniz html dosyalarını canlı olarak görebilirsiniz.

Videolu anlatım: <https://www.youtube.com/watch?v=0JxhGm8NKYE>

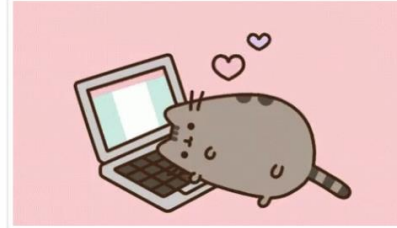
# HTML & CSS RECALL

# HTML RECALL

Image: **img**

```

```



Link: **a** (note: not **link**)

```
<a href="google.com">click here!</a>
```

[click here!](https://www.google.com)

Strong (bold): **strong** (note: don't use **b**)

```
<strong>Be BOLD</strong>
```

**Be BOLD**

Emphasis (italic): **em** (note: don't use **i**)

```
He's my <em>brother</em> and all
```

He's my *brother* and all

# CSS-RECALL

- Bir web sayfasının görünümünü ve düzenini tanımlar.
- Stil setlerini tanımlayan CSS kurallarından oluşur.



# HTML TYPES

Her HTML elementi, HTML spesifikasyonuna göre üç ana kategoriye ayrılır:

- **blok(block)**: büyük içerik blokları, yüksekliği ve genişliği vardır. `<p>`, `<h1>`, `<blockquote>`, `<ol>`, `<ul>`, `<table>`
- **satır içi(inline)**: az miktarda içerik, yüksekliği veya genişliği yoktur. `<a>`, `<em>`, `<strong>`, `<br>`
- **satır içi blok(inline block)**: yüksekliği ve genişliği olan satır içi içerik. `<img>`
- **meta veri**: sayfa hakkında bilgi, genellikle görünmez. `<title>`, `<meta>`

# BLOCK ELEMENTS

Örnekler:

`<p>`, `<h1>`, `<blockquote>`, `<ol>`, `<ul>`, `<table>`

- Sayfanın tam genişliğini kaplar (üstten alta akar) .
- Yükseklik ve genişliğe sahip olabilir. (width-height)
- Blok veya satır içi elementleri içiçe(nested) olarak barındırabilir.



# BLOCK EXAMPLE

```
<h1>About Simba</h1>  
<p>  
  He likes <em>wet food</em>  
</p>
```



Tarayıcıda görüntülenen

## About Simba

He likes *wet food*

# BLOCK EXAMPLE

```
h1 {  
  border: 5px solid red;  
}
```

h1 block elemanına  
5 piksellik border eklenirse



Tarayıcıda görüntülenen

**About Simba**

He likes *wet food*



# BLOCK EXAMPLE

```
h1 {  
  border: 5px solid red;  
  width: 50%;  
}
```

Block elemanları için width property değeri 50% atandığında boyutu içinde bulunduğu elemanın yarısına düşer.

**About Simba**

He likes *wet food*

# INLINE ELEMENTS

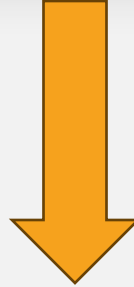
Örnekler: `<a>`, `<em>`, `<strong>`, `<br>`

- Satırıçi- Inline elemanlar yalnızca gerektiği kadar genişlik kaplar (soldan sağa akar).
- Yükseklik ve genişliğe sahip olamaz.
- İçiçe blok eleman bulunduramaz .
- Pozisyonlanamaz (yani, CSS özellikleri olan float ve position satırı içi elemanlara uygulanmaz).
- Bunun yerine dışındaki blok elemanını pozisyonlandırmalısınız.



# INLINE EXAMPLE

```
<strong>Web programlama Öğreniyorum:</strong>  
<a href="https://www.harran.edu.tr">HRU Web Sitesi</a>  
<a href="https://www.tubitak.gov.tr">TUBİTAK</a>  
<a href="https://www.google.com.tr">Google</a>
```



