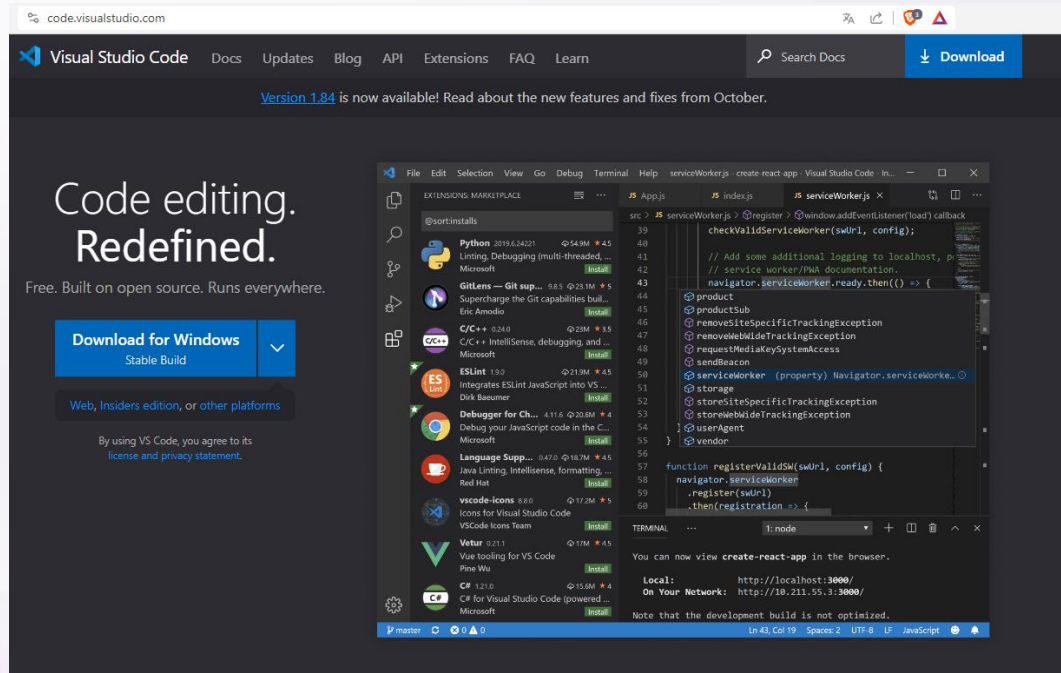




Wise Quarter

Batch : HYBRID
Date : COURSE
Lesson : HTML CSS
Subject : HTML
Instructor : BABAYIGIT

VSCODE KURULUM



VSCODE KURULUM



Microsoft Visual Studio Code (User) - Kurulum Yardımcısı

Lisans Anlaşması

Lütfen devam etmeden önce aşağıdaki önemli bilgileri okuyun.

Lütfen Aşağıdaki Lisans Anlaşmasını okuyun. Kurulumu devam edebilmek için bu anlaşmayı kabul etmelisiniz.

Bu lisans, Visual Studio Code ürünü için geçerlidir. Visual Studio Code için Kaynak Kodu <https://github.com/Microsoft/vscode> adresinde, <https://github.com/microsoft/vscode/blob/main/LICENSE.txt> adresindeki MIT lisans sözleşmesi altında bulunmaktadır. Ek lisans bilgileri <https://code.visualstudio.com/docs/supporting/faq> adresindeki SSS'lerimizde bulunabilir.

MICROSOFT YAZILIMI LİSANS ŞARTLARI

MICROSOFT VISUAL STUDIO CODE

☒ Anlaşmayı kabul ediyorum.
☐ Anlaşmayı kabul etmiyorum.

Sonraki > İptal

Microsoft Visual Studio Code (User) - Kurulum Yardımcısı

Hedef Konumunu Seçin

Visual Studio Code nereye kurulsun?

Visual Studio Code uygulaması şu klasöre kurulacak.

Devam etmek için Sonraki üzerine tıklayın. Farklı bir klasör seçmek için Gözet üzerine tıklayın.

C:\Users\murat\AppData\Local\Programs\Microsoft VS Code Gözet...

En az 357.2 MB boş disk alanı gereklidir.

< Önceki Sonraki > İptal

Microsoft Visual Studio Code (User) - Kurulum Yardımcısı

Kurulmaya Hazır

Visual Studio Code bilgisayarınıza kurulumaya hazır.

Kurulumu devam etmek için Sonraki üzerine, ayarları gözden geçirip değiştirmek için Önceki üzerine tıklayın.

