



جلسه اول

- آشنایی با مفاهیم وب
- آشنایی با مفاهیم پایه‌ای طراحی وب
- شروع کار با HTML

آشنایی با مفاهیم وب

وظایف وب دولوپر چیه؟

ما وظیفه‌مون اینه که صفحات اپلیکیشن‌های تحت وب رو توسعه بدیم، نگهداری کنیم و در صورت نیاز ارتقاء بدیم.

توسعه دادن اپلیکیشن‌های وب دو بخش مجزا داره:

- فرانت‌اند
- بک‌اند

آشنایی با مفاهیم وب

تو توسعه فرانت‌اند باید چیکار کنیم؟

- استفاده از HTML برای ساختار دادن به صفحات
- استفاده از CSS برای طراحی، ریسپانسیو کردن و زیباسازی صفحات
- پیاده‌سازی منطق برای بخش‌های مختلف صفحه با استفاده از JavaScript
- بهینه‌سازی صفحه برای دسترسی بهتر و در نهایت SEO

آشنایی با مفاهیم وب

تو توسعه بک‌اند باید چیکار کنیم؟

- توسعه منطق سمت سرور و اتصال اپلیکیشن به دیتابیس
- انتخاب فریمورک مناسب برای پیاده‌سازی بک‌اند
- اتصال وبسایت به API‌ها و سرویس‌های شخص ثالث
- پیاده‌سازی مواردی مثل امنیت، ذخیره‌سازی داده، تایید هویت و ...

آشنایی با مفاهیم وب

از چه ابزارهایی برای پیاده‌سازی وب استفاده می‌شود؟

HTML



CSS



JavaScript

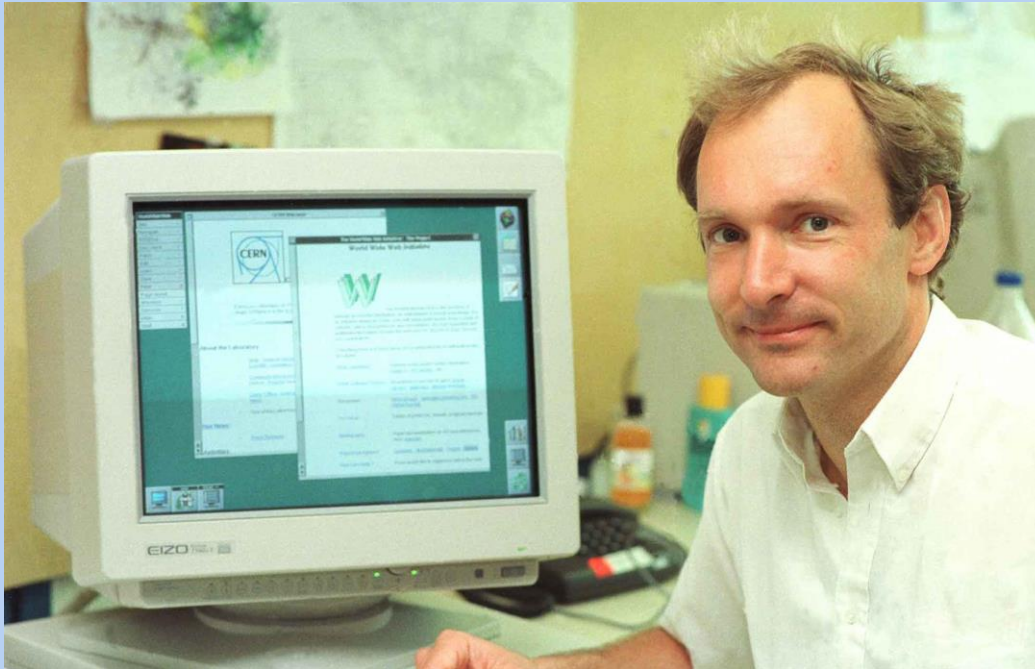


SQL



آشنایی با مفاهیم وب

کمی تاریخ: مخترع وب، آقای تیم برنرز-لی



تیم برنرز لی، سازنده وب



مارک آندریسن: سازنده موزائیک و
نتاسکیپ اولین مرورگر گرافیکی
دنیا

وب چطوری کار می‌کنه؟

اینترنت، وب، وبسایت‌ها چطوری کار می‌کنن؟

اینترنت

وب (www)

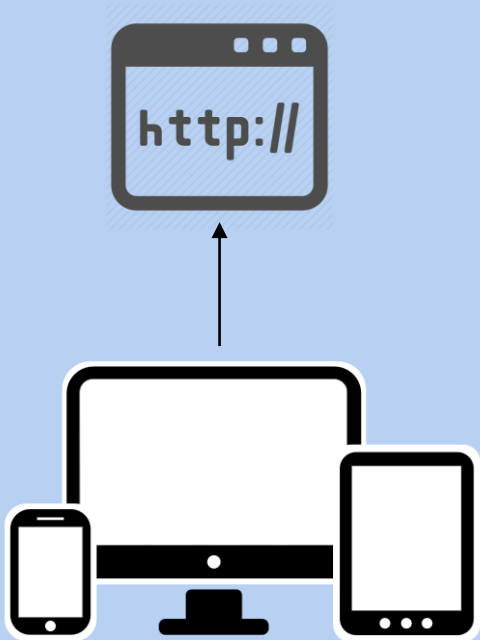
نرم‌افزاری که توی اینترنت کار می‌کنه

زیرساخت / سخت‌افزار / شبکه‌ها

وب چطوری کار می‌کنه؟

اینترنت، وب، وبسایت‌ها چطوری کار می‌کنن؟

- فرض کنید می‌خوایم یه وبسایت مثل گوگل رو باز کنیم به محض این که آدرس رو تو مرورگر می‌نویسیم اتفاقات زیر می‌افته:
 - آدرس دامنه یا URL ورودی بررسی میشه
 - یه درخواست به سرور گوگل فرستاده میشه
 - جواب دریافتی از سرور تجزیه و تحلیل یا Parse میشه
 - صفحه برای ما رندر میشه
 - اینجا google.com دامنه یا domain نه



وب چطوری کار می‌کنه؟

اینترنت، وب، وبسایت‌ها چطوری کار می‌کنن؟

- کارمندای گوگل، گوگل رو ساختن

- گوگل تشکیل شده از چندین و چند تا فایل که هرکدوم وظیف خودش رو داره
- کدهایی که کارمندای گوگل زدن، جایی تو کامپیوترهای خودشون تو آمریکاست
- این فایل‌ها همون HTML و CSS و JavaScriptهایی هستن که کارمندای گوگل ساختن



HTML



CSS

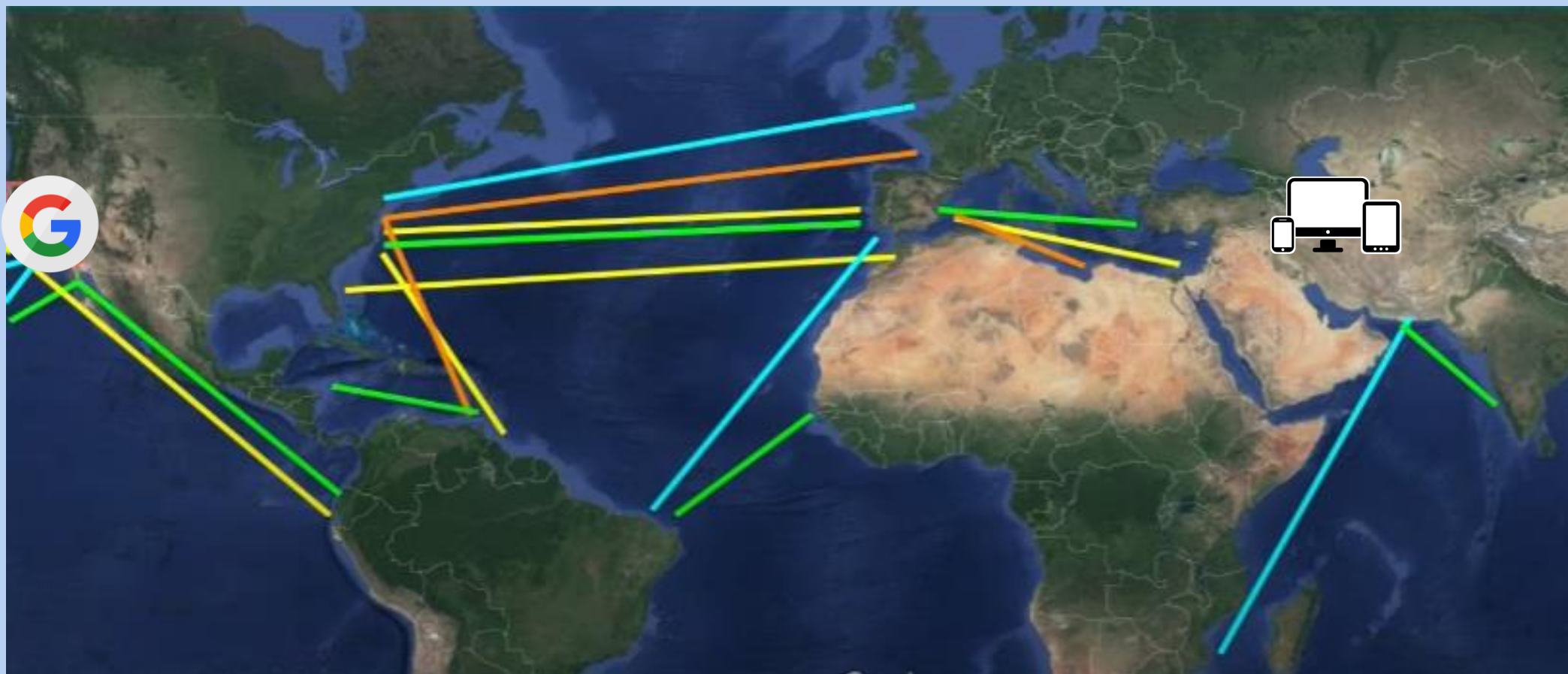


JavaScript



وب چطوری کار می‌کنه؟

این فایل‌ها چطوری به ما می‌رسن؟



وب چطوری کار می‌کنه؟

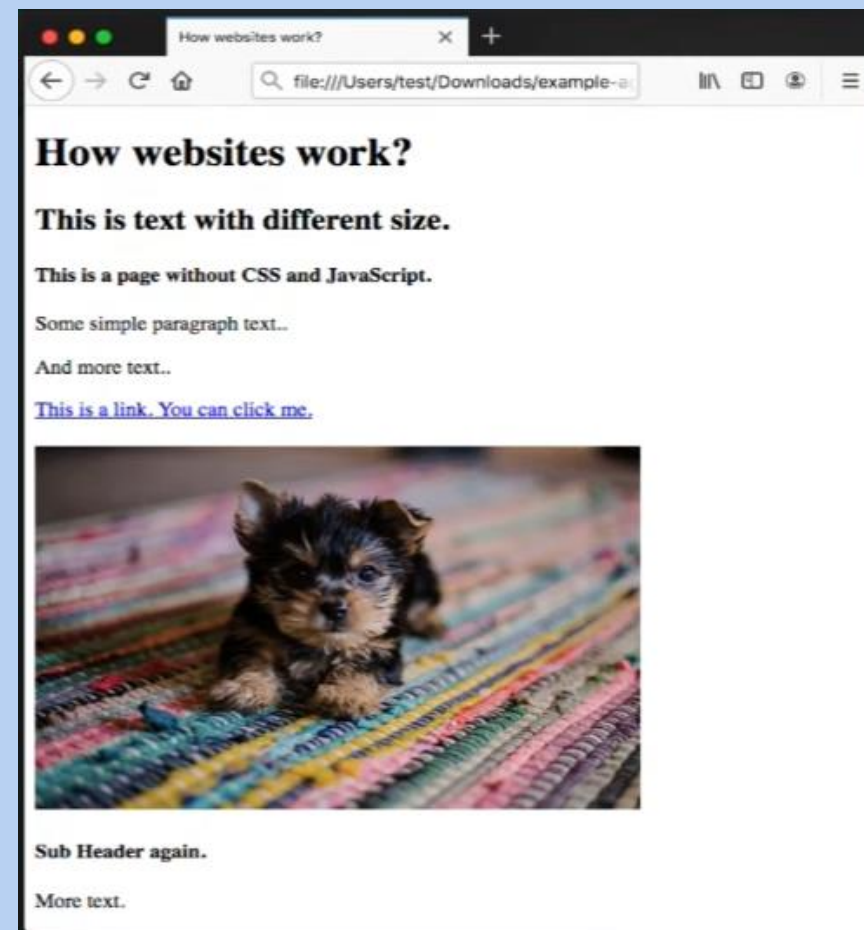
اینترنت، وب، وبسایت‌ها چطوری کار می‌کنن؟