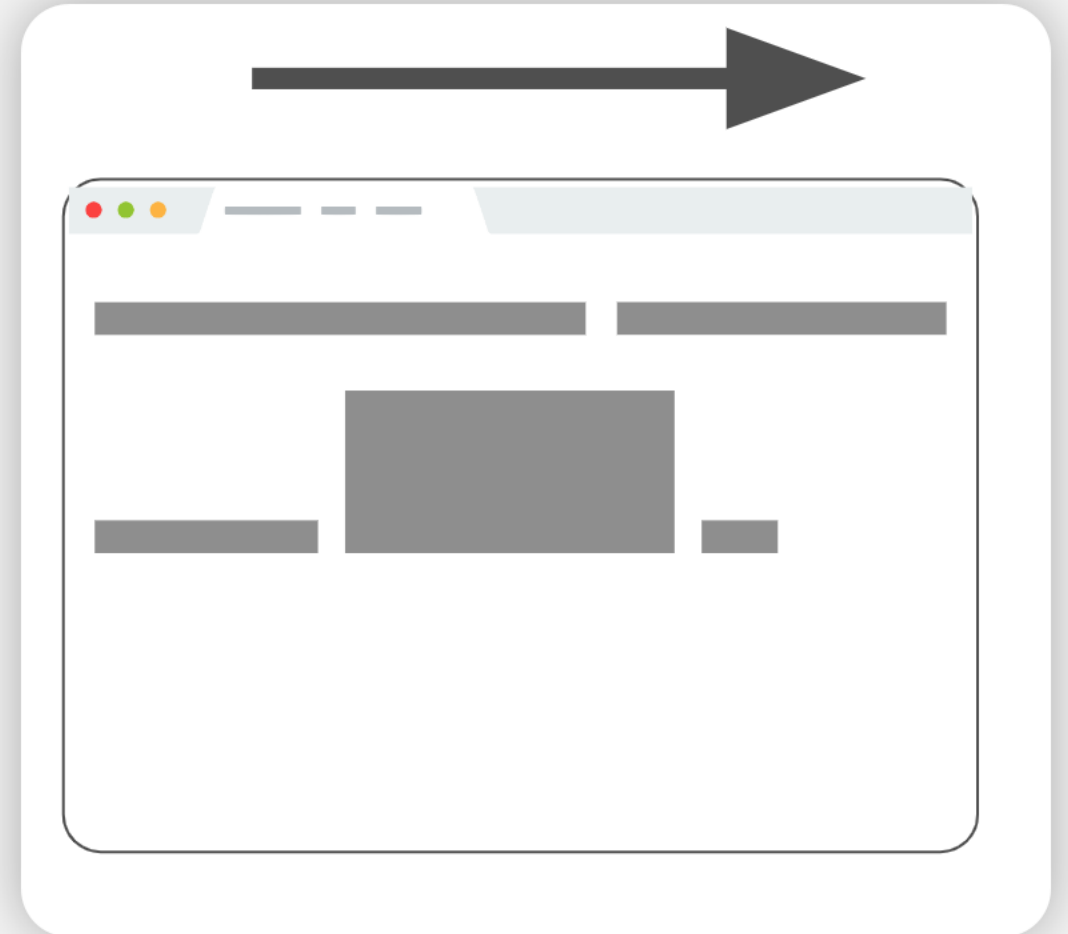
**Web programlama Öğreniyorum:** [HRU Web Sitesi](https://www.harran.edu.tr) [TUBİTAK](https://www.tubitak.gov.tr) [Google](https://www.google.com.tr)

İnline-satırıçi elemanlar width-genişlik değeri alsalar bile tarayıcı bunları yorumlamaz ve görmezden gelir.

# INLINE-BLOCK ELEMENTS

Örnekler: `<img>`

- Width değeri, içeriğin boyutudur; yani yalnızca gerektiği kadar alan kaplar (soldan sağa akar).
- Yükseklik ve genişliğe sahip olabilir.
- Bir blok elemanı içerebilir.
- Pozisyonlandırılabilir (yani, float ve position gibi CSS özellikleri uygulanır)



# INLINE-BLOCK EXAMPLE

```
img {  
  width: 50px;  
}
```

Tarayıcı da bu nasıl görünür?

```
  
  
  
  

```

<http://i.imgur.com/WJToVGv.jpg> =



# INLINE-BLOCK EXAMPLE



# CSS PROPERTIES (DISPLAY)

**display** özelliği bir elemanın öntanımlı gösterim özelliğini değiştirmek için kullanılır.

```
p {  
  display: inline;  
}
```

```
a {  
  display: block;  
}
```

```
display: block;
```

```
display: inline-block;
```

```
display: none;
```

```
display: flex;
```

```
display: grid;
```

Display,in alabileceği değerler bunlardır.  
Herbiri hakkında uygulamalı ve detaylı bilgiyi [burdan](#) alabilirsiniz.