Hedef konumu:
C:\Users\murat\AppData\Local\Programs\Microsoft VS Code

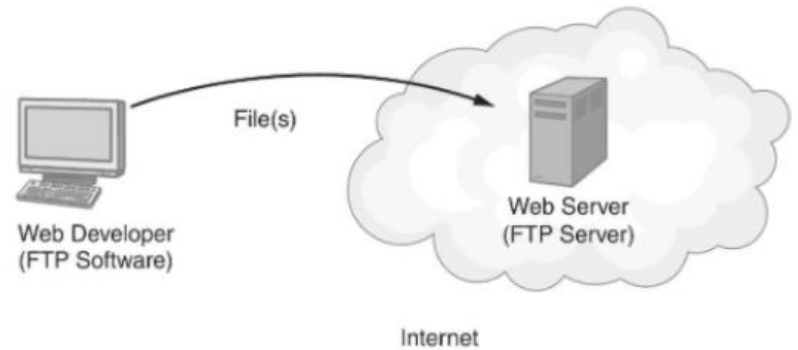
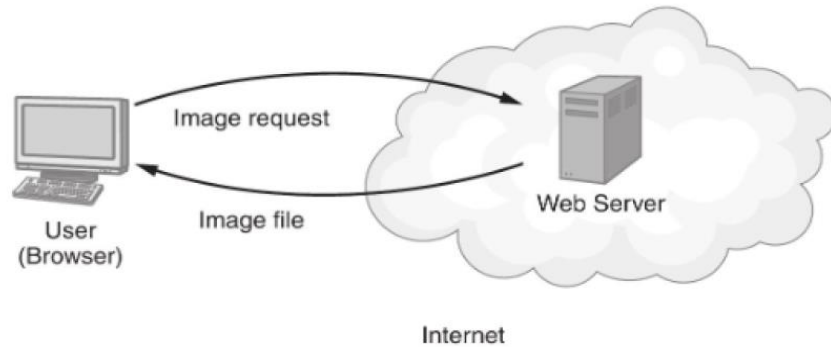
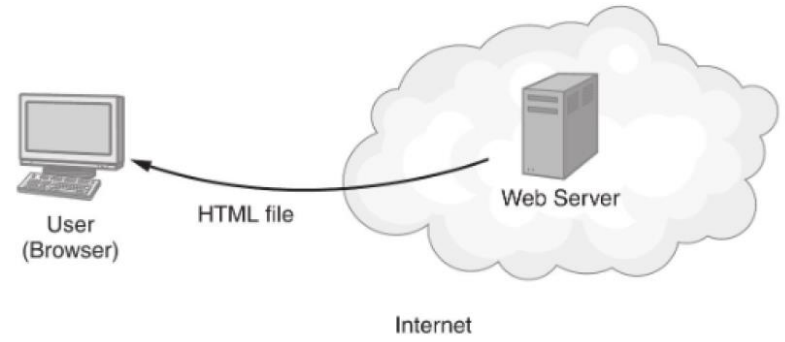
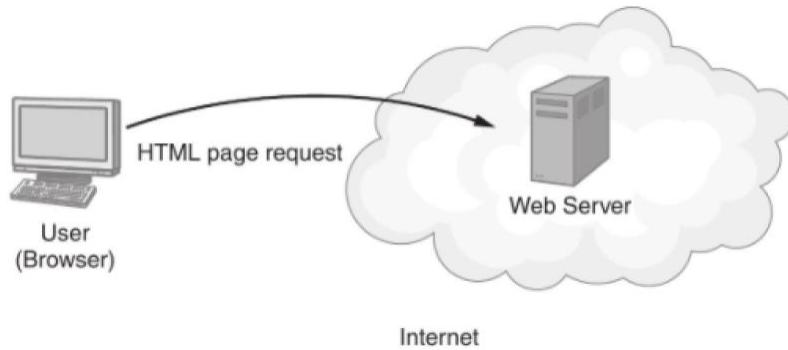
Bağlat Menüsü klasörü:
Visual Studio Code

Ek işlemler:
Diğer:
Code uygulamasını desteklenen dosya türleri için bir düzenleyici olarak kayıt et
PATH'e ekle (yeniden bağlatıldıktan sonra kullanılabilir)

< Önceki Kur İptal

HTML

Hypertext Markup Language



HTML NEDİR?

Web sayfaları hazırlamak için kullanılan bir işaretleme dilidir.

HTML bir programlama dili değildir, işaretleme dilidir.
(Hyper Text Markup Language)

- Html komutlarına **tag** denir.
- Taglar büyüktür küçüktür işaretlerinin arasına yazılır <>

Örnek : <html> <u> <p>

- İstisnalar dışında tagların başlangıç ve bitişleri vardır.

Bitiş tagları "/" işareti ile belirtilir.

Örnek : Merhaba <p>Nasılsın</p>

Version	Date
HTML 1.0	1989
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML 1.0	2001
XHTML 2.0	discontinued in 2009
HTML 5.0	2012
HTML 5.2	2017

two-sided tag
enclosing element
content

empty elements,
which do not
contain content

several elements
nested within
another element

```
<!DOCTYPE html>
<html>
<head>
  <title>Curbside Thai</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <link href="ct_base.css" rel="stylesheet" />
  <link href="ct_layout1.css" rel="stylesheet" />
</head>

<body>
  <header>
    
  </header>
  <nav>
    <a href="ct_about.html"></a>
    <a href="ct_locations.html"></a>
    <a href="ct_menu.html"></a>
    <a href="ct_reviews.html"></a>
    <a href="ct_catering.html"></a>
    <a href="ct_contact.html"></a>
  </nav>
  <footer>
    Curbside Thai ☎️ 411 Belde Drive, Charlotte NC 28201 ☎️ 704-555-1151
  </footer>
</body>
</html>
```

an element attribute

Element Hiyerarşisi

Bir HTML belgesinin tüm yapısı iç içe geçmiş hiyerarşik bir ağaçtaki öğeler gibi bir dizi olarak düşünülebilir. Ağacın en üstünde html elemanı bulunur ve tüm belge. Html öğesinin içinde, bilgileri çevreleyen head ögesi bulunur belgenin kendisi ve web sayfasının içeriğini çevreleyen gövde ögesi hakkında. Bu nedenle bir HTML dosyasının genel yapısı şöyledir

```
<!DOCTYPE html>  
<html>  
  
  <head>  
    <title> BAŞLIK </title>  
  </head>  
  <body>  
  
    </body>  
  
</html>
```

burada head content ve body content iç içe yerleştirilmiş öğelerdir. belge başı ve gövdesi. Gövde öğesinin her zaman baş öğesinden sonra yerleştirildiğini unutmayın

HTML, web sayfaları oluştururken kullanılan bir işaretleme dilidir. İngilizce adıyla, "Hyper-text Markup Language", Türkçe adıyla "Hiper Metin İşaretleme Dili"nin kısaltmasıdır.

Bir web sayfasına girildiğinde, son kullanıcının gördüğü her zaman HTML çıktısıdır. Onun daha güzel ve düzenli görünmesini sağlayan CSS, daha dinamik bir yapıda kullanılmasını sağlayan ise JavaScripttir.

CSS ve JavaScript konularına, HTML'den sonra bakılması yeni başlayacak olanların faydasına olacaktır. Zira HTML'i iyi öğrendikten sonra sırasıyla CSS ve JavaScript, front-end geliştirme yapmak isteyenlerin izlemesi gereken bir yoldur.

- HTML, uzun adıyla "Hyper Text Markup Language" demektir.
- HTML, Türkçe adıyla "Hiper Metin İşaretleme Dili" demektir.
- HTML ile yazılar işaretlenerek, web sayfalarının yapıları belirlenir.
- HTML ile işaretleme yapılırken belli başlı html etiketleri kullanılır.
- HTML etiketleri, "başlık", "paragraf", "tablo" vb. bölümleri işaretleme için kullanılır.

i UNUTMAYIN

Web tarayıcıları, HTML etiketlerini göstermez ancak sayfaların içeriklerini oluşturmak için bu etiketleri kullanır. Bu yüzden kurallarına uygun HTML yazmak, hem tarayıcıların kodlarınızı yorumlamasını kolaylaştırır hem de arama motoru dostu sayfalar oluşturmanızı sağlar.