وب چطوری کار می‌کنه؟

هرکدوم از فایلایی که دریافت کردیم چه کاری انجام میدن؟

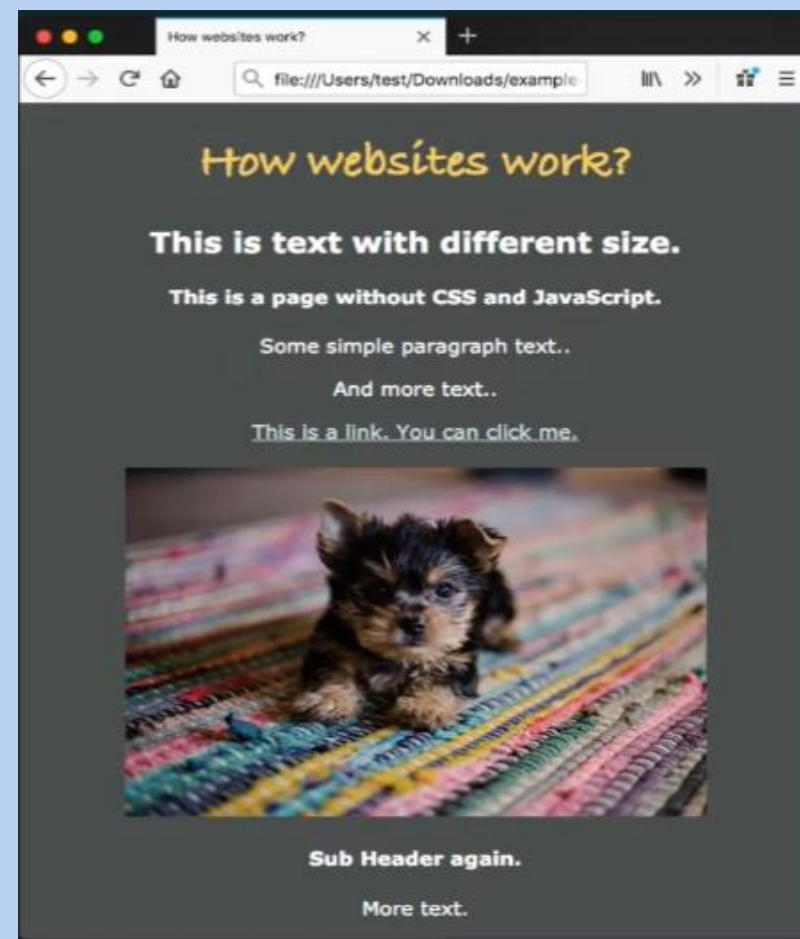
HTML



وب چطوری کار می‌کنه؟

هرکدوم از فایلایی که دریافت کردیم چه کاری انجام میدن؟

CSS



وب چطوری کار می‌کنه؟

ولی با HTML و CSS هیچ منطق و عملکردی توی سایت‌ها نداریم!

JavaScript



- جاوا اسکریپت عملکردها رو به صفحات وبسایت اضافه می‌کنه
 - مثل آپلود
 - کامنت گذاشتن
 - باز کردن یه پاپ‌آپ
 - ارسال یه پیام
 - لایک کردن
 - و ...

وب چطوری کار می‌کنه؟

ولی با HTML و CSS هیچ منطق و عملکردی توی سایت‌ها نداریم!

JavaScript



- پس برای همه عملکردها که به UI یا User Interface یا رابط کاربری مربوط میشه نیاز به استفاده از JavaScript داریم

وب چطوری کار می‌کنه؟

ولی بخشی از فایلای مربوط به وبسایت از طریق اینترنت روی کامپیوتر ما دانلود نمیشه!

- وقتی به عکس رو آپلود می‌کنی، نیاز داریم تا اون تغییرات رو بقیه هم ببینن

- برای این کار، نیاز داریم تا تغییراتمون دائمی باشه

- با HTML و CSS و JavaScript انجام این کار غیر ممکنه

- پس اینجاست که نیاز داریم اطلاعاتمون رو برای سمت دیگه‌ای از وب بفرستیم: بک‌اند

- این فایل‌ها روی سرورها ذخیره میشن و وظیفه‌شون ارتباط با دیتابیس



وب چطوری کار می‌کنه؟

داده‌ها و دیتابیس چی؟



- داده‌ها و اطلاعات تو یه سری کامپیوتر تو دیتاسنترها ذخیره میشه
- اکثر شرکت‌های بزرگ، یه ساختمان پر از کامپیوتر دارن که به هرکدوم میگن سرور
- این کامپیوترها باید در تمام مدت آنلاین باشن تا ارتباط ما با اون وبسایت قطع نشه



تو این دوره ما با چیا کار داریم؟

HTML



CSS



JavaScript



حالا موروگرا چطوری کار می‌کنن؟



Headings

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

```
<h1>Heading 1</h1>
```

```
<h2>Heading 2</h2>
```

```
<h3>Heading 3</h3>
```

```
<h4>Heading 4</h4>
```

```
<h5>Heading 5</h5>
```

```
<h6>Heading 6</h6>
```

HTML Attributes

```
<a href="https://google.com">Visit Google</a>



<p style="color:red;">This is a red paragraph.</p>
<!DOCTYPE html>
<html lang="en">
<body>
...
</body>
</html>
<p title="I'm a tooltip">This is a paragraph.</p>
```

HTML Paragraphs

```
<h1>This is heading 1</h1>
<p>This is some text.</p>
<hr>
<h2>This is heading 2</h2>
<p>This is some other text.</p>
<hr>
```

This is heading 1

This is some text.

This is heading 2

This is some other text.

```
<p>This is<br>a paragraph<br>with line breaks.</p>
```

```
<pre>
  My Bonnie lies over the ocean.

  My Bonnie lies over the sea.

  My Bonnie lies over the ocean.

  Oh, bring back my Bonnie to me.
</pre>
```

```
My Bonnie lies over the ocean.
My Bonnie lies over the sea.
My Bonnie lies over the ocean.
Oh, bring back my Bonnie to me.
```

HTML Lists

```
<ul>  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ul>
```

```
<ol>  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>
```


Styles

```
<tagname style="property:value;">
```

```
<body style="background-color:powderblue;">
```

```
<h1 style="background-color:powderblue;">This is a heading</h1>
```

```
<h1 style="color:blue;">This is a heading</h1>
```

```
<p style="color:red;">This is a paragraph.</p>
```

```
<h1 style="font-family:verdana;">This is a heading</h1>
```

```
<p style="font-family:courier;">This is a paragraph.</p>
```

```
<h1 style="font-size:300%;">This is a heading</h1>
```

```
<p style="font-size:160%;">This is a paragraph.</p>
```

```
<h1 style="text-align:center;">Centered Heading</h1>
```

```
<p style="text-align:center;">Centered paragraph.</p>
```

HTML Comments

```
<!-- This is a comment -->
```

```
<p>This is a paragraph.</p>
```

```
<!-- Remember to add more information here -->
```

HTML Formatting Elements

``This text is bold``

``This text is important!``

`<i>`This text is italic`</i>`

``This text is emphasized``

`<small>`This is some smaller text.`</small>`

`<p>`Do not forget to buy `<mark>`milk`</mark>` today.`</p>`

`<p>`My favorite color is ``blue`` red.`</p>`

`<p>`My favorite color is ``blue`` `<ins>`red`</ins>`.`</p>`

`<p>`This is `_{`subscripted`}` text.`</p>`

`<p>`This is `^{`superscripted`}` text.`</p>`

HTML Quotations

```
<p>Here is a quote from WWF's website:</p>
```

```
<blockquote cite="http://www.worldwildlife.org/who/index.html">
```

```
For 60 years, WWF has worked to help people and nature thrive.  
As the world's leading conservation organization, WWF works in  
nearly 100 countries. At every level, we collaborate with  
people around the world to develop and deliver innovative  
solutions that protect communities, wildlife, and the places in  
which they live.
```

```
</blockquote>
```

HTML Colors

```
<h1 style="color:Tomato;">Hello World</h1>  
<p style="color:DodgerBlue;">Lorem ipsum...</p>  
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