## HATIRLATMA

- Eğer bir CSS özelliği çalışmıyorsa, block elmanına satır içi eleman özelliği eklemeye çalışıp çalışmadığınızı kontrol edin.



# DIV AND SPAN

Bu iki genel elemanın belirli bir amacı veya stili yoktur:

- <div>: genel bir **blok** elemanı
- <span>: genel bir **satır içi** eleman

# DIV AND SPAN

```
<h3>Web programlama Öğreniyorum:</h3>
<a href="https://www.harran.edu.tr">HRU Web Sitesi</a>
<a href="https://www.tubitak.gov.tr">TUBİTAK</a>
<a href="https://www.google.com.tr">Google</a>
</br>
<p> CREDITS: <strong>VICKI</strong></p>
```

**Web programlama Öğreniyorum:**

[HRU Web Sitesi](https://www.harran.edu.tr) [TUBİTAK](https://www.tubitak.gov.tr) [Google](https://www.google.com.tr)

**CREDITS: VICKI**

```
<h3>Web programlama Öğreniyorum:</h3>
<a href="https://www.harran.edu.tr">HRU Web Sitesi</a>
<a href="https://www.tubitak.gov.tr">TUBİTAK</a>
<a href="https://www.google.com.tr">Google</a>
</br>
<p> CREDITS: <span style="font-weight: bold;">VICKI</span></p>
```

**strong** gibi bir inline eleman yerine **span** kullanarak CSS yardımıyla aynı sonuca varabiliriz.

# CSS SELECTORS

HTML elemanları içerisinde yer alan bazı elemanlar için ayrıştırıcı CSS özelliği eklemek isteyebiliriz. Yada tekrarlayan elemanlar için belli bir şablon stil kullanabiliriz.

Bu tür bir ihtiyacı karşılamak için HTML elemanlarının **ID** ve **CLASS** özelliklerinden faydalanılır.

# IDS AND CLASSES

Eleman Seçici (Mevcutta kullandığımız)	<span>	Tüm span elemanlarını seçer.
ID seçici	#kimlik	id="kimlik" olan elemanı seçer. I html içinde tektir.
CLASS seçici	.sinif	class="sinif" olan tüm elemanları seçer. Bir eleman birden fazla class kullanabilir. Arada boşluk bırakarak kullanabilirsiniz.

```
<h1 id="kimlik">Proje</h1>  
<span class="sinif">Proje1</span> haftaya teslim.<br/>  
<span class="sinif uyari">Proje2</span> iki haftaya  
teslim.<br/>  
<span>All homework due at 11:59pm.</span>
```

# OTHER SELECTORS

Syntax	Kullanım	Örnek
<code>eleman.sinifadi</code> ( <code>selector.classname</code> )	<code>span.sinif</code>	<code>&lt;span class='sinif'&gt; Hello world</code>
<code>eleman eleman(selector selector)</code> ( <b>Decendent</b> )	<code>div .sinif</code>	<code>&lt;div&gt;   &lt;strong     class='sinif'&gt;Helloworld&lt;/strong&gt; &lt;/div&gt;</code>
<code>eleman, eleman(selector, selector)</code> ( <b>Decendent</b> )	<code>div, a { font-family: Arial }</code>	<code>&lt;div&gt;   &lt;a href='#'&gt;Link&lt;/a&gt; &lt;/div&gt;</code>
<code>eleman#id</code>	<code>div#kimlik</code>	<code>&lt;div id='kimlik'&gt;&lt;p&gt;Pick Me&lt;/p&gt;&lt;/div&gt;</code>

# SELECTOR TEST

<b>p</b>	• Tüm <p> elemanları
<b>.abc</b>	• class="abc" olan tüm elemanlar
<b>#abc</b>	• id="abc" olan eleman
<b>p.abc</b>	• class="abc" olan <p> elemanları
<b>p#abc</b>	• abc id'sine sahip <p> elemanı (p gereksiz)
<b>div strong</b>	• <div>'in alt elemanları olan <strong> elemanları
<b>h2, div</b>	• <h2> elemanları ve <div>'ler

# SELECTOR TEST

Yandaki CSS kodu neyi seçer?