HTML sayfası oluşturmak

Bir web sayfası oluşturmak için yapılması gerekenler:

**Html uzantılı
dosya oluştur**



**html
head
body
/html**

HTML sayfası oluşturmak

Bir web sayfasının en temel şablonu :

```
<html>
|   <head>
|   |   <title>Ilk sayfam</title>
|   </head>
|   <body>
|   |   Merhaba. Bu benim ilk web sayfam
|   </body>
| </html>
```

HTML5

Bir html sayfasının tarayıcılar tarafından Html5 olarak algılanabilmesi için:
Sayfanın en üst kısmına
<!doctype html>
eklenmesi gerekir.

```
<!DOCTYPE html>  
<html>  
  
</html>
```

HTML

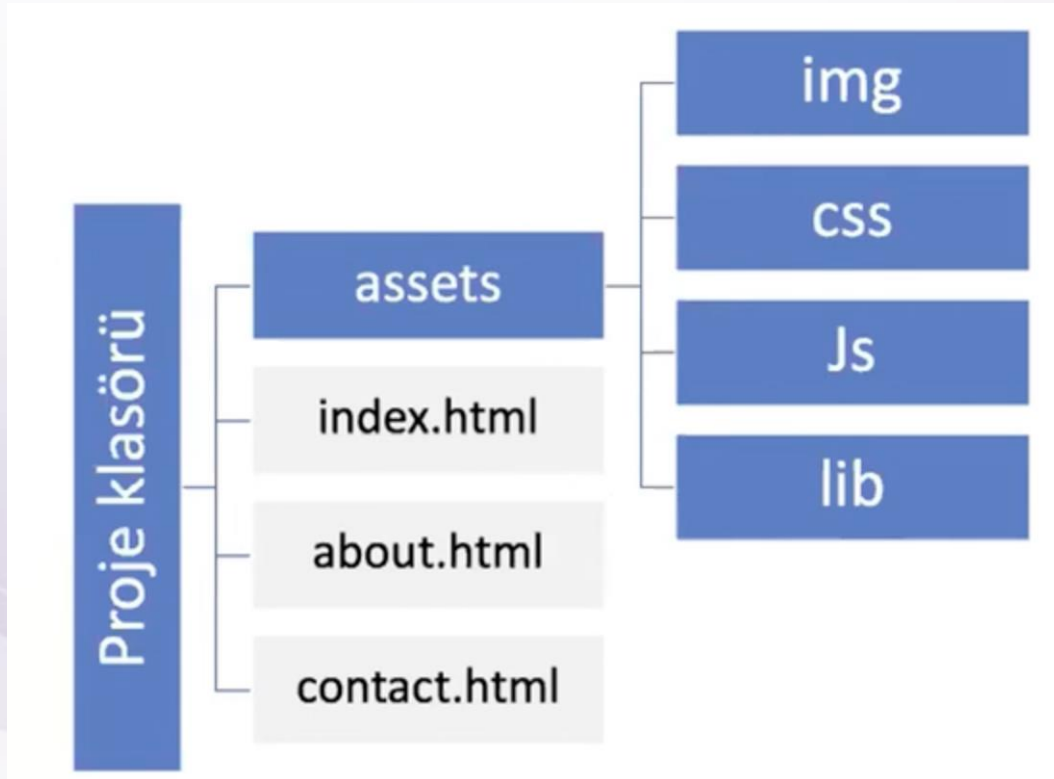


HTML bir site oluşturma

1. Desktop'umuzda MyPage isimli bir klasör, içine de index.html dosyası ve assets klasörü, assts içine de img, css, js, lib isimli klasörler oluşturuyoruz.
2. index.html dosyasına web sayfası için gerekli olan tagları oluşturuyoruz.

İpucu: VS Code'da ! işareti yazıp enter'a basarsanız gerekli taglar oluşturulur.

KLASÖR YAPISI



Temel biçimlendirme tagları

p ➔
paragraph

Yazıları paragraf haline getirir.

`<p>`Lorem Ipsum, adı bilinmeyen bir matbaacının bir hurufat numune kitabı oluşturmak üzere bir yazı galerisini alarak karıştırdığı 1500'lerden beri endüstri standardı sahte metinler olarak kullanılmıştır.`</p>`

Temel biçimlendirme tagları

br ➔

break

Metin içinde satır başı yapmak için kullanılır.

`<p>`Lorem Ipsum, adı bilinmeyen bir matbaacının bir
hurufat numune kitabı oluşturmak üzere `
`
bir yazı galerisini alarak karıştırdığı 1500' lerden beri
endüstri standardı sahte metinler olarak kullanılmıştır.`</p>`

HTML kodlarken alt satıra geçmek için enter tuşuna basılmasının veya kelimeler arasında birden fazla boşluk bırakmak için space tuşuna basılmasının ziyaretçi tarafında bir etkisi yoktur.



Temel biçimlendirme tagları

h_x →
head

Yazıları başlık haline getirir.

Standart 6 tane başlık tagı bulunmaktadır.

`<h1>Başlık 1</h1>`

`<h2>Başlık 2</h2>`

`<h3>Başlık 3</h3>`

hx taglarının kullanımı

<h1>IT ALANINDAKİ MESLEKLER</h1>

<p>Bu yazımızda IT sektöründe uzmanlaşabileceğiniz meslekleri anlattım.</p>

<h2>Automation Engineer</h2>

<p>...</p>

<h3>Full Stack Developer</h3>

<p>...</p>

Head taglarının kullanımı

IT ALANINDAKİ MESLEKLER

Bu yazımızda IT sektöründe uzmanlaşabileceğiniz meslekleri anlattım.

Automation Engineer

...

Full Stack Developer

.....

Temel biçimlendirme tagları

hr
horizontal
row



Yatay çizgi oluşturmak için kullanılır.

<hr>

Lorem Ipsum, adı bilinmeyen bir matbaacının bir hurufat numune kitabı

oluşturmak üzere bir yazı galerisini alarak karıştırdığı 1500'lerden beri endüstri standardı sahte metinler olarak kullanılmıştır.