```
<h1 style="border:2px solid Tomato;">Hello World</h1>  
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>  
<h1 style="border:2px solid Violet;">Hello World</h1>
```

نحوه رنگ‌دهی در HTML

`rgb(255, 99, 71)`

`#ff6347`

`hsl(9, 100%, 64%)`

`rgba(255, 99, 71, 0.5)`

`hsla(9, 100%, 64%, 0.5)`

RGB

`rgb(255, 0, 0)`

`rgb(0, 0, 255)`

`rgba(0, 255, 0, 1)`

HEX

A solid red rectangular color swatch with a thin white border.

#ff0000

A solid green rectangular color swatch with a thin white border.

#00ff00

A solid blue rectangular color swatch with a thin white border.

#0000ff

HSL

`hsl(0, 100%, 50%)`

`hsl(240, 100%, 50%)`

`hsl(147, 50%, 47%)`

`hsl(300, 76%, 72%)`

`hsl(39, 100%, 50%)`

`hsl(248, 53%, 58%)`

CSS

CSS چیه؟

- CSS مخفف کلمه Cascading Style Sheets هست که به فارسی بهش میگن شیوه‌نامه آبشاری
- CSS مشخص می‌کنه که هرکدوم از المان‌های HTML با چه طراحی ظاهری خاصی باید نمایش داده بشن
- با استفاده از CSS این امکان رو داریم تا چندین و چند صفحه رو به طور همزمان از نظر ظاهری مدیریت کنیم
- به سه روش می‌تونیم از CSS استفاده کنیم:
 - به صورت inline
 - به صورت internal
 - به صورت external

آشنایی با سینتکس CSS

- اصطلاحات به هر فایل CSS می‌گیم Stylesheet (استایل‌شیت) به معنی برگه طراحی
- برای این که بتونیم یک المان از HTML رو طراحی کنیم، باید اول اون المان رو انتخاب کنیم
- برای این کار، باید قبل از هرچیز، اسم اون تگ رو بنویسیم
- سینتکس اعمال حالت ظاهری توی CSS به این شکله:

• Selector: به معنای انتخاب‌گر. تو این بخش، اسم

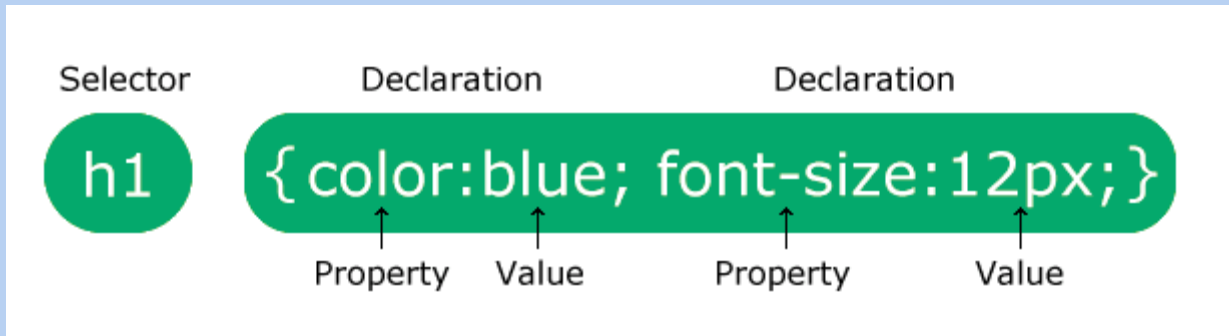
المان مد نظرمون رو می‌نویسیم. مثلا اینجا تگ `h1`

انتخاب شده.

- باید حتما از { و } برای شروع و پایان حالت دادن استفاده کنیم. اصطلاحا به هر بخش که با این دو

علامت شروع میشه، می‌گیم بلاک

- چیزایی که داخل آکولادها نوشته میشن، نشون دهنده اطلاعات ظاهری تگ انتخابیمون هستن.



آشنایی با سینتکس CSS

```
p {  
  color: red;  
  text-align: center;  
}
```

آشنایی با سلکتورهای CSS

- انتخاب کردن المان‌های HTML به سه روش انجام می‌شود
- از طریق انتخاب المنت که اصطلاحاً به آن Element Selector می‌گوییم
- انتخاب از طریق تگ id، که به آن id Selector می‌گوییم
- انتخاب از طریق نام کلاس یا رده، که به آن Class Selector می‌گوییم

CSS Element Selector

```
p {  
  text-align: center;  
  color: red;  
}
```

CSS id Selector

```
#para1 {  
    text-align: center;  
    color: red;  
}
```

```
<p id="para1">This paragraph refers to two classes.</p>
```


CSS Class Selector

```
.center {  
    text-align: center;  
    color: red;  
}
```

```
<p class="center large">This paragraph refers to two classes.</p>
```

CSS Class Selector

```
p.center {  
    text-align: center;  
    color: red;  
}
```

```
<p class="center large">This paragraph refers to two classes.</p>
```

CSS Class Selector

```
<p class="center large">This paragraph refers to two classes.</p>
```

CSS Universal Selector

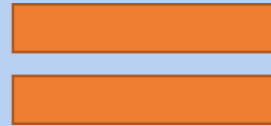
```
* {  
  text-align: center;  
  color: blue;  
}
```

CSS Grouping Selector