CSS

```
#main li.important strong {  
  color: red;  
}
```

**Sağdan sola doğru okuyun:** <strong> elemanını kapsayan "important" sınıfına ait elemanları içeren li elemanlarını kapsayan "main" id'sine sahip elemanlar.

# CSS RULES

Eğer iki CSS kuralı çakışıyorsa, en spesifik olan(elemanı daha net tanımlayan) öncelik hakkı kazanır ve tarayıcı tarafından yorumlanır.

```
div strong { color: red; }  
strong { color: blue; }  
  
<div>  
    <strong>What color am I?</strong>  
</div>
```

id>sınıf

direkt hedeflenen eleman > kalıtım yoluyla alınan



# PSEUDO-CLASSES

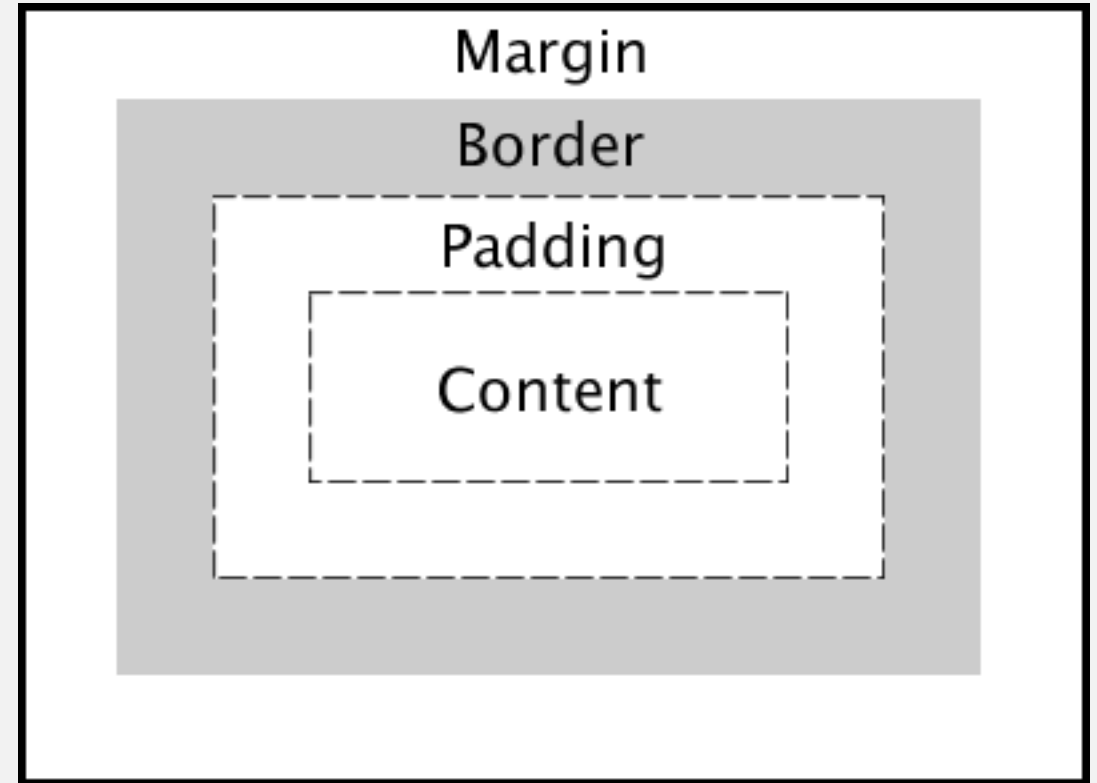
Sözdizimi	Açıklama
a	Tüm bağlantı etiketleri (bağlantılar) tüm durumlarda
a:visited	Ziyaret edilmiş bir bağlantı
a:link	Ziyaret edilmemiş bir bağlantı
a:hover	Bağlantının üzerine geldiğinizdeki stil
a:active	Bir bağlantıyı "etkinleştirdiğinizde" (tıklandığında) stil

Ödev:Aşağıda yazılı diğer Pseudo-Class türlerini araştırın.  
:hover, :active, :focus, :visited, :link,  
:first-child, :last-child, :nth-child(n),  
:not(selector), :checked, :disabled, :enabled

# CSS BOX MODEL

Her öge 4 katmandan oluşur:

- Öğenin içeriği
- Öğenin içeriğinin etrafındaki kenar
- İçerik ile kenar arasındaki boşluk (padding)
- Kenarın etrafındaki alanı temizleyen dış boşluk (margin)



# BORDER

**`border: width style color;`**

Border özellikleri tek tek tanımlanabilir.

```
border-style: dotted;  
border-width: 3px;  
border-color: purple;
```

```
p {  
  border: 1px solid magenta;  
  background-color: #AAAAAA;  
  padding: 10px; /* Optional: add some padding for better appearance */  
  text-align: left; /* Align paragraph content to the left */  
}
```

Border her bir kenar için tanımlanabilir

```
border-top  
border-bottom  
border-left  
border-right
```

`border-radius: 10px;`

Dört değer - `border-radius: 15px 50px 30px 5px;` (ilk değer üst-sol köşeye, ikinci değer üst-sağ köşeye, üçüncü değer alt-sağ köşeye ve dördüncü değer alt-sol köşeye uygulanır):

# MARGIN

Margin, kenar ile diğer öğeler arasındaki boşluktur.

margin-top, margin-bottom, margin-left, margin-right belirtilerek tanımlanabilir. Margin değeri negatif olabilir. Ör: -10px

Kısayol:

margin: 2px 4px 3px 1px; <- üst|sağ|alt|sol

margin: 10px 2px; <- üst+alt|sol+sağ

Harran Üniversitesini

Web Programlama

Ödev Tablosu.

margin: 0px;  
border: 2px solid magenta;



Harran Üniversitesini

Web Programlama

Ödev Tablosu.

margin: 25px;  
border: 2px solid magenta;

# PADDING

Padding, kenar ile içerik arasındaki boşluktur.

padding-top, padding-bottom, padding-left, padding-right belirtilerek tanımlanabilir.

Ayrıca bir kısayol var:

padding: 2px 4px 3px 1px; <- üst|sağ|alt|sol

padding: 10px 2px; <- üst+alt|sol+sağ

Bu elemanın kenarlarla arası çok yakın.



Bu elemanın kenarlarla arasında 25 piksel var.

padding: 0px

padding: 25px

# HTML HAKKINDA

- HTML sayfalar diğer birçok programlama dilinin aksine hata olsa bile sayfayı olabildiğince en iyi şekilde göstermeye eğilimlidir.
- Örneğin bir elemanı kapatmasanız veya bir elemanın özelliklerini doğru yazmazsanız bile sayfa görüntülenir. Burada amaç kullanıcıya kötü bir deneyim sunmamaktır.
- Fakat interneti ve ağı etkileyen durumlar için özel hata sayfaları yer almaktadır. Örnek: 404, 403, 201

# WHAT IS MORE FOR CSS

<https://www.w3schools.com/css/default.asp>

Websitesini ziyaret edip CSS özelliklerini ve nasıl kullanıldığını test edebilirsiniz.

Tarayıcı içerisinde bir elemanın CSS değerini görebilirsiniz ve düzenleyip canlı olarak gözlemleyebilirsiniz. Bunun için Developer Tool'lar vardır

Chrome için Videolu anlatım: <https://www.youtube.com/watch?v=Bf5DCpbeTIQ>

# PORTFOLIO PROJECT PHASE I

Daha fazla bilgi ilgili haftanın yanında PDF olarak sunulmuştur.

Bknz: [Link](#)

## Simba The Cat



[Hakkında](#) [Projeler](#) [Yetenekler](#) [İletişim](#)

### Hakkında

Merhaba! Ben Simba The Cat, bir yazılım geliştiricisiyim. Web teknolojileri ve veri bilimi alanlarında çalışıyorum. Projelerimi ve yeteneklerimi bu portföyde bulabilirsiniz.

### Projeler

TA - To Be Announced

### Yetenekler

- HTML & CSS
- JavaScript
- Python
- Veri Analizi

### İletişim

Bana ulaşmak için: [simba@example.com](mailto:simba@example.com)





EOF\*

\*End of Fun/File

This Photo by Unknown Author is licensed under [CC BY-SA](#)



# REFERENCES / CREDITS

1. Stepp M, Miller J, Kirst V. Web Programming Step by Step. Step by Step Publishing; 2012.
2. CS193X Web Programming Fundamentals Course Slides at Stanford Uni by Victoria Kirst
3. <https://medium.com/swlh/an-introduction-to-git-and-github-22ecb4cb1256>