Temel biçimlendirme tagları

strong ➡

Metinleri kalınlaştırmak için kullanılır.

<p>Lorem Ipsum, adi bilinmeyen bir matbaacının **** bir hurufat numune kitabı **** oluşturmak üzere bir yazı galerisini alarak karıştırdığı 1500'lerden beri endüstri standardi sahte metinler olarak kullanılmıştır.**</p>**

Temel biçimlendirme tagları


i
italic



Yazıyı eğik (*italik*) yazdırır.

`<i>`Lorem Ipsum, adi bilinmeyen bir matbaacının bir hurufat numune kitabı oluşturmak üzere bir yazı galerisini alarak karıştırdığı 1500'lerden beri endüstri standardi sahte metinler olarak kullanılmıştır.`</i>`

İç içe taglar



İç içe tag kullanıldığında dikkat edilmesi gereken kural :
Önce açılan tag en son kapatılır.

`<p>Merhaba<u>Dunya</u></p>`

Temel biçimlendirme tagları

U



underline

Yazının altını çizmek için kullanılır.

Bu yazının altını çizer

div & span

div →

Blok elemandır, kapladığı alan ne olursa olsun, tüm satırları işgal eder.

span →

Inline elemandır, sadece kapladığı alan kadar yer işgal eder.

Resume Sitesi Oluřturma



Tüm sayfalarımızın ortak iskeleti

HEADER

CONTENT

FOOTER

Resume Sitesi Oluřturma



Home sayfamızın iskeleti

AD SOYAD ÜNVAN

MENÜ

TELEFON

EMAIL

SOCIAL

COPYRIGHT

Resume Sitesi Oluşturma

- Home sayfası için header, content ve footer bölümlerini oluşturunuz.
- Header bölümündeki ad, soyad, unvan ve menü bölümlerini oluşturunuz.
- Footer bölümündeki telefon, email, sosyal medya ve copyright bölümlerini oluşturunuz.
- Bu işlemleri tamamladıktan sonra **about.html** ve **contact.html** dosyalarımıza kopyalıyoruz. Bu sayede ortak olan alanları hepsi için tek seferde tamamlamış oluyoruz. Her bir sayfanın Content bölümünü de ayrı ayrı şekillendireceğiz.

Resume Sitesi Oluřturma



Home sayfamızın Content bölümü

AD SOYAD ÜNVAN

MENÜ

FOTOĞRAF

KISA AÇIKLAMA

TOOLBAR

TELEFON

EMAIL

SOCIAL

COPYRIGHT

Resume Sitesi Oluřturma

Home sayfamızın Content bölümü

AD SOYAD ÜNVAN

MENÜ

FOTOĞRAF

KISA AÇIKLAMA

TOOLBAR

TELEFON

EMAIL

SOCIAL

COPYRIGHT

Resume Sitesi Oluşturma

- Home sayfası content bölümündeki about kısmını oluşturunuz.
- Fotoğraf için gerekli alanı oluşturunuz.
- Kısa açıklama kısmını oluşturunuz.
- Kısa açıklama içinde toolbar bölümünü oluşturunuz.

Bağlantı tagı

a
anchor



Web sitelerinde,

- Bir sayfadan başka bir sayfaya,
- Farklı bir siteye

Bağlantı vermek için `<a>` tagı kullanılır.

Bağlantı tagı

a ►
anchor

Başka bir siteye link vermek için;

```
<a href="https://google.com.tr">Google' a Git</a>
```

Bir dosyaya link vermek için;

```
<a href="img/resim1.jpg">Fotoğrafı aç</a>
```

Bağlantıyı yeni bir sekmede açmak için;

```
<a href="https://google.com.tr" target="_blank">Google' a Git</a>
```

Resume Sitesi Oluşturma

- Header bölümündeki menü bölümünün bağlantılarını oluşturunuz.
- Content kısmındaki toolbar bölümünün bağlantılarını oluşturunuz.
- Footer bölümündeki linklerin bağlantılarını oluşturunuz.

Fotoğraf yükleme

img ▶
image

Fotoğraf ekleme işlemleri img tagi ile yapılır.

```

```

Resume Sitesi Oluřturma

→ Home page'deki Content bölümündeki fotoğrafı ekleyiniz.

unsplash.com, pexels.com,
flaticon.com ücretsiz fotoğraf
kullanabileceğiniz bir sitedir.

Resume Sitesi Oluřturma

Resume sayfamızın Content bölümü

Page Title

Resume Sitesi Oluşturma

- Resume page'deki Content bölümündeki Page Title, Education ve Work Experience alanlarını oluşturunuz.

Resume Sitesi Oluřturma

Contact sayfamızın Content bölümü

Page Title

Resume Sitesi Oluřturma

→ Contact page'deki Content bölümündeki Page Title ve Form alanlarını oluřturunuz.

Form oluşturma

Kullanıcıdan bilgi almak için kullanılan yapılara form denir. Form içinde textbox, button, radio button, checkbox vb elemanlar olabilir.

- **form** : Tüm form elemanlarını içinde barındıran taşıyıcıdır.
- **input** : Kullanıcıdan bilgi almak için kullanılan form elemanlarının genel ismi.
- **label** : Kullanıcıya form elemanı ile ilgili bilgi vermek için kullanılan eleman.

Form oluşturma

```
<form>  
<label>Eposta</label>  
<input type="text">  
<input type="button" value="Kaydet">  
</form>
```

Eposta

Kaydet

Input

```
<input type="text | email | hidden ...." name="adi" value="ali" placeholder="">
```

text → Genel bilgi girişi için kullanılır.

number → Sadece rakam girişi için kullanılır.

email → Sadece e-posta girişlerini kabul eder.

password → Şifre girişi için kullanılır. Girilen değer gösterilmez.

Input

```
<input type="radio | checkbox | file ...." name="hobiler" value="futbol">
```

checkbox → Birden fazla işaretleme yapılabilecek seçenekler sunar.

Sadece tek seçim yapabileceği seçenekler sunar.

radio → Cinsiyet gibi. Name attribute'u aynı olan radio butonlar kendi aralarında grup olurlar.

file → Dosya yüklemek için kullanılır.