```
h1 {  
  text-align: center;  
  color: red;  
}
```

```
h2 {  
  text-align: center;  
  color: red;  
}
```

```
p {  
  text-align: center;  
  color: red;  
}
```



```
h1, h2, p {  
  text-align: center;  
  color: red;  
}
```

همه سلکتورهای ساده در CSS

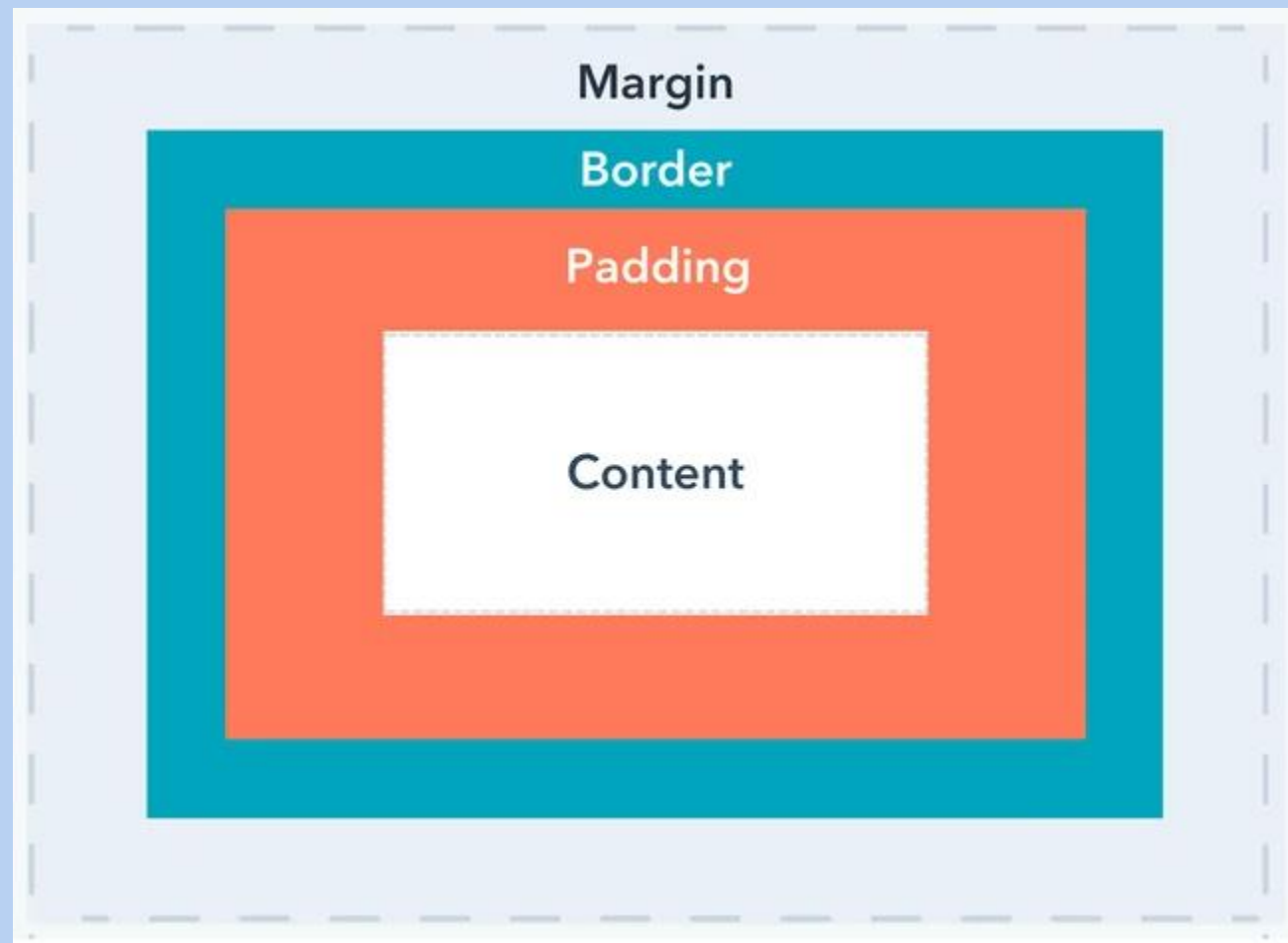
توضیح	مثال	سلکتور
انتخاب یک المنت در HTML با id="firstname"	#firstname	#id
انتخاب تمام المنت‌ها در HTML با class="firstname"	.firstname	.class
تنها انتخاب p‌هایی که دارای class="intro" هستند	p.intro	element.class
انتخاب همه المنت‌ها	*	*
انتخاب همه المان‌های div و p و h1	div, p, h1	element, element, ...

آشنایی با المنت div

- عمدتاً از div در HTML به عنوان نگهدارنده (یا اصطلاحاً یک ظرف و به انگلیسی Container) استفاده می‌شود.
- المنت div هیچ اتریبیوت الزامی ندارد و صرفاً می‌تواند در صورت نیاز اتریبیوت‌های style، id و class را دریافت کند.
- با ادغام المنت div و CSS می‌توان طراحی مد نظر برای بخش‌های مختلف وبسایت را روی آن انجام داد.

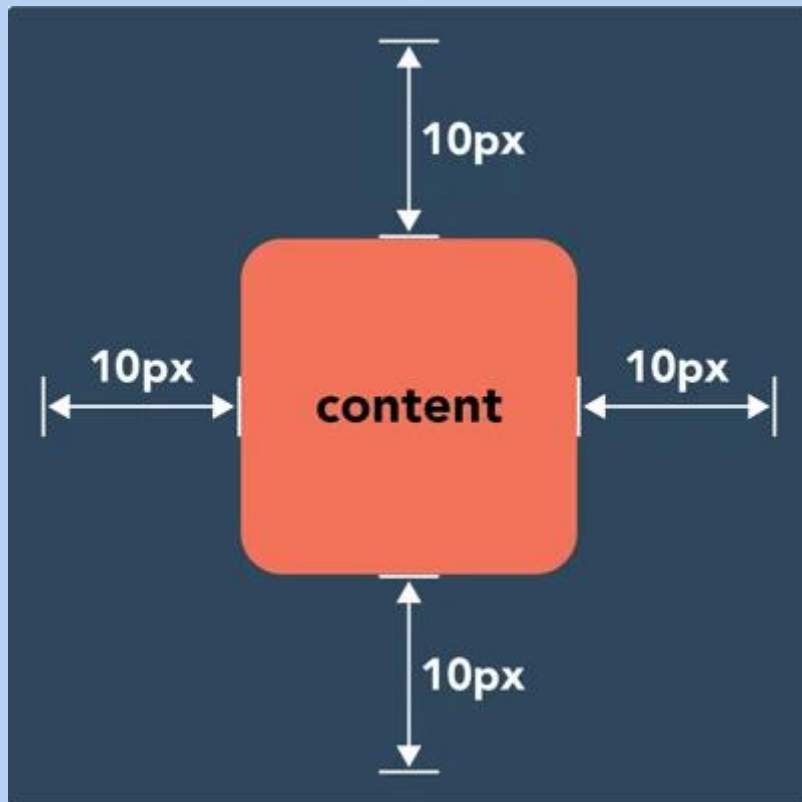
```
<div style="background-color:black;color:white;padding:20px;">  
  <h2>London</h2>  
  <p>London is the capital city of England. It is the most populous city in the  
United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>  
</div>
```

آشنایی با بُردر، مارجین و پدینگ

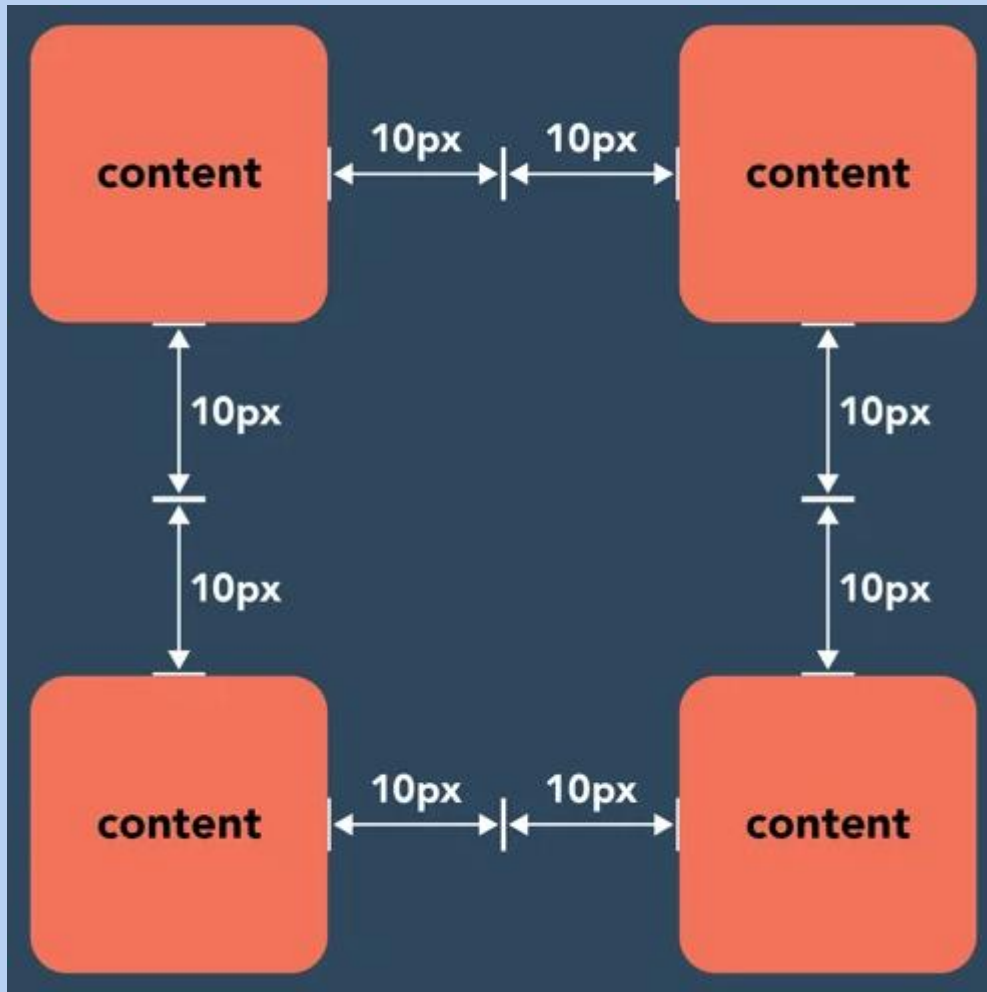


آشنایی با مارجین یا Margin

- مارجین، حاشیه اطراف یه المنت رو مشخص می‌کنه
- وقتی به یه المنت مارجین می‌دیم، به همون اندازه المنت‌ها در همسایگیش رو از خودش دور می‌کنه



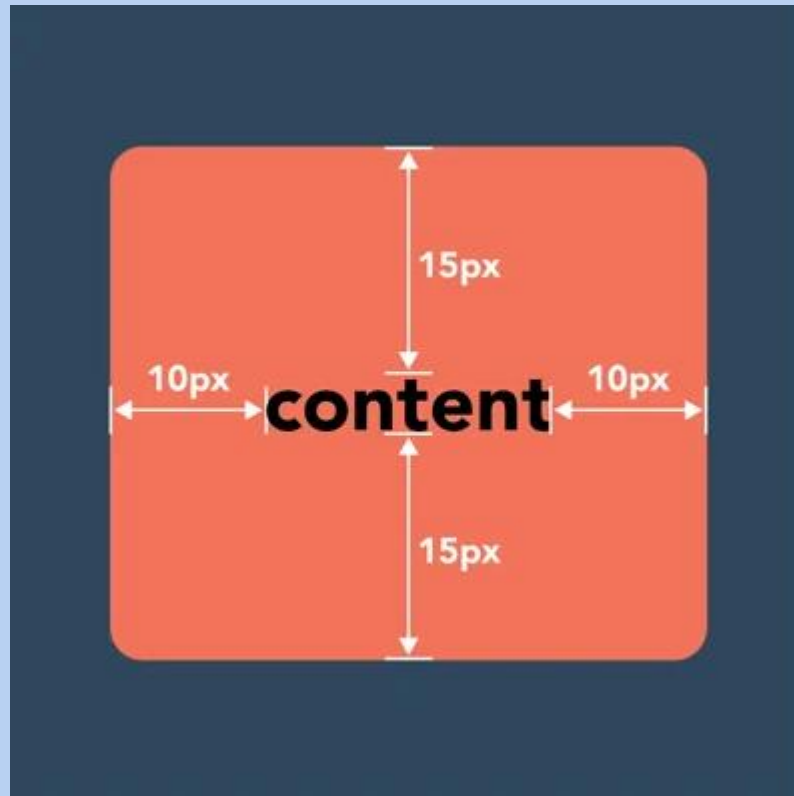
آشنایی با مارجین یا Margin



- مارجین می‌تونه از اطراف به یک المان وارد بشه
- چپ، راست، بالا و پایین

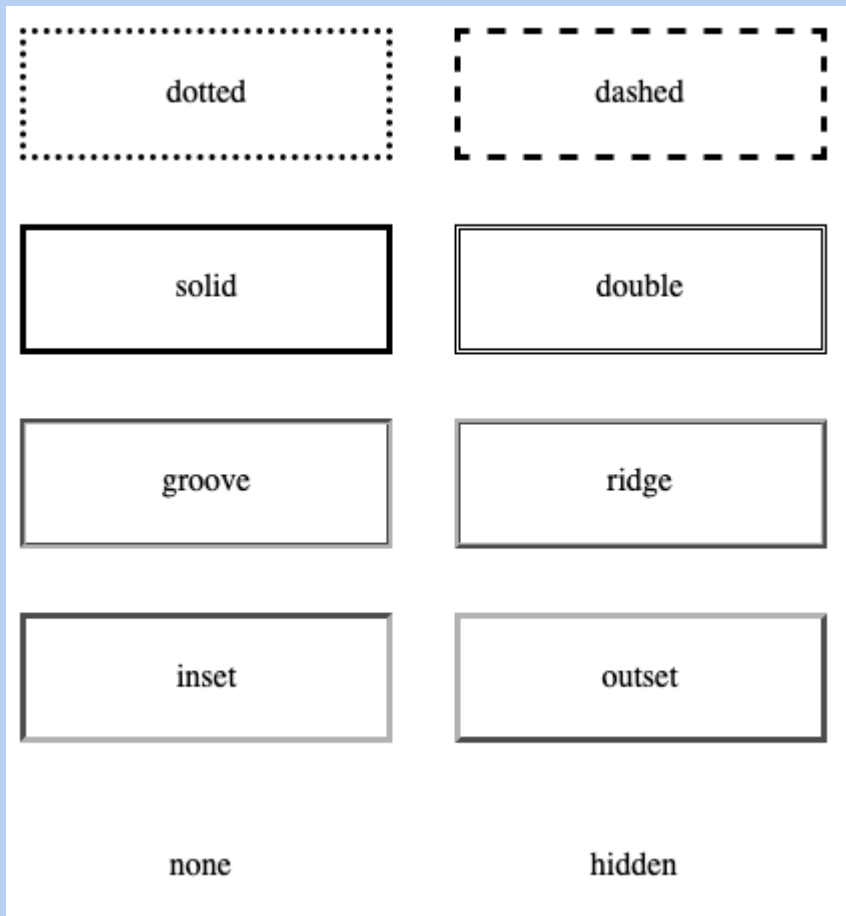
آشنایی با پدینگ یا Padding

- پدینگ برعکس مارجین، به حاشیه بیرونی المنت کاری نداره، بلکه حاشیه داخلی المنت رو تعیین می‌کنه
- یعنی، یه فاصله از مرزهای خودش تا محتوای داخلش ایجاد می‌کنه



آشنایی با بُردر یا Border

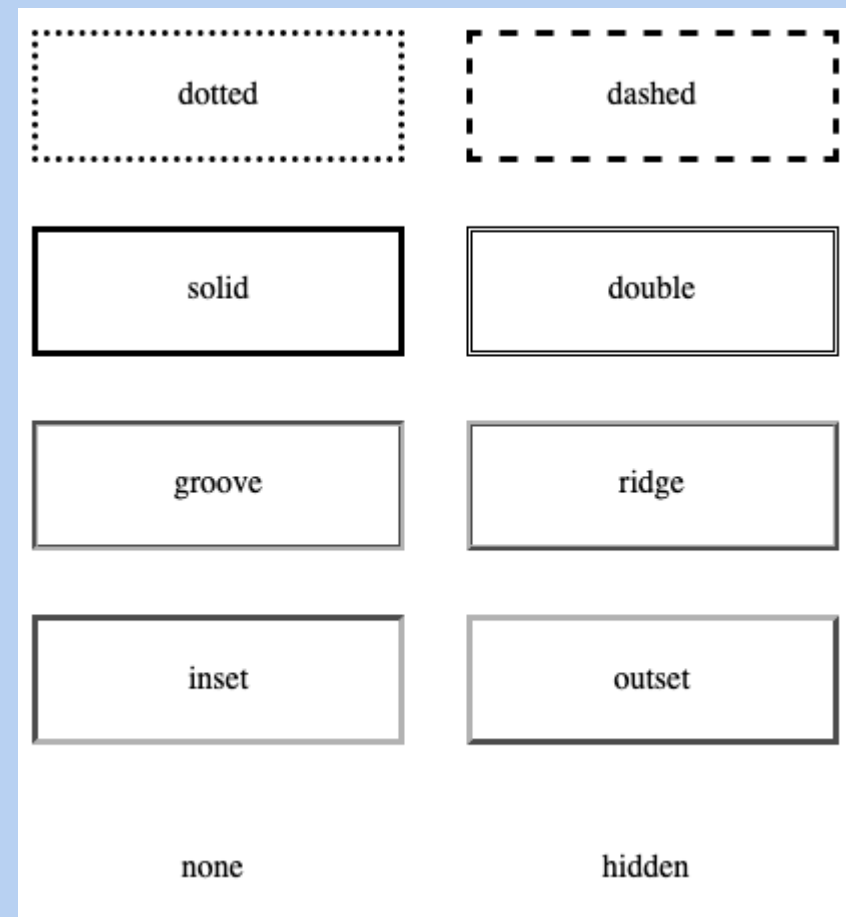
- به مرز بین المنت و فضای بیرون می‌گیم بُردر



بُردر یا Border در CSS

- برای اعمال بُردر روی یه المنت HTML باید به صورت زیر عمل کنیم:

```
p.dotted {border-style: dotted;}  
p.dashed {border-style: dashed;}  
p.solid {border-style: solid;}  
p.double {border-style: double;}  
p.groove {border-style: groove;}  
p.ridge {border-style: ridge;}  
p.inset {border-style: inset;}  
p.outset {border-style: outset;}  
p.none {border-style: none;}  
p.hidden {border-style: hidden;}  
p.mix {border-style: dotted dashed solid  
double;}
```



بُردر یا Border در CSS