Select

select → Kullanıcıya listeden seçim yaptırmak için kullanılır. Ülke, şehir gibi seçimler bu şekilde yapılır. Bunun için select tagı kullanılır. Her bir seçenek ise option tagı içine konulur.

```
<select name="iller">  
  <option value="01">Adana</option>  
  <option value="02">Adıyaman</option>  
  <option value="03">Afyon</option>  
  <option value="04">Ağrı</option>  
</select>
```

Textarea

textarea → Çok uzun miktarda yazı girilmesini sağlayan form elemanı textarea'dır. Geçerli değer textarea tagları arasına yazılabilir.

```
<textarea name="mesaj" rows="4"></textarea>
```

Rows attribute ile textarea nın aynı anda gösterebileceği satır sayısı belirlenir

Button

```
<input type="button | submit | reset" name="btn" value="Kaydet">
```

- button** → Genel amaç için button oluşturur. Butonun default olarak bir görevi yoktur.
- submit** → Form bilgilerini backend'e gönderen bir buton oluşturur.
- reset** → Formdaki bilgileri resetleyen bir buton oluşturur

Form oluşturma

Submit butonlara basıldığında formdaki bilgiler sunucuya gönderilir. Bu gönderme işlemi yapılırken inputların name attribute'u ile value attribute'u kullanılır.

```
<form action="saveDatabase.php">
  <label>Ad</label>
  <input type="text" name="ad" value="">
  <br>
  <label>Şifre</label>
  <input type="password" name="sifre" value="">
  <br>
  <label>Yaş</label>
  <input type="number" name="yas" value="">
</form>
```



ad=Ali
sifre=12345
yas=25

Resume Sitesi Oluřturma

→ Contact bölümündeki formu oluřturunuz.

CSS

Cascading Style Sheet

(Basamaklı Stil Şablonları)

Murat BABAYIGIT

Full-Stack Automation Engineer - Instructor

Css nedir?

HTML elemanlarının stil özelliklerini belirleyen bir işaretleme dilidir. CSS sayesinde tek bir tanımlama ile birçok elementin özelliği aynı anda değiştirilebilir. HTML ile sayfanın iskeleti ve içeriği tanımlanırken, CSS ile biçimlendirmesi yapılır.

CSS eklemek için ilgili tagın içine style attribute' u eklenir. Buna inline stil denir.

```
<p style="color: red">Merhaba</p>
```

Property
(Özellik)

Value
(Değer)

Birden fazla stil ekleneceği zaman, stil özelliklerinin arasına ; (noktalı virgül) konulur.

```
<div style="color:red; text-align:center">Merhaba</div>
```

İnline stiller sadece bulundukları tagın içinde etkilidirler.



Temel CSS özellikleri

font-size

Yazının büyüklüğünü değiştirmek için kullanılır.

```
<a href="sayfa1.html" style="font-size: 20px">click</a>
```

text-align

Yazının bulunduğu tag içindeki yataydaki hizasını belirlemek için kullanılır. Kullanılabilecek değerler: **left, right, center, justify**

```
<div style="text-align: left">Sitemize hoşgeldiniz</div>
```



Temel CSS özellikleri

color

Yazının rengini değiştirmek için kullanılır. Kullanılabilecek değerler: **hazır renk tanımları(red, green, yellow ...), hexadecimal system (#12AF45), rgb, rgba**

```
<p style="color:#1255FA">Merhaba</p>
```

background-color

Html elemanının zemin rengini değiştirmek için kullanılır. Kullanılabilecek değerler color ile aynıdır.

```
<p style="background-color:#1255FA">Merhaba</p>
```

Sayfa seviyesinde style (external)

Site içindeki tüm sayfalarda etkili olabilecek stil tanımlaması yapmak için harici stil sayfaları kullanılır.

- Bunun için css uzantılı bir dosya oluşturulup tüm stil işlemleri bunun içinde yapılır.
- Oluşturulan bu stil dosyası ile web sayfaları <link> komutu ile ilişkilendirilir. <link> tagı head tagı içinde tanımlanır.

```
<link href="css/stil.css" rel="stylesheet">
```

Seçiciler (Selectors)

- Sadece istenilen belli html elemanlarına stil uygulamak için selector'lerin doğru kullanılması gerekir.

1

Tüm html elemanlarını seçmek için * işareti kullanılır.

```
* {  
  text-align: center;  
  color: blue;  
}
```

2

Belli türdeki html elemanlarını seçmek için elemanın ismi seçici olarak kullanılır.

```
h1 {  
  text-align: right;  
  font-weight: 300;  
}
```

Temel CSS özellikleri

3

Bazı html elemanlarını seçmek için class tanımlaması yapılır.

CSS

```
.ortala {  
  text-align: center;  
  color: blue;  
}
```

```
#ortala {  
  text-align: center;  
  color: blue;  
}
```

HTML

```
<p class="ortala">Merhaba</p>
```

```
<p id="ortala">Merhaba</p>
```


Metin Stilleri

text-decoration

Metnin altına, üstüne veya üzerine çizgi çekmek için kullanılır. Alabileceği değerler **underline**, **overline**, **line-through**, **none**

```
a{  
    text-decoration: none;  
}
```

font-style

Yazıyı italik hale getirmek için kullanılır. Alabileceği değerler **normal**, **italic**

```
font-style : italic;
```

font-weight

Yazıyı kalınlaştırmak için kullanılır. Alabileceği değerler **100 – 900**, **bold**, **bolder**, **lighter**

```
font-weight : 500;
```

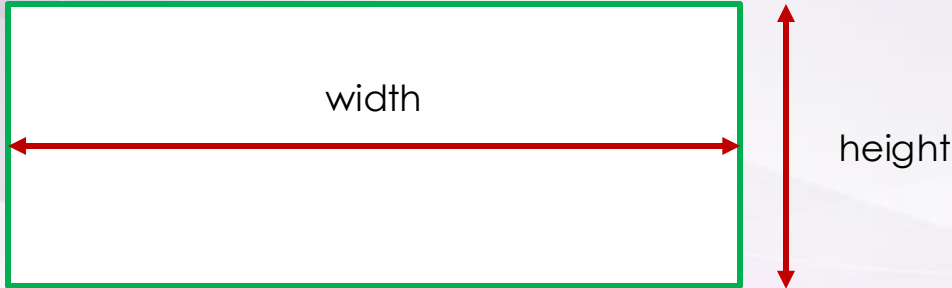
Temel CSS özellikleri

width

Html taglarının genişliğini (yatayda kapladığı alanı) ayarlamak için kullanılır.

height

Html taglarının yüksekliğini (dikeyde kapladığı alanı) ayarlamak için kullanılır.



Width ve height için %, px birim olarak kullanılabilir.

Border (Çerçeve)

- Elementlere çerçeve vermek için border stili uygulanır.

```
border: genişlik tip renk;
```

```
border: 3px solid red;
```

```
border: 3px dashed blue;
```

```
border: 5px dotted gray;
```

Tip için verilebilecek değerler: **dotted, dashed, solid, double, groove, ridge, inset, outset, none**

Border (Çerçeve)

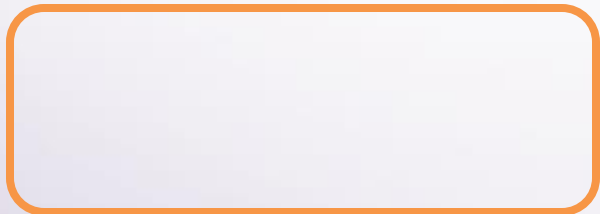
- Elementlerin 4 tarafına border verilebildiği gibi sadece istenilen tarafına da verilebilir.

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

```
border-top: 3px solid red;
```

Border (Çerçeve)

- Çerçevelerin köşelerini yumuşatmak için border-radius stili kullanılır.



```
div{  
    border:2px solid red;  
    border-radius: 8px;  
}
```



```
div{  
    background-color:red;  
    border-radius: 8px;  
}
```

Shadow (Gölge)

Yazılara gölge vermek için

text-shadow

Elementlere gölge vermek için

box-shadow

x y opacity color;

Gölgenin dikey uzaklığı *

Gölgenin yatay uzaklığı *

Şeffaflık

Renk

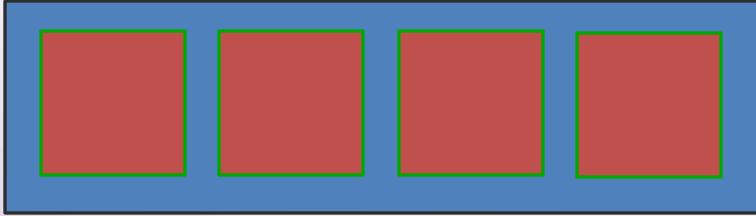
```
box-shadow: 5px 5px 10px red
```

Resume Sitesi

- Resume sitesi için harici stil dosyası oluşturunuz.
- Tüm elemanlar için box-sizing tanımlaması yapalım.
- Body için font ve padding ayarlaması yapınız.
- Tüm linklerdeki alt çizgiyi kaldırınız.
- Form inputlara border, padding ve border-radius verelim.
- Buton'lara border, border-radius, padding ve background-color verelim.

Flexbox

- Flexbox, html elemanlarının sayfa üzerinde yerleşimlerini ayarlamak için kullanılan bir CSS özelliğidir.



```
<div class="container">
  <div></div>
  <div></div>
  <div></div>
  <div></div>
</div>
```

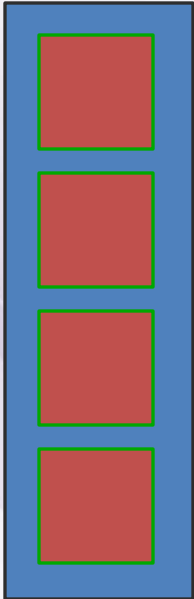
```
.container{
  display: flex;
}

.container div{
  width:160px;
  height: 160px;
  background-color:
#13aa45;
}
```


Flexbox

flex-direction

Flex-direction ile bu dizilimin yönü belirlenebilir. Row olursa yan yana, column olursa alt alta gösterilir. Default değer: row



```
<div class="container">
  <div></div>
  <div></div>
  <div></div>
  <div></div>
</div>
```

```
.container{
  display: flex
  flex-direction:
column;
}

.container div{
  width:160px;
  height: 160px;
  background-color:
#13aa45;
}
```

Flexbox

justify-content

Flex içindeki elemanları hizalamak için kullanılır.

flex-start | flex-end | center | space-between | space-around

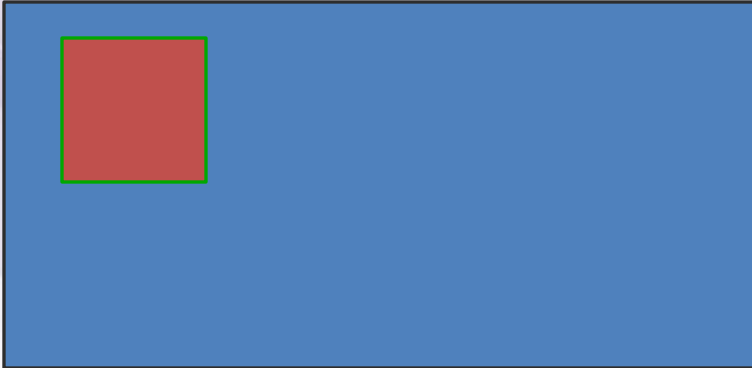
Baş tarafa
hizalar

Son tarafa
hizalar

Ortalar

Eşit boşluklarla
satıra yayar.