```
p.one {  
  border-style: solid;  
  border-width: 5px;  
}
```

```
p.two {  
  border-style: solid;  
  border-width: medium;  
}
```

```
p.three {  
  border-style: dotted;  
  border-width: 2px;  
}
```

```
p.four {  
  border-style: dotted;  
  border-width: thick;  
}
```

- می‌تونیم با استفاده از border-width ضخامت بُردرها رو مشخص کنیم:

بُردر یا Border در CSS

- می‌تونیم با استفاده از `border-direction-style` حالت هر سمت از بردر یه المنت رو مشخص کنیم:

```
p {  
  border-top-style: dotted;  
  border-right-style: solid;  
  border-bottom-style: dotted;  
  border-left-style: solid;  
}
```

Different Border Styles

بُردر یا Border در CSS

- می‌تونیم با استفاده از border-radius بُردرهای اعمال شده به المنت‌ها مون رو خمیده کنیم:

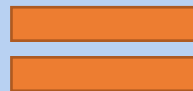
```
p {  
  border: 2px solid red;  
  border-radius: 5px;  
}
```

Round border

مارجین یا Margin در CSS

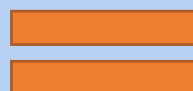
- برای حاشیه دادن به اطراف یه المنت از مارجین یا Margin استفاده می‌کنید

```
p {  
  margin-top: 100px;  
  margin-bottom: 100px;  
  margin-right: 150px;  
  margin-left: 80px;  
}
```



```
p {  
  margin: 100px 150px 100px 80px  
}
```

```
p {  
  margin-top: 100px;  
  margin-bottom: 150px;  
  margin-right: 150px;  
  margin-left: 80px;  
}
```

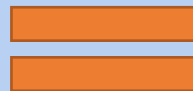


```
p {  
  margin: 100px 150px 80px  
}
```

مارجین یا Margin در CSS

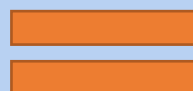
- برای حاشیه دادن به اطراف یه المنت از مارجین یا Margin استفاده می‌کنید

```
p {  
  margin-top: 100px;  
  margin-bottom: 100px;  
  margin-right: 150px;  
  margin-left: 150px;  
}
```



```
p {  
  margin: 100px 150px;  
}
```

```
p {  
  margin-top: 100px;  
  margin-bottom: 100px;  
  margin-right: 100px;  
  margin-left: 100px;  
}
```

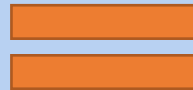


```
p {  
  margin: 100px;  
}
```

پدینگ یا Padding در CSS

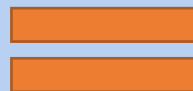
- برای حاشیه دادن به اطراف یه المنت از مارجین یا Margin استفاده می‌کنید

```
p {  
  padding-top: 100px;  
  padding-bottom: 100px;  
  padding-right: 150px;  
  padding-left: 150px;  
}
```



```
p {  
  padding: 100px 150px;  
}
```

```
p {  
  padding-top: 100px;  
  padding-bottom: 100px;  
  padding-right: 100px;  
  padding-left: 100px;  
}
```

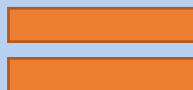


```
p {  
  padding: 100px;  
}
```

پدینگ یا Padding در CSS

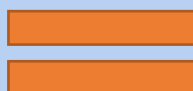
- برای حاشیه دادن به داخل یه المنت از Padding استفاده می‌کنیم

```
p {  
  padding-top: 100px;  
  padding-bottom: 100px;  
  padding-right: 150px;  
  padding-left: 80px;  
}
```



```
p {  
  padding: 100px 150px 100px 80px  
}
```

```
p {  
  padding-top: 100px;  
  padding-bottom: 150px;  
  padding-right: 150px;  
  padding-left: 80px;  
}
```



```
p {  
  padding: 100px 150px 80px  
}
```

امروز فقط با سلک‌تورها کار داریم

آشنایی با Table

```
<table>
  <thead>
    <tr>
      <th>Company</th>
      <th>Contact</th>
      <th>Country</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Alfreds Futterkiste</td>
      <td>Maria Anders</td>
      <td>Germany</td>
    </tr>
    <tr>
      <td>Centro comercial Moctezuma</td>
      <td>Francisco Chang</td>
      <td>Mexico</td>
    </tr>
  </tbody>
</table>
```

- برای ساختن جدول از تگ <table> استفاده می‌کنیم
- نمایش هر سطر از جدول، از طریق تگ <tr> مشخص می‌شود
- هر سلول در جدول، با استفاده از تگ <td> مشخص می‌شود
- برای مشخص کردن عناوین جدول، باید به جای <tr> از <thead> استفاده کنیم و به جای <td> از <th>

استایل کردن حاشیه و خطوط سلول‌های Table

- ما می‌تونیم با استفاده از CSS، جدولی که ساختیم رو تغییر بدیم و اصطلاحاً استایلش کنیم
- برای اضافه کردن خطوط به اطراف و بین سلول‌های table می‌تونیم از این نوع استایل دادن استفاده کنیم:

```
table, th, td {  
  border: 1px solid black;  
}
```


استایل کردن حاشیه و خطوط سلول‌های Table

- برای جلوگیری از این که جدول ایجاد شده، دو تا خط داشته باشه، می‌تونیم از پراپرتی border-collapse استفاده کنیم و اون رو برابر با collapse قرار بدیم. با این کار، خط دو لایه جدول به حالت تک لایه تغییر پیدا می‌کنه

```
table, th, td {  
  border: 1px solid black;  
  border-collapse: collapse;  
}
```


استایل کردن حاشیه و خطوط سلول‌های Table

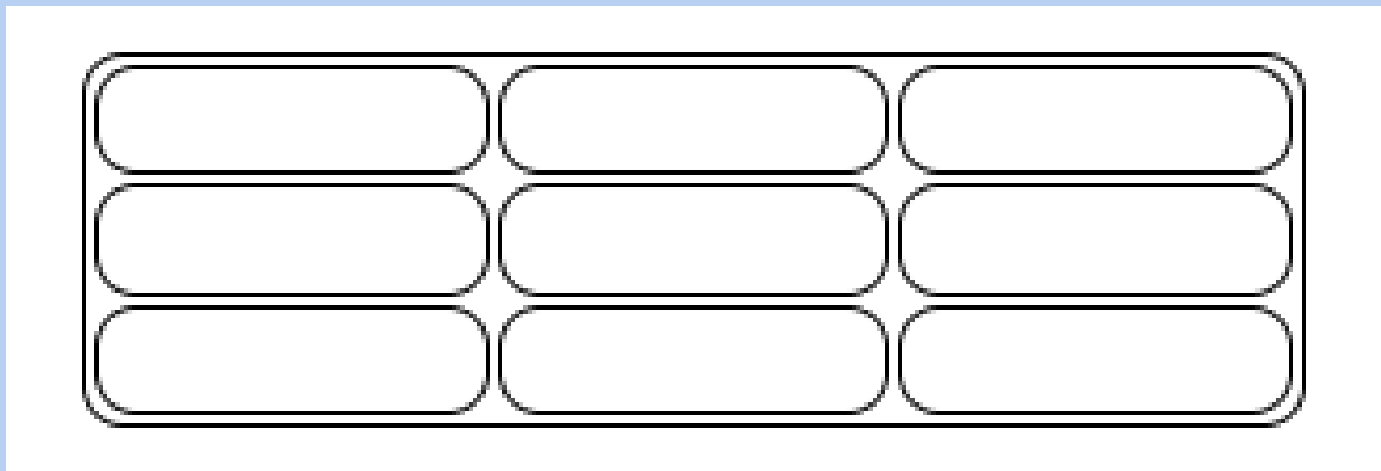
- کارای زیادی همیشه با جدول کرد و شکل‌های متفاوتی رو بهش داد، مثلا می‌تونیم رنگش رو تغییر بدیم:

```
table, th, td {  
  border: 1px solid white;  
  border-collapse: collapse;  
}  
th, td {  
  background-color: #96D4D4;  
}
```


استایل کردن حاشیه و خطوط سلول‌های Table

- یا حتی می‌تونیم حاشیه‌های گرد براش ایجاد کنیم

```
table, th, td {  
  border: 1px solid black;  
  border-radius: 10px;  
}
```



استایل کردن حاشیه و خطوط سلول‌های Table

- یا خطوط بین سلول‌ها رو رنگی کنیم:

```
th, td {  
  border-color: #96D4D4;  
}
```


آشنایی با Form ها

- فرم‌ها تو HTML برای جمع‌آوری داده‌های ورودی توسط کاربر استفاده می‌شن.
- عمدتاً، اطلاعات ثبت شده توی فرم HTML برای پردازش به سمت سرور ارسال میشه.
- تگ **<form>** یه فرم HTML می‌سازه و به عنوان یه ظرف برای المان‌های دیگه فرم مورد استفاده قرار می‌گیره.
- این المان‌ها شامل: input، checkbox، radio button، submit button ها و ... میتونن باشن.

<form>

- المان‌های فرم

- **</form>**

آشنایی با تگ input

- پرکاربردترین تگ بین المان‌های HTML فرم محسوب می‌شود.
- با استفاده از **<input>** ما می‌توانیم یه باکس ایجاد کنیم تا کاربر بتونه توش تایپ کنه.
- این تگ یه اتریبیوت خیلی مهم داره به اسم **type** که با استفاده ازش می‌تونیم حالات مختلفی از input رو برای المان فرمون ایجاد کنیم

نحوه نوشتن	توضیحات
<code><input type="text"></code>	یه تکست‌فیلد یک خطی ایجاد می‌کنه
<code><input type="radio"></code>	یه دکمه از نوع radio button ایجاد می‌کنه (انتخاب یکی از بین چند گزینه)
<code><input type="checkbox"></code>	یه چک‌باکس درست می‌کنه. یه باکس که با انتخابش، یه تیک توش به نمایش در میاد (معمولا برای چند انتخاب از بین تعدادی گزینه مورد استفاده قرار می‌گیره)
<code><input type="submit"></code>	یه دکمه برای تایید فرم ایجاد می‌کنه
<code><input type="button"></code>	یه دکمه ایجاد می‌کنه

آشنایی با Text Field

- با استفاده از `<input type="text">` می‌تونیم یه Text Field بسازیم.
- در واقع با این کار، یه باکس می‌سازیم که امکان تایپ کردن رو برای کاربر فراهم می‌کنیم

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

آشنایی با Label

- تگ <label> مشخص کننده یه شناسه برای المان‌های فرم
- مثل یه برچسب برای هر فرم عمل می‌کنه
- این تگ به اسکرین‌ریدرها کمک می‌کنه تا هر اینپوت رو بخونن
- همچنین کمک می‌کنه تا با کلیک روی لیبل، بتونیم اینپوت مورد نظر رو انتخاب کنیم
- این تگ یه اتریبیوت مهم داره به اسم for که وقتی مساوی با id یکی از اینپوت‌ها باشه، اونوقت مربوط به همدیگه‌ان

```
<form>  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname"><br>  
  <label for="lname">Last name:</label><br>  
  <input type="text" id="lname" name="lname">  
</form>
```

آشنایی با Radio Button

- با استفاده از `<input type="radio">` به اینپوت می‌سازیم که از نوع Radio Button
- رادیو باتن‌ها در واقع گزینه‌هایی هستند که از بین مجموعه‌ای از انتخاب‌ها، فقط امکان انتخاب یکیشون وجود دارد

```
<form>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaScript">
  <label for="javascript">JavaScript</label>
</form>
```


آشنایی با Checkboxes

- با استفاده از `<input type="checkbox">` می‌تونیم چک‌باکس داشته باشیم.
- چک‌باکس در واقع انتخاب‌هایی مثل رادیو باتن‌ها هستن، با این تفاوت که به کاربر اجازه انتخاب چند گزینه رو می‌ده.

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label>
</form>
```

آشنایی با Submit Button

- با استفاده از `<input type="submit">` می‌تونیم یه دکمه بسازیم
- البته برای ساخت دکمه می‌تونیم از تگ `<button type="submit">` هم استفاده کنیم
- برای ارسال اطلاعات داخل فرم به سرور، از این دکمه استفاده میشه

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

آشنایی با اتریبیوت Name در اینپوت‌ها

- احتمالا تا الان متوجه اتریبیوت name شده باشید، این اتریبیوت لزوما باید برای همه اینپوت‌ها انتخاب بشه
- این مورد نشون دهنده اسم داده‌ایه که از سمت اینپوت قراره به سرور ارسال بشه
- اسم‌گذاری name نباید از اسپیس یا خط فاصله استفاده کنیم
- اگه از این اتریبیوت برای یه اینپوت داخل فرم‌ها استفاده نکنیم، بعد از submit هیچ داده‌ای به سرور ارسال نمیشه

```
<form action="/action_page.php">  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" value="John"><br><br>  
  <input type="submit" value="Submit">  
</form>
```

آشنایی با تگ meta در HTML

- تگ متا به سری اطلاعات راجع به پرونده HTML تعریف می‌کند که اصطلاحاً به این اطلاعات می‌گویند metadata.
- تگ meta همیشه باید داخل تگ <head> باشد و عمدتاً برای مشخص کردن اطلاعاتی از پرونده HTML از آن استفاده می‌شود. این اطلاعات می‌توانند شامل viewport، نویسنده پرونده، کلمات کلیدی و به توصیف از پرونده باشد
- متادیتا هرگز توی صفحه نمایش داده نمیشه اما ماشین‌ها می‌تونن اون رو تحلیل کنن و مورد پردازش قرار بدن
- متادیتا توسط مرورگرها مورد استفاده قرار می‌گیره (این که چطوری به صفحه رو نمایش بدن و مثلاً چطوری ریلودش کنن)، یا توسط موتورهای جستجو (برای پیدا کردن کلمات کلیدی).

آشنایی با تگ meta در HTML

Attribute	Value	Description
<u>charset</u>	<i>character_set</i>	Specifies the character encoding for the HTML document
<u>content</u>	<i>text</i>	Specifies the value associated with the http-equiv or name attribute
<u>http-equiv</u>	content-security-policy content-type default-style refresh	Provides an HTTP header for the information/value of the content attribute
<u>name</u>	application-name author description generator keywords viewport	Specifies a name for the metadata

آشنایی با تگ meta در HTML

```
<meta name="keywords" content="HTML, CSS, JavaScript">
```

```
<meta name="description" content="Free Web tutorials for HTML and CSS">
```

```
<meta name="author" content="John Doe">
```

```
<meta http-equiv="refresh" content="30">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

آشنایی با تگ meta در HTML

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- مهمترین تگ متا که مشخص کننده محیط قابل تماشای صفحه است، مربوط به همین viewport می‌شود
- دستور بالا رو باید برای همه صفحاتتون در نظر داشته باشید
- با استفاده از این دستور، سند HTML ما به مرورگر این دستور رو میده که چطوری باید ابعاد و نزدیک‌نمایی و دورنمایی صفحه رو مشخص کنه
- با استفاده از width=device-width به مرورگر می‌گیم که عرض رو باید عرض صفحه نمایش دستگاه در نظر بگیره
- با استفاده از initial-scale=1.0 میزان نزدیک‌نمایی (زوم) صفحه به صورت دیفالت رو نشون میدیم
- تو اسلاید بعد یه تصویر بدون تگ بالا و همراه با تگ بالا رو می‌تونید ببینید

آشنایی با تگ meta در HTML



Without the viewport meta tag



With the viewport meta tag

آشنایی با Background در CSS

• ۶ تا دستور مربوط به background داریم که به ترتیب زیرن:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position
- background (خلاصه)

آشنایی با Background در CSS

- به اندازه کافی با background color آشنا شدیم، میریم سراغ background-image

```
body {  
  background-image: url("paper.gif");  
}
```

```
p {  
  background-image: url("paper.gif");  
}
```

آشنایی با Background در CSS

background-repeat •

```
body {  
    background-image: url("gradient_bg.png");  
    background-repeat: repeat-x;  
}
```

```
body {  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
}
```

```
body {  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
}
```

آشنایی با Background در CSS

background-attachment •

```
body {  
  background-image: url("img_tree.png");  
  background-repeat: no-repeat;  
  background-position: right top;  
  background-attachment: fixed;  
}
```

```
body {  
  background-image: url("img_tree.png");  
  background-repeat: no-repeat;  
  background-position: right top;  
  background-attachment: scroll;  
}
```

آشنایی با Background در CSS

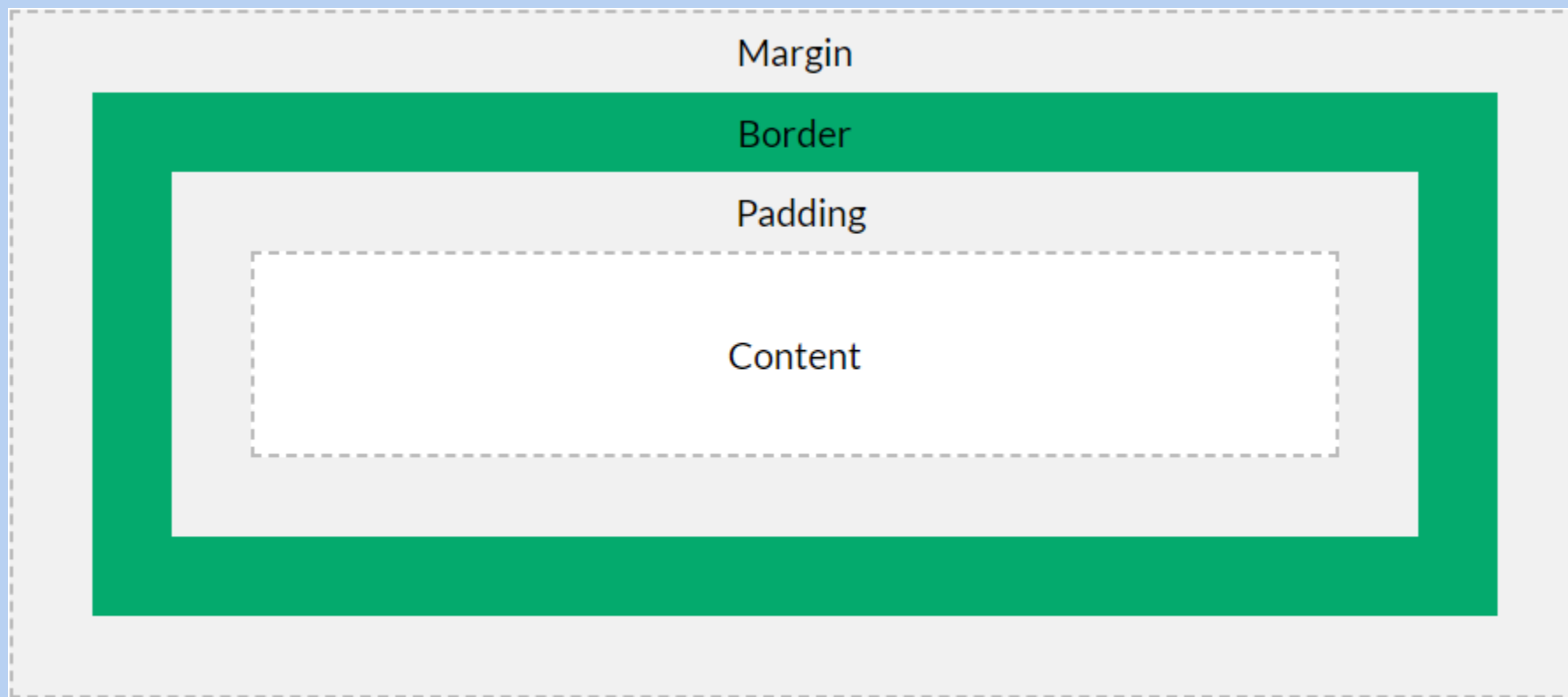
background •

```
body {  
    background-color: #ffffff;  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
}
```

```
body {  
    background: #ffffff url("img_tree.png") no-repeat right top;  
}
```

آشنایی با مفهوم Box Model در CSS

- Box Model در CSS در واقع یه مفهومه. این مفهوم مربوط به یه چارچوبه که دور هرکدوم از المانهای HTML قرار می‌گیره. که همیشه شامل: مارجین‌ها، بُدرها، پدینگ‌ها و خود محتوای داخلش



آشنایی با مفهوم Box Model در CSS

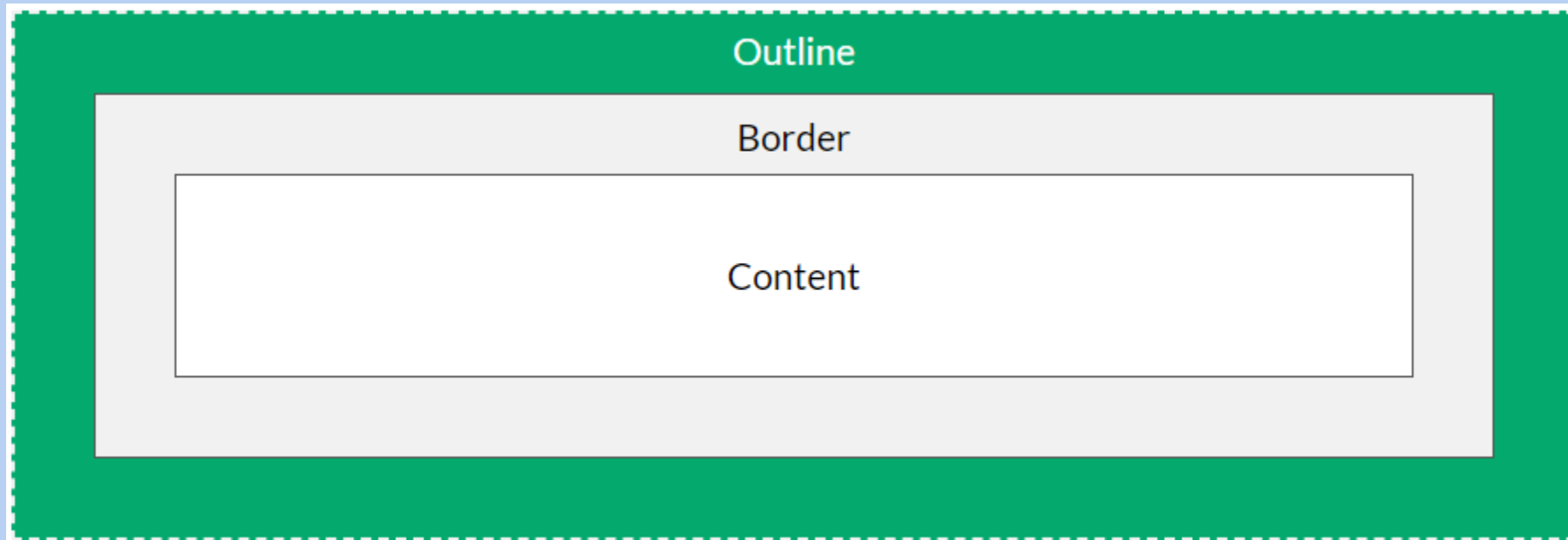
- وقتی اندازه طول و عرض المان‌های HTML رو تو CSS مشخص می‌کنیم، صرفاً اندازه مربوط به محتوای داخل باکس رو مشخص کردیم. برای به دست آوردن اندازه حقیقی کل المان، باید اندازه مارجین، پدینگ و بُدرها رو هم در نظر بگیریم.

```
div {  
  width: 320px;  
  padding: 10px;  
  border: 5px solid gray;  
  margin: 0;  
}
```

320px (width)
+ 20px (left + right padding)
+ 10px (left + right border)
+ 0px (left + right margin)
= 350px

آشنایی با مفهوم Outline در CSS

- چیزی شبیه به بُردر با این تفاوت که تو مرز بیرونی border ایجاد میشه



آشنایی با مفهوم Outline در CSS

- چیزی شبیه به بُردر با این تفاوت که تو مرز بیرونی border ایجاد میشه

```
p.ex1 {  
  border: 1px solid black;  
  outline-style: solid;  
  outline-color: red;  
  outline-width: thin;  
}
```

```
p.ex2 {  
  border: 1px solid black;  
  outline-style: solid;  
  outline-color: red;  
  outline-width: medium;  
}
```

```
p.ex3 {  
  border: 1px solid black;  
  outline-style: solid;  
  outline-color: red;  
  outline-width: thick;  
}
```

```
p.ex4 {  
  border: 1px solid black;  
  outline-style: solid;  
  outline-color: red;  
  outline-width: 4px;  
}
```

آشنایی با مفهوم Outline در CSS

- به این صورت همیشه دستور Outline رو به صورت خلاصه نوشت

```
p.ex1 {outline: dashed;}  
p.ex2 {outline: dotted red;}  
p.ex3 {outline: 5px solid yellow;}  
p.ex4 {outline: thick ridge pink;}
```

آشنایی با مفهوم Outline در CSS

- با استفاده از outline-offset می‌تونیم به فاصله بین outline و border (یا اگره بُردر نداشت، مرز المنت) ایجاد کنیم

```
p {  
  margin: 30px;  
  background: yellow;  
  border: 1px solid black;  
  outline: 1px solid red;  
  outline-offset: 15px;  
}
```

This paragraph has an outline of 15px outside the border edge.

آشنایی با مفهوم Text در CSS

- برای تغییر رنگ نوشته‌های داخل یک المان HTML از طریق CSS از المان color استفاده می‌کنیم

```
body {  
    background-color: lightgrey;  
    color: blue;  
}  
  
h1 {  
    background-color: black;  
    color: white;  
}  
  
div {  
    background-color: blue;  
    color: white;  
}
```

آشنایی با مفهوم Text در CSS

- با استفاده از text-align وضعیت موقعیت مکانی متن به المان HTML مشخص می‌شود:

```
h1 {  
  text-align: center;  
}
```

```
h2 {  
  text-align: left;  
}
```

```
h3 {  
  text-align: right;  
}
```

آشنایی با مفهوم Text در CSS

- همچنین با استفاده از خصیصه justify برای text-align، می‌تونیم نوشته‌های یه متن رو توی یه المان HTML از نظر عرضی طوری تعبیه کنیم، که همه خطوط به طور مساوی در یه راستا باشن (از نظر عرضی).

```
div {  
  text-align: justify;  
}
```

آشنایی با مفهوم Text در CSS

- با استفاده از خصیصه text-align-last می‌تونیم وضعیت مکانی افقی خط آخر یه متن رو مشخص کنیم:

```
p.a {  
  text-align-last: right;  
}
```

```
p.b {  
  text-align-last: center;  
}
```

```
p.c {  
  text-align-last: justify;  
}
```

آشنایی با مفهوم Text در CSS

- گاهی اوقات ممکنه بخوایم جهت نوشتاری متن رو مشخص کنیم. مثلا نوشته‌های فارسی و عربی از راست به چپ هستند و

```
p {  
  direction: rtl;  
}
```


آشنایی با مفهوم Text Decoration در CSS

• با استفاده از text-decoration می‌تونیم تغییراتی تو نحوه نمایش متن بدیم. خصیصه‌های این مورد عبارت‌اند از:

- text-decoration-line
- text-decoration-color
- text-decoration-style
- text-decoration-thickness
- text-decoration

text-decoration-line

- با استفاده از text-decoration-line می‌تونیم خط متن رو تغییر بدیم. موارد زیر می‌تونن با هم ترکیب شن

```
h1 {  
  text-decoration-line: overline;  
}  
  
h2 {  
  text-decoration-line: line-through;  
}  
  
h3 {  
  text-decoration-line: underline;  
}  
  
p {  
  text-decoration-line: overline underline;  
}
```

text-decoration-line

```
h1 {  
  text-decoration-line: overline;  
  text-decoration-color: red;  
}  
  
h2 {  
  text-decoration-line: line-through;  
  text-decoration-color: blue;  
}  
  
h3 {  
  text-decoration-line: underline;  
  text-decoration-color: green;  
}  
  
p {  
  text-decoration-line: overline underline;  
  text-decoration-color: purple;  
}
```

• همچنین می‌توانید رنگ خطی که ایجاد کردید رو تغییر بدید

text-decoration-line

- حتی امکان تغییر حالت خط رو هم دارید

```
h1 {  
  text-decoration-line: underline;  
  text-decoration-style: solid;  
}
```

```
h2 {  
  text-decoration-line: underline;  
  text-decoration-style: double;  
}
```

```
h3 {  
  text-decoration-line: underline;  
  text-decoration-style: dotted;  
}
```

```
p.ex1 {  
  text-decoration-line: underline;  
  text-decoration-style: dashed;  
}
```

```
p.ex2 {  
  text-decoration-line: underline;  
  text-decoration-style: wavy;  
}
```

```
p.ex3 {  
  text-decoration-line: underline;  
  text-decoration-color: red;  
  text-decoration-style: wavy;  
}
```

text-decoration-line

• و همچنین می‌تونید ضخامت خطی که ایجاد کردید رو مشخص کنید

```
h1 {  
  text-decoration-line: underline;  
  text-decoration-thickness: auto;  
}
```

```
h2 {  
  text-decoration-line: underline;  
  text-decoration-thickness: 5px;  
}
```

```
h3 {  
  text-decoration-line: underline;  
  text-decoration-thickness: 25%;  
}
```

```
p {  
  text-decoration-line: underline;  
  text-decoration-color: red;  
  text-decoration-style: double;  
  text-decoration-thickness: 5px;  
}
```

text-decoration-line

- امکان خلاصه نویسی هم وجود داره:

```
h1 {  
  text-decoration: underline;  
}  
  
h2 {  
  text-decoration: underline red;  
}  
  
h3 {  
  text-decoration: underline red double;  
}  
  
p {  
  text-decoration: underline red double 5px;  
}
```

text-decoration-line

- بعضی المان‌های HTML به صورت پیش‌فرض دارای بعضی استایل‌ها هستند. مثلاً لینک‌ها همیشه به underline دارند. برای این که underline رو از بین ببریم، می‌تونیم به این شکل عمل کنیم:

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

text-transform

- با استفاده از text-transform واسه کلمه‌های انگلیسی، این امکان رو داریم که حروف رو بزرگ یا کوچک کنیم:

```
p.uppercase {  
  text-transform: uppercase;  
}
```

```
p.lowercase {  
  text-transform: lowercase;  
}
```

```
p.capitalize {  
  text-transform: capitalize;  
}
```


آشنایی با Text Spacing در CSS

• با استفاده از المان‌های مربوط به Text Spacing می‌تونیم فاصله گذاری کلمه‌ها رو مشخص کنیم

- `text-indent`
- `letter-spacing`
- `line-height`
- `word-spacing`

text-indent

- با استفاده از text-indent حاشیه مربوط به خط اول هر پاراگراف رو مشخص کنیم

```
p {  
  text-indent: 50px;  
}
```

letter-spacing

- با استفاده از letter-spacing می‌تونیم فاصله بین حروف تو یه متن رو مشخص کنیم. تو مورد اول، فاصله زیاد و تو مورد دوم فاصله بین حروف کاهش پیدا کرده.

```
h1 {  
  letter-spacing: 5px;  
}
```

```
h2 {  
  letter-spacing: -2px;  
}
```

line-height

- با استفاده از line-height ارتفاع یه خط رو مشخص می‌کنیم

```
p.small {  
  line-height: 0.8;  
}
```

```
p.big {  
  line-height: 1.8;  
}
```

word-spacing

- با استفاده از word-spacing فاصله بین کلمات رو مشخص می‌کنیم

```
p.one {  
  word-spacing: 10px;  
}
```

```
p.two {  
  word-spacing: -2px;  
}
```

white-space

- با استفاده از white-space می‌تونیم مشخص کنیم که

```
p.one {  
  word-spacing: 10px;  
}
```

```
p.two {  
  word-spacing: -2px;  
}
```

آشنایی با فونت‌ها

- تو CSS ۵ خانواده از فونت‌ها داریم
- Serif: فونت‌هایی که مرز کوچک تو لبه‌هاشون دارن.
- Sans-serif: فونت‌هایی که هیچ مرزی ندارن و شبیه فونت‌های مدرن هستن
- Monospace: همه حروف تو این فونت‌ها عرض یکسان دارن. باعث نمایش مکانیکال میشه.
- Cursive: فونت‌هایی که دستخط انسان رو تداعی می‌کنه
- Fantasy: فونت‌های متفاوت و خاص



آشنایی با فونت‌ها

• چندتا مثال از فونت‌ها

Generic Font Family	Examples of Font Names
Serif	Times New Roman Georgia Garamond
Sans-serif	Arial Verdana Helvetica
Monospace	Courier New Lucida console Monaco
Cursive	Brush Script MT Lucida Handwriting
Fantasy	Copperplate Papyrus

آشنایی با فونت‌ها

- تو CSS ما با استفاده از font-family می‌تونیم خانواده فونت مد نظرمون رو مشخص کنیم. دقت کنید، که ما باید همیشه به فونت پشتیبان یا اصطلاحاً Fallback Font هم داشته باشیم تا در صورت پیدا نشدن فونت انتخابی، فونت پشتیبان انتخاب شه. به همین خاطر، اول اسم فونت مد نظرمون رو می‌نویسیم و بعد با انتخاب خانواده فونت انتخابیمون، مشخص می‌کنیم که در صورت پیدا نشدن فونت مد نظر، به فونت مشابه از اون خانواده رو انتخاب کنه.

```
.p1 {  
  font-family: "Times New Roman", Times, serif;  
}
```

```
.p2 {  
  font-family: Arial, Helvetica, sans-serif;  
}
```

```
.p3 {  
  font-family: "Lucida Console", "Courier New",  
  monospace;  
}
```

آشنایی با فونت‌های Web Safe

- اصطلاحاً به فونت‌های معمولی که توسط تعداد زیادی از مرورگرها پشتیبانی میشن، می‌گیم Web Safe. البته این موضوع ۱۰۰ درصدی نیست. به همین خاطر مهمه که همیشه از فونت‌های پشتیبان یا Fallback Fonts استفاده کنیم.

- Arial (sans-serif)
- Verdana (sans-serif)
- Tahoma (sans-serif)
- Trebuchet MS (sans-serif)
- Times New Roman (serif)
- Georgia (serif)
- Garamond (serif)
- Courier New (monospace)
- Brush Script MT (cursive)

آشنایی با Font Style

- با استفاده از Font Style می‌تونیم حالت فونت مد نظرمون رو مشخص کنیم. برای این کار می‌تونیم از دستور font-style استفاده کنیم

```
p.normal {  
  font-style: normal;  
}
```

```
p.italic {  
  font-style: italic;  
}
```

```
p.oblique {  
  font-style: oblique;  
}
```

آشنایی با Font Weight

- با استفاده از font-weight این امکان رو داریم که وزن یک فونت رو مشخص کنیم. وزن یه فونت مشخص می‌کنه اون فونت چقدر درشت‌تر نوشته بشه.

```
p.normal {  
  font-weight: normal;  
}
```

```
p.thick {  
  font-weight: bold;  
}
```

آشنایی با Font Size

- با استفاده از font-size می‌تونیم مشخص کنیم که اندازه فونت یه المان HTML چه اندازه‌ای باشه

```
h1 {  
  font-size: 40px;  
}
```

```
h2 {  
  font-size: 30px;  
}
```

```
p {  
  font-size: 14px;  
}
```

آشنایی با Font Size

- اندازه فونت‌ها می‌تونن ثابت یا متغیر باشن.
- اندازه ثابت یا absolute اندازه رو به یه سائز مشخص میرسونه و به کاربر اجازه تغییر اندازه فونت رو نمی‌ده. زمانی کاربرد داره که اندازه فیزیکی دستگاه خروجی به طور دقیق در اختیارمون باشه.
- اندازه نسبی یا Relative اندازه فونت رو به نسبت فضای اطرافش مشخص می‌کنه و به کاربر اجازه می‌ده در صورت نیاز و لزوم اون رو کوچک‌تر یا بزرگ‌تر کنه.

آشنایی با پیکسل

```
h1 {  
  font-size: 40px;  
}
```

```
h2 {  
  font-size: 30px;  
}
```

```
p {  
  font-size: 14px;  
}
```

آشنایی با em

- برای این که به کاربرها اجازه بدیم تا بتونن بنابر نیازشون اندازه فونت رو تغییر بدن، می‌تونیم از em استفاده کنیم. 1em مساویه با اندازه کنونی فونت. از اونجایی که اندازه پیش‌فرض فونت تو HTML برابر با ۱۶ پیکسل، اونوقت 1em برابر با 16px.

```
h1 {  
  font-size: 2.5em; /* 40px/16=2.5em */  
}  
  
h2 {  
  font-size: 1.875em; /* 30px/16=1.875em */  
}  
  
p {  
  font-size: 0.875em; /* 14px/16=0.875em */  
}
```


آشنایی با vw

- همچنین می‌تونیم از واحدی به اسم vw استفاده کنیم که مخفف viewport width به معنای عرض نمایشگره. با استفاده از vw به مرورگر می‌گیم اندازه فونت رو نسبت به عرض صفحه نمایش مشخص کن.

```
<h1 style="font-size:10vw">Hello World</h1>
```

آشنایی با Link ها در CSS

- با تگ a آشنا هستیم. حالا می‌خوایم ببینیم که چه حالت‌هایی دارد و چطوری می‌تونیم با استفاده از CSS حالت‌های مختلفش رو طراحی کنیم:

```
/* unvisited link */  
a:link {  
    color: red;  
}
```

```
/* visited link */  
a:visited {  
    color: green;  
}
```

```
/* mouse over link */  
a:hover {  
    color: hotpink;  
}
```

```
/* selected link */  
a:active {  
    color: blue;  
}
```

آشنایی با Link ها در CSS

- ما می‌تونیم با استفاده از text-decoration حالت لینک رو تغییر بدیم

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

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

```
a:hover {  
  text-decoration: underline;  
}
```

```
a:active {  
  text-decoration: underline;  
}
```

آشنایی با Link ها در CSS

- همینطور می‌تونیم با استفاده از background-color رنگ پس‌زمینه‌اش رو تغییر بدیم:

```
a:link {  
    background-color: yellow;  
}  
  
a:visited {  
    background-color: cyan;  
}  
  
a:hover {  
    background-color: lightgreen;  
}  
  
a:active {  
    background-color: hotpink;  
}
```

آشنایی با Link ها در CSS

- و با ترکیب این ساختارها می‌تونیم از تگ a یه دکمه بسازیم:

```
a:link, a:visited {  
    background-color: #f44336;  
    color: white;  
    padding: 14px 25px;  
    text-align: center;  
    text-decoration: none;  
    display: inline-block;  
}  
  
a:hover, a:active {  
    background-color: red;  
}
```

آشنایی با List ها در CSS

- حالا می‌خوایم با استفاده از CSS به مقدار ظاهر لیست‌ها رو تغییر بدیم:

```
ul.a {  
  list-style-type: circle;  
}
```

```
ul.b {  
  list-style-type: square;  
}
```

```
ol.c {  
  list-style-type: upper-roman;  
}
```

```
ol.d {  
  list-style-type: lower-alpha;  
}
```

```
ul {  
  list-style-image: url('sqpurple.gif');  
}
```

آشنایی با List ها در CSS

- با استفاده از list-style-position این امکان رو داریم که مشخص کنیم که علامت نقطه کنار المان لیست کجا قرار بگیرن

```
ul.a {  
  list-style-position: outside;  
}
```

```
ul.b {  
  list-style-position: inside;  
}
```

• Coffee - A brewed drink prepared from roasted coffee beans...
• Tea
• Coca-cola

• Coffee - A brewed drink prepared from roasted coffee beans...
• Tea
• Coca-cola

آشنایی با List ها در CSS

- همچنین میتونید حالت پیش فرض شکل لیست ها رو از بین ببرید:

```
ul {  
  list-style-type: none;  
  margin: 0;  
  padding: 0;  
}
```


آشنایی با List ها در CSS

- حالا با چیزایی که گفتیم این طرح رو بنویسید:

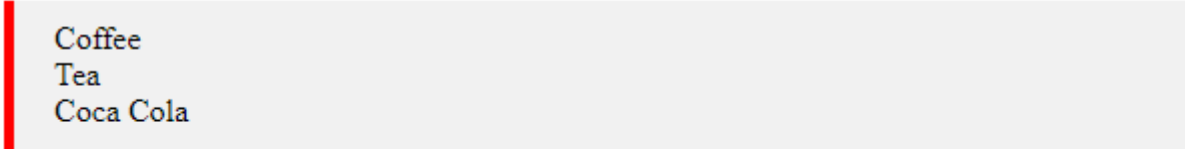
1. Coffee
2. Tea
3. Coca Cola

- Coffee
- Tea
- Coca Cola

آشنایی با List ها در CSS

- و همینطور این طرح رو:

List with a red left border

A list with a red left border. The list items are Coffee, Tea, and Coca Cola, displayed on a light gray background.

- Coffee
- Tea
- Coca Cola

A bordered list

Coffee
Tea
Coca Cola

آشنایی چیدمان‌ها و display در CSS

- صفت display پرکاربردترین صفت CSS برای مدیریت کردن چیدمان CSS به شمار می‌ره. در واقع display مشخص می‌کنه که یه المان چطوری نمایش داده بشه و یا حتی اصلا نمایش داده بشه یا نه؟
- هر المان HTML به صورت پیش‌فرض یه مقدار به عنوان نحوه نمایش داره. بیشتر المان‌ها مقدار صفت display یی که دارن یا block هست و یا inline

المان‌های سطح Block

- المان‌هایی که صفت display شوند برابر با block هست، از خط جدید شروع و تمام عرض قابل بهره‌برداری رو از آن خودش می‌کنه (در واقع تا انتهای سمت راست – و یا تو حالت راست‌چین تا انتهای سمت چپ- رو مال خودش می‌کنه).

The `<div>` element is a block-level element.

- `<div>`
- `<h1>` - `<h6>`
- `<p>`
- `<form>`
- `<header>`
- `<footer>`
- `<section>`

المان‌های inline

- المان‌هایی که به صورت پیش‌فرض از خط جدید شروع نمیشن و کل عرض قابل بهره‌برداری رو مال خودشون نمی‌کنن.

This is an inline element inside a paragraph.

- ``
- `<a>`
- ``

آشنایی با `display: none`

- با استفاده از `display:none` برای پنهان کردن، می‌توانیم اوان‌ها را مخفی کنیم. معمولاً با استفاده از جاوا اسکریپت، این مقدار به یکی از اوان‌ها پاس داده می‌شود تا اوان‌ها مخفی شوند.

تغییر حالت پیش فرض لیست با استفاده از display

- ما می‌تونیم مقدار display پیش‌فرض هرکدام از المان‌ها رو تغییر بدیم. برای این کار به راحتی این امکان رو داریم که با استفاده از CSS مقدارش رو عوض کنیم. به عنوان مثال، می‌دونیم که تگ `li` به صورت پیش‌فرض مقدار `block` داره. حالا می‌خوایم این مقدار رو `inline` کنیم تا دیگه تمام عرض ممکن رو پوشش نده:

```
li {  
  display: inline;  
}
```

آشنایی با max-width

- تو بخشای قبلی به این مسئله اشاره کردیم که المان‌های سطح بلاک، همیشه تمام عرض رو پوشش می‌دن. ما می‌تونیم با استفاده از width یه عرض ثابت بهش بدیم و با استفاده از margin: auto اون المان رو وسطچین کنیم.
- اما مشکل وقتی پیش میاد که عرض صفحه نمایش از عرض مشخص شده ما کوچیکتر باشه. اونوقت عرض المان همچنان ثابت می‌مونه و مرورگر یه اسکرول‌بار افقی به صفحه اضافه می‌کنه.
- برای حل این مشکل، میتونیم از max-width استفاده کنیم. با استفاده از max-width، یه حداکثر عرض مشخص می‌کنیم و المانمون هرگز از اون مقدار بزرگتر نمیشه اما تو صفحه نمایش‌های کوچیکتر، عرض المان با عرض صفحه نمایش همگام‌سازی میشه.

```
div.ex1 {  
  width: 500px;  
  margin: auto;  
  border: 3px solid #73AD21;  
}
```

```
div.ex2 {  
  max-width: 500px;  
  margin: auto;  
  border: 3px solid #73AD21;  
}
```