Eşit boşluklarla satıra yayar.
Başta ve sonda da boşluk verir.



```
.container{  
    display: flex;  
    justify-content: flex-  
start;  
}
```

```
<div class="container">  
    <div></div>  
</div>
```

Flexbox

align-items

Flex içindeki elemanları hizalamak için kullanılır.

stretch | center | flex-start | flex-end

Taşıyıcı yüksekliğine
sığdırır

Ortalar

Üste hizalar

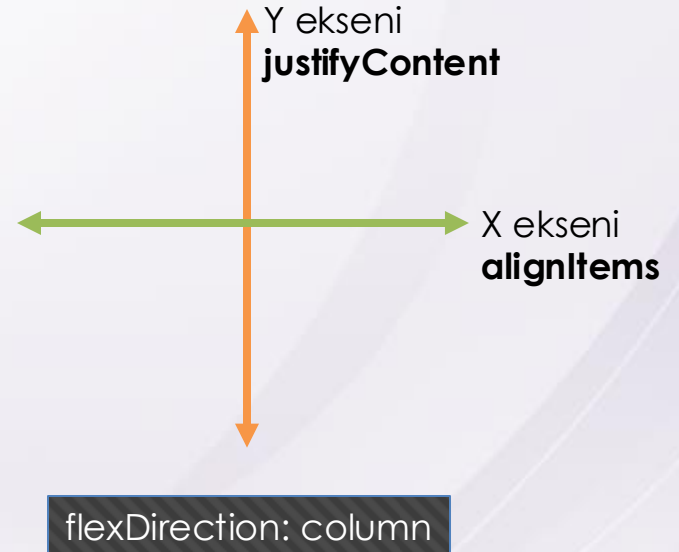
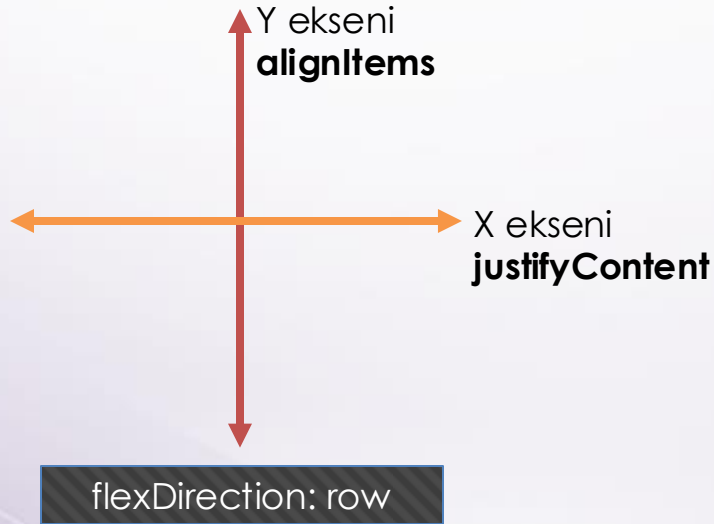
Alta hizalar



```
.container{  
    display: flex;  
    align-items : center;  
}
```

```
<div class="container">  
    <div></div>  
</div>
```

Flexbox



Flex-direction:

row olarak ayarlandığında **justifyContent** X ekseninde, **alignItems** Y ekseninde hizalama yapar.

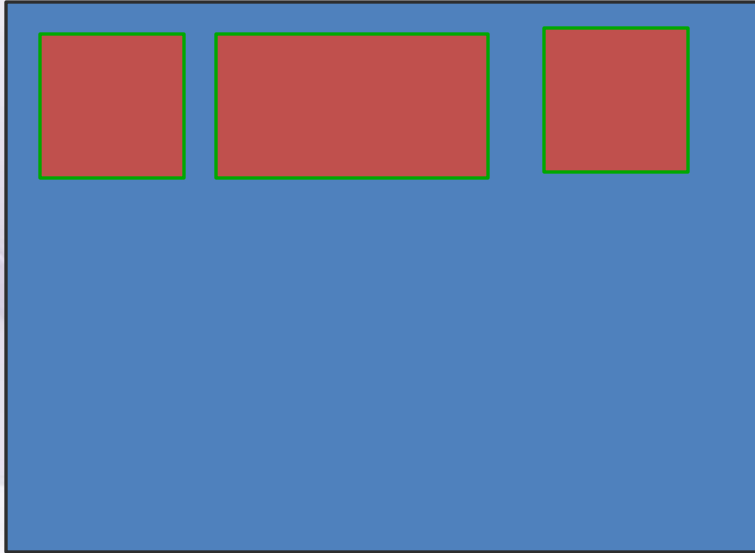
Flex-direction:

column olarak ayarlandığında ise tam ters davranırlar.

Flexbox

flex

Aynı flexbox içindeki elemanlara göre büyüklük oranını belirler.



```
.container{  
    display: flex;  
}  
  
.container .second{  
    flex:2  
}
```

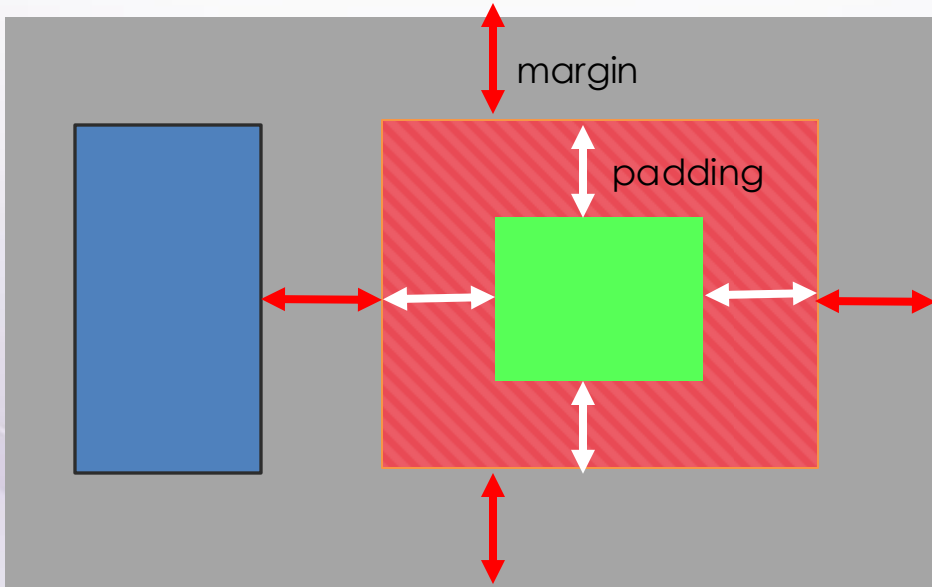
```
<div class="container">  
    <div></div>  
    <div class="second"></div>  
    <div></div>  
</div>
```

Resume Sitesi

Aşağıdaki elemanlar için flexbox ayarlaması yapınız :

- Header
- Content
- Card
- About
- Description
- Toolbar
- Footer

Padding & Margin



```
<div>  
    <div class="kutu">  
        <div>  
        </div>  
    </div>  
</div>
```

```
.kutu{  
    padding: 50px;  
    margin: 90px;  
}
```

padding

Elementin içindeki diğer elementlerin kenarlardan olan uzaklığını belirler.

margin

Elementin dıştaki elementlerden olan uzaklığını belirler.

Resume Sitesi

- Header'daki ad soyad bölümün başına yuvarlak bir ikon oluşturunuz.
- Page title bölümün başına yuvarlak bir ikon oluşturunuz.
- Toolbar ve card'lar için gölge oluşturunuz.

INSTALLATION



Visual Studio Code

JavaScript Fundamentals

- Hello World - Variables
- Data Types
- Concatenation & Interpolation
- Objects & Arrays
- Operators
- Conditional Statements
- JavaScript functions

Package.json oluşturalım

- Masaüstünde bir Klasör oluşturalım
- Klasörü VSCode içine taşıyalım
- Terminal açalım ve npm init basit ayarlamaları yapıp Json dosyasını oluşturacağız
- dependencies ve devDependencies kısımlarını oluşturalım. Buraya ihtiyaç halinde kullanabileceğimiz bazı kütüphaneleri koyacağız.



Hello World

- day01 isimli bir klasör oluşturalım.
- Klasörün içine P01.js dosyasını oluşturalım
- Terminal açıp cd day01 ile klasör içine girelim.
- node P01.js yazıp çalıştıralım

```
1 // Hello World yazalım
2 console.log("Hello World")
3 console.log(" ")
4 console.log(" ")
5
6 // Variable oluşturalım
7 var firstName="Murat"
8 let lastName="Yiğit"
9 var age,dateOfBirth,
10 age=43
11 dateOfBirth=1980
12 const occupation="Instructor"
13 console.log("First Name:",firstName,
14 "\nLast Name:",lastName,
15 "\nAge:",age,
16 "\nDate Of Birth:",dateOfBirth,
17 "\nOccupation:",occupation)
18 console.log("")
19 //bazi variable değerlerini değiştirelim
20 age=44
21 occupation="Teacher"
22 console.log("First Name:",firstName,
23 "\nLast Name:",lastName,
24 "\nAge:",age,
25 "\nDate Of Birth:",dateOfBirth,
26 "\nOccupation:",occupation)
27 console.log("")
28
29
```

Variables

Değişken tanımlamak için var, let, veya const anahtar kelimeleri kullanılır.

- **var:** Genişletilmiş scope, eski ve global scope değişken tanımlama için kullanılır.
- **let:** Block scope, modern ve değişebilir (mutable) değişken tanımlama için kullanılır.
- **const:** Block scope, modern ve değişmez (immutable) değişken tanımlama için kullanılır.

```
1 // Hello World yazalım
2 console.log("Hello World")
3 console.log(" ")
4 console.log(" ")
5
6 // Variable oluşturalım
7 var firstName="Murat"
8 let lastName="Yiğit"
9 var age,dateOfBirth,
10 age=43
11 dateOfBirth=1980
12 const occupation="Instructor"
13 console.log("First Name:",firstName,
14 "\nLast Name:",lastName,
15 "\nAge:",age,
16 "\nDate Of Birth:",dateOfBirth,
17 "\nOccupation:",occupation)
18 console.log("")
19 //bazı variable değerlerini değiştirelim
20 age=44
21 occupation="Teacher"
22 console.log("First Name:",firstName,
23 "\nLast Name:",lastName,
24 "\nAge:",age,
25 "\nDate Of Birth:",dateOfBirth,
26 "\nOccupation:",occupation)
27 console.log("")
28
29
```

Data Types

- 5 Primitive Data Types.
 - `var firstname="Murat"`
 - `var age=44`
 - `var isHeRetired=false`
 - `var dateOfRetiring=null`
 - `var numberOfRetired=undefined`

```
// Data Types
// JavaScriptte String, Number, Boolean,
// Null ve undefined olmak üzere 5 primitive data tipi vardır

var middleName='Berk'
//String

var dateOfDay=25
//Number '25' yazarsak matematiksel işlemlerde kullanılamaz

var isHeRetired=false
//Boolean false ise değil true ise öyle anlamı verir

var yearsOfRetiring=null
//herhangi bir değer atamadan kullanılır

var numberOfRetired=undefined
//Teknik bir data tanımsız olduğu için
//kullanılmaması gerektiğini ifade eder.
```

Concatenation & Interpolation

String birleştirme (concatenation) ve string içine değer eklemek için iki yaygın yöntem vardır: concatenation ve interpolation.

String interpolation, string içinde değişken veya ifadeleri doğrudan eklemeyi sağlar. Modern JavaScript (ES6 ve sonrası) template literals (``) kullanarak string interpolation sağlar.

concatenation ve interpolation'un aynı mesajı nasıl oluşturduğunu görebilirsiniz. Interpolation kullanmak, daha temiz ve okunabilir kod sağlar. Bu nedenle, özellikle modern JavaScript projelerinde template literals ve interpolation sıkça tercih edilir.



Objects

JavaScript'te bir nesne (object), birden fazla değer içeren yapılardır. Nesneler özelliklerin (property) ve metotların (method) birleşimini içeren bir veri yapısıdır.

Nesneler, bir anahtar-değer çiftleri koleksiyonunu temsil eder.

```
var person = {  
  firstName: "John",  
  lastName: "Doe",  
  age: 30,  
  isStudent: false,  
};
```

Arrays

JavaScript'te bir dizi (array), bir dizi veriyi saklamak için kullanılan bir veri yapısıdır. Diziler, sıralı bir liste oluşturmak ve bu listedeki her öğeye bir indeks aracılığıyla erişmek için kullanılır.

```
var fruits = ["Elma", "Armut", "Muz", "Çilek"];
```

Operatörler

1. Aritmetik Operatörler
2. Atama Operatörleri
3. Mantıksal Operatörler

Conditional Statements

1. If
2. If-Else
3. If-Else-If
4. Ternary

Functions

1. Fonksiyon Tanımlama ve Çağırma
2. Fonksiyon Parametreleri ve Döndürülen Değerler
3. Anonim Fonksiyon İfadesi
4. Arrow (Ok) Fonksiyonları (ES6 ve Sonrası)
5. Fonksiyon Scope: