

- مقدمه
- برنامه‌نویسی وب
- HTML (زبان نشانه‌گذاری ابرمتنی)
- DOM (Document Object Model)
- عناصر HTML بیشتر
- فرم‌ها
- CSS (Cascading Style Sheets)
- طراحی واکنش‌گرا
- Bootstrap
- Sass (Syntactically Awesome Style Sheets)

مقدمه

این دوره ادامه‌ی CS50 است که طی آن به طراحی و ایجاد وب اپلیکیشن‌ها خواهیم پرداخت. در طول این دوره مهارت‌های طراحی وب را با کار بر روی چندین پروژه تقویت خواهیم کرد، نهایتاً یک پروژه‌ی نهایی پایان باز خواهیم داشت که طی آن فرصت طراحی یک وب‌سایت برای خودتان را خواهید داشت. در این دوره، نیاز به یک ویرایشگر متن (text editor) دارید تا از طریق آن کدها را به طور محلی (local) و بر روی کامپیوتر خود بنویسید. برخی از ویرایشگرهای متنی متداول عبارتند از: [Visual Studio Code](#)، [Sublime Text](#)، [Atom](#)، و [Vim](#)، البته موارد بسیار زیاد دیگری نیز وجود دارند که از میان آنها می‌توانید یکی را به دلخواه انتخاب کنید.

برنامه‌نویسی وب

موضوعات دوره:

- گرچه بعداً جزئیات بیشتری را ارائه خواهیم کرد، اما در اینجا یک مرور کلی به موضوعاتی خواهیم داشت که در طول این دوره مورد بحث قرار خواهند گرفت:
1. HTML و CSS (یک زبان نشانه‌گذاری برای طراحی کلی صفحه‌ی وب، و رویه‌ای که برای افزایش جذابیت بصری سایت‌ها مورد استفاده قرار می‌گیرد)
 2. Git (برای کنترل ورژن و همکاری)
 3. Python (زبان برنامه‌نویسی بسیار متداولی که برای پویا تر (dynamic) کردن وب‌سایت خود از آن استفاده خواهیم کرد)

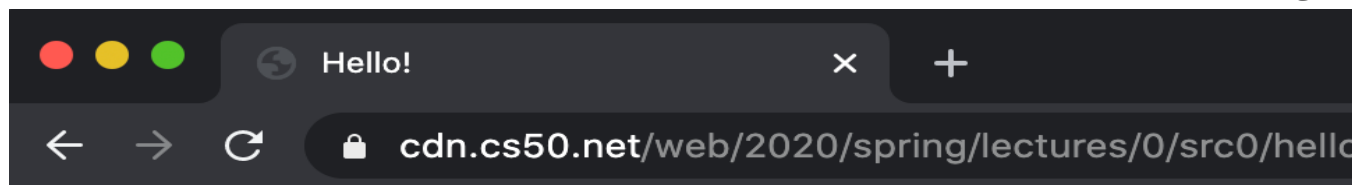
4. Django (یک فریم‌ورک رایج وب که از آن برای سمت سرور (backend) استفاده خواهیم کرد)
5. Migrations, Models, SQL (زبانی که برای ذخیره‌سازی و بازیابی داده از آن استفاده می‌شود، و روش‌های مختص Django که برقراری تعامل با پایگاه‌داده‌های SQL را تسهیل می‌نماید)
6. JavaScript (زبان برنامه‌نویسی که برای سریع‌تر نمودن و تعاملی‌تر ساختن وب‌سایت‌ها مورد استفاده قرار می‌گیرند)
7. رابط‌های کاربر (User Interfaces) (روش‌هایی برای ساده‌ترسازی هرچه بیشتر یک وب‌سایت)
8. CD, CI, Testing (یادگیری روش‌های مختلف، که به منظور حصول اطمینان از کارکرد صحیح بروزرسانی‌های صفحات وب مورد استفاده قرار می‌گیرد)
9. مقیاس‌پذیری و امنیت (حصول اطمینان از این امر که تعداد بسیاری از کاربران به طور همزمان می‌توانند به وب‌سایت ما دسترسی داشته باشند و در عین حال از مقاصد مخرب نیز در امان بمانند)

Hypertext Markup Language (HTML)

- HTML یک زبان نشانه‌گذاری است که ساختار یک صفحه‌ی وب را تعریف می‌کند. جهت نمایش محتوا بر روی صفحه‌ی نمایش، کدهای HTML از طریق مرورگر شما (Safari, Google Chrome, Firefox, ...) تفسیر می‌شوند.
- بیایید کار را با نوشتن یک فایل ساده‌ی HTML آغاز کنیم!

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Hello!</title>
  </head>
  <body>
    Hello, world!
  </body>
</html>
```

- زمانی که این فایل را در مرورگر خود باز کنید، نتیجه به این صورت خواهد بود:



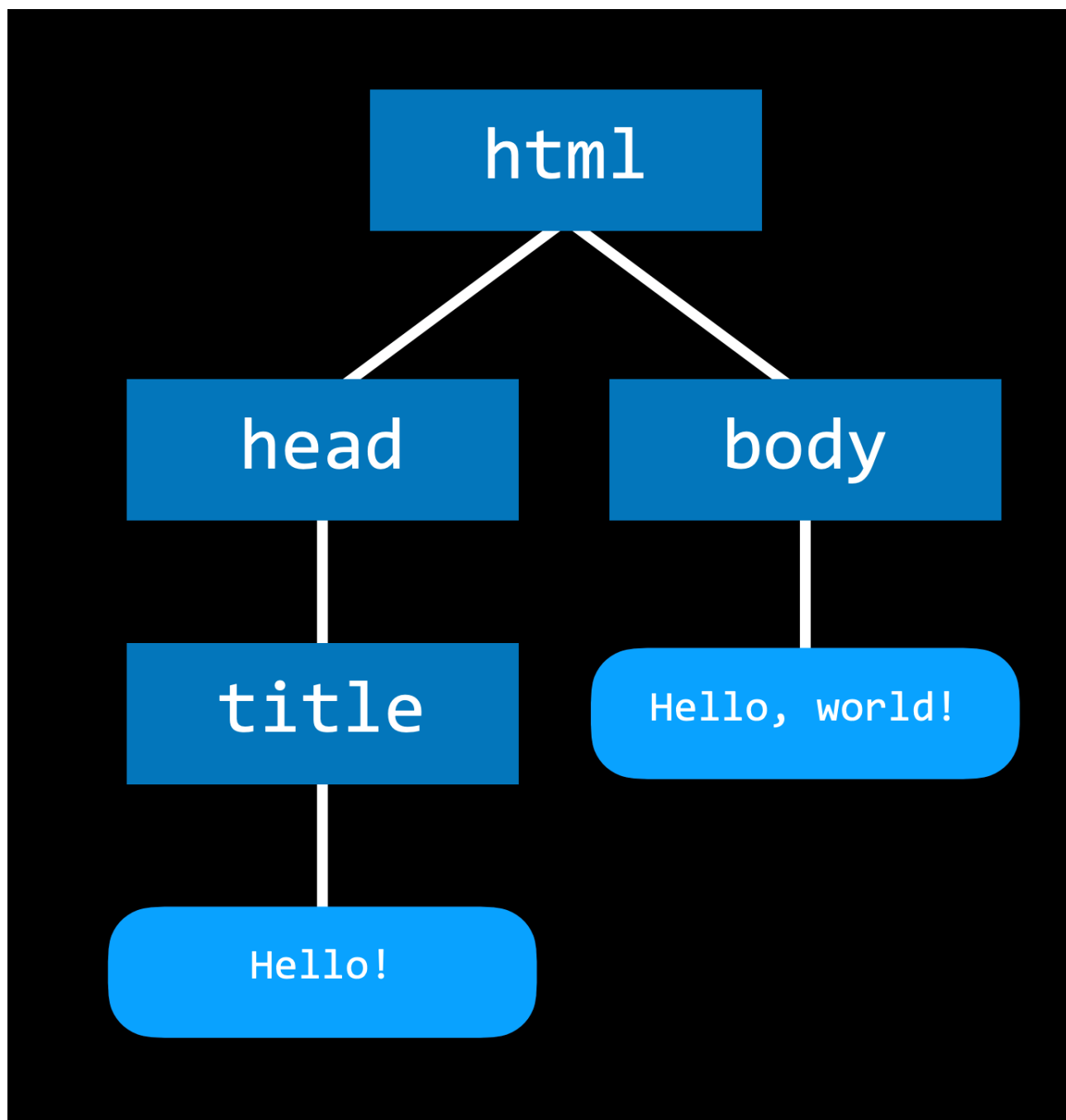
Hello, world!

- اکنون بیاید در مورد فایلی که نوشته‌ایم کمی بحث کنیم، برای صفحه‌وبی به این سادگی، چنین کدی بسیار پیچیده به نظر می‌آید.
- در سطر اول، (به مرورگر وب) اعلام می‌کنیم که فایل را با استفاده از نسخه‌ی نهایی HTML یعنی HTML5 نوشته‌ایم.
- بعد از آن، صفحه از چندین تگ HTML تودرتو (نظیر html و body) تشکیل شده است، که هر کدام دارای یک تگ باز و بسته بوده و با استفاده از <element> در آغاز و </element> در پایان المان، نشانه‌گذاری شده است.
- توجه داشته باشید که هر المان داخلی نسبت به المان قبلی خود دارای مقداری تورفتگی است. استفاده از این الگو از سوی مرورگر الزامی نمی‌باشد، اما تبعیت از آن برای تان سودمند خواهد بود.
- المان‌های HTML می‌توانند حاوی برخی مشخصه‌ها (attributes) باشند، این مشخصه‌ها برخی اطلاعات اضافی در رابطه با المان را، در اختیار مرورگر قرار می‌دهند. برای نمونه، با نوشتن `en="lang"` در تگ ابتدایی، در واقع به مرورگر اعلام می‌کنیم که زبان پایه‌ی استفاده شده توسط ما زبان انگلیسی خواهد بود.
- در درون تگ HTML، عموماً از دو تگ head و body استفاده می‌کنیم.

تگ `head`، حاوی اطلاعاتی درباره‌ی صفحه‌ی وب است که اساساً نمایش داده نمی‌شود، در مقابل محتوای تگ `body`، برای کاربران استفاده کننده از وب‌سایت قابل مشاهده خواهد بود.

- در درون تگ `head` یک `title` برای صفحه‌ی وب خود انتخاب کرده‌ایم، و همانطور که مشاهده می‌کنید این عنوان در تب موجود در بالای مرورگر وب نمایش داده شده است.
- نهایتاً یک متن `"Hello , World"` را در تگ `body` قرار داده‌ایم، که بخش قابل نمایش صفحه‌ی ما را شکل می‌دهد.

(Document Object Model) DOM



- DOM یک روش ساده برای بصری‌سازی نحوه‌ی ارتباط‌های HTML با یکدیگر و با استفاده از یک ساختار درختی است. شکل بالا طرح کلی DOM صفحه وبی را به نمایش می‌گذارد که به تازگی ایجاد کرده‌ایم.

المان‌های بیشتر از HTML

- المان‌های HTML بسیاری نظیر `heading`، `lists`، و بخش‌های `bold` شده وجود دارند که برای شخصی‌سازی صفحه وب خودتان می‌توانید از آنها استفاده کنید.
- یک نکته‌ی دیگر: `<!-- -->` امکان درج کامنت در HTML را فراهم می‌آورد که از آن برای درج توضیحات در زیر هر المان استفاده خواهیم کرد.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>HTML Elements</title>
  </head>
  <body>
    <!-- We can create headings using h1 through h6 as tags. -->
    <h1>A Large Heading</h1>
    <h2>A Smaller Heading</h2>
    <h6>The Smallest Heading</h6>

    <!-- The strong and i tags give us bold and italics respectively. -->
    A <strong>bold</strong> word and an <i>italicized</i> word!

    <!-- We can link to another page (such as cs50's page) using a. -->
    View the <a href="https://cs50.harvard.edu/">CS50 Website</a>!

    <!-- We used ul for an unordered list and ol for an ordered one. both ordered and unordered lists contain li, or list
    items. -->
    An unordered list:
    <ul>
      <li>foo</li>
      <li>bar</li>
      <li>baz</li>
    </ul>
    An ordered list:
    <ol>
      <li>foo</li>
      <li>bar</li>
      <li>baz</li>
    </ol>

    <!-- Images require a src attribute, which can be either the path to a file on your computer or the link to an image
    online. It also includes an alt attribute, which gives a description in case the image can't be loaded. -->
    An image:
    
    <!-- We can also see above that for some elements that don't contain other ones, closing tags are not necessary. -->

    <!-- Here, we use a br tag to add white space to the page. -->
    <br/> <br/>

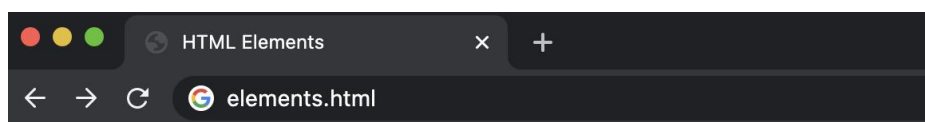
    <!-- A few different tags are necessary to create a table. -->
    <table>
      <thead>
        <th>Ocean</th>
        <th>Average Depth</th>
        <th>Maximum Depth</th>
```

```

</thead>
<tbody>
  <tr>
    <td>Pacific</td>
    <td>4280 m</td>
    <td>10911 m</td>
  </tr>
  <tr>
    <td>Atlantic</td>
    <td>3646 m</td>
    <td>8486 m</td>
  </tr>
</tbody>
</table>
</body>
<html>

```

این صفحه در زمان اجرا، شبیه به تصویر زیر خواهد بود:



A Large Heading

A Smaller Heading

The Smallest Heading

A **bold** word and an *italicized* word! View the [CS50 Website](#)! An unordered list:

- foo
- bar
- baz

An ordered list:

1. foo
2. bar
3. baz



An image:

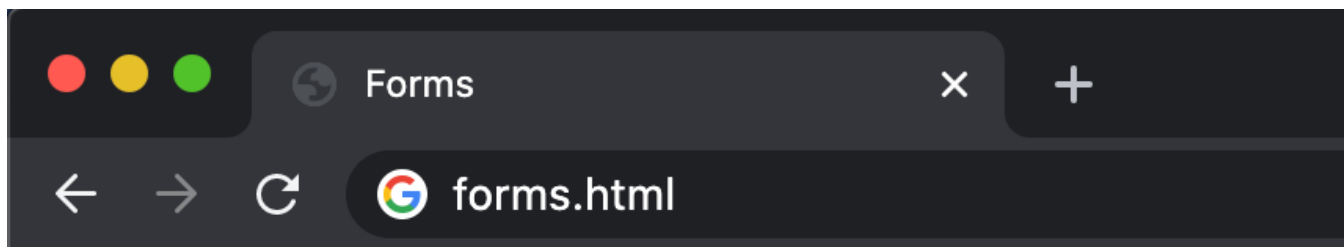
| Ocean | Average Depth | Maximum Depth |
|----------|---------------|---------------|
| Pacific | 4280 m | 10911 m |
| Atlantic | 3646 m | 8486 m |

- چنانچه درباره‌ی چگونگی به خاطر سپاری این تگ ها نگران هستید، باید به این نکته اشاره کنم که نیازی به نگرانی نیست، به سادگی می‌توانید با انجام یک جستجوی ساده نظیر "درج تصویر در HTML"، تگ `img` را پیدا کنید. یک منبع بسیار مفید در رابطه با این المان‌ها [W3Schools](#) می‌باشد.

Forms فرم‌ها

- مجموعه‌ی دیگری از المان‌ها که در زمان ایجاد وبسایت بسیار اهمیت دارند، نحوه‌ی گردآوری اطلاعات از کاربران می‌باشد. می‌توانید این امکان را برای کاربران فراهم بیاورید که با استفاده از یک فرم HTML اطلاعاتی را وارد نمایند، این اطلاعات می‌تواند حاوی انواع مختلفی از ورودی‌ها باشد. بعداً در همین دوره، نحوه‌ی مدیریت این اطلاعات بعد از ارسال فرم را خواهیم آموخت.
- مانند المان‌های دیگر HTML، به خاطر سپاری این المان نیز مورد نیاز نبوده، و `W3School` یک منبع فوق‌العاده جهت یادگیری این المان‌ها خواهد بود.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Forms</title>
</head>
<body>
  <form>
    <input type="text" placeholder="First Name" name="first">
    <input type="password" placeholder="Password" name="password">
    <div>
      Favorite Color:
      <input name="color" type="radio" value="blue"> Blue
      <input name="color" type="radio" value="green"> Green
      <input name="color" type="radio" value="yellow"> Yellow
      <input name="color" type="radio" value="red"> Red
    </div>
    <input type="submit">
  </form>
</body>
</html>
```



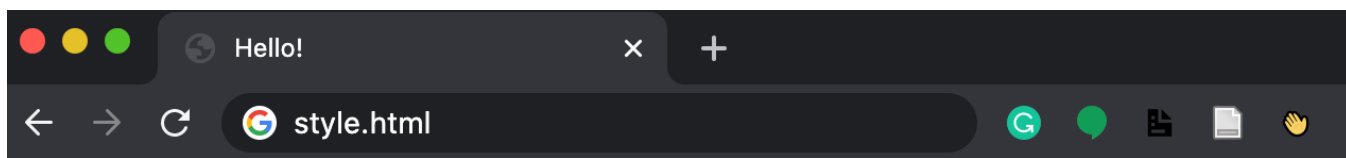
First Name Password

Favorite Color: ☐ Blue ☐ Green ☐ Yellow ☒ Red

Cascading Style Sheets (CSS)

- از CSS به منظور سفارشی ساختن ظاهر وبسایت خود بهره می گیریم.
- گرچه فعلا در آغاز کار خود قرار داریم، ولی در همین سطح نیز با افزودن مشخصه‌ی (style attribute) به هر یک از المان‌های HTML می‌توانیم CSS را به آن اعمال نماییم.
- با تغییر ویژگی‌های CSS، یک المان، می‌توان style آن را تغییر داد، مثلا با نوشتن `color: blue` یا `text-align: center`.
- در مثال زیر، تغییر اندکی بر روی فایل اولیه‌ی خود ایجاد نموده و بخش heading آن را رنگی می‌کنیم:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Hello!</title>
  </head>
  <body>
    <h1 style="color: blue; text-align: center;">A Colorful Heading!</h1>
    Hello, world!
  </body>
</html>
```

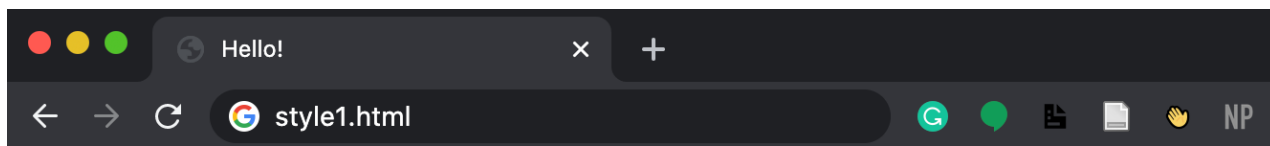



A Colorful Heading!

Hello, world!

- با ایجاد style برای یک المان والد، تمامی المان‌های فرزند آن نیز به طور خودکار از همان style برخوردار خواهند شد. با انتقال style ایجاد شده در تگ heading به تگ body، این تفاوت را می‌توانید مشاهده کنید:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Hello!</title>
  </head>
  <body style="color: blue; text-align: center;">
    <h1 >A Colorful Heading!</h1>
    Hello, world!
  </body>
</html>
```



A Colorful Heading!

Hello, world!

گرچه طبق روند ذکر شده می‌توانیم **style** متفاوتی را برای صفحه‌ی وب خود ایجاد کنیم، اما طراحی صفحه‌ی ما زمانی مطلوب‌تر خواهد شد که بتوانیم این **style** ها را منحصر به برخی خطوط خاص نماییم، طوری که طی آن، باقی خطوط از این **style** برخوردار نباشند.

- یک روش برای انجام این کار افزودن سبک‌ها در درون تگ‌های **<style>** درون تگ **head** است. در درون این تگ‌ها، انواع المان‌های مورد نظر و نوع سبکی را که می‌خواهیم از آن برخوردار باشند را قید می‌کنیم. برای نمونه:

```
<html lang="en">
<!DOCTYPE html>
<head>
  <title>Hello!</title>
  <style>
    h1 {
      color: blue;
      text-align: center;
    }
  </style>
</head>
<body>
  <h1>A Colorful Heading!</h1>
  Hello, world!
</body>
</html>
```

یک روش دیگر برای انجام این کار، استفاده از تگ **<link>** در **head** است، این المان حاوی لینکی به فایل **styles.css** می‌باشد که دربرگیرنده‌ی استایل‌های مورد نظر است. با اعمال این روش، فایل **html** به صورت زیر تغییر خواهد کرد:

<html lang="en">

```
<html lang="en">
<!DOCTYPE html>
<head>
  <title>Hello!</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1>A Colorful Heading!</h1>
  Hello, world!
</body>
</html>
```

متقابلا فایل `styles.css` فراخوانی شده نیز به صورت زیر خواهد بود:

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

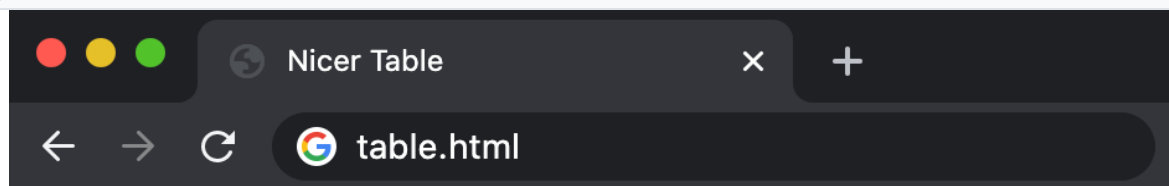
- ویژگی‌های CSS بسیار زیادی وجود دارند که می‌توانیم در مورد آنها بحث کنیم، اما درست به مانند المان‌های HTML، با یک جستجوی ساده در گوگل نظیر “تغییر رنگ فونت به آبی در CSS” می‌توانیم به نتایج مورد نیاز خود دست‌یابیم. با این حال برخی از ویژگی‌های بسیار متداول در ادامه آورده شده است:
- `color`: رنگ متن
- `text-align`: مکان قرارگیری المان‌ها بر روی صفحه
- `background-color`: می‌تواند بر روی هر رنگی تنظیم شود.
- `width`: مقدار آن بر اساس پیکسل یا درصد از صفحه خواهد بود.
- `height`: مقدار آن بر اساس پیکسل یا درصد از صفحه خواهد بود.
- `padding`: چه میزان فضا بایستی در درون هر المان کنار گذاشته شود.
- `margin`: چه میزان فضا بایستی در خارج از هر المان کنار گذاشته شود.
- `font-family`: نوع فونت مربوط به متن درج شده بر روی صفحه
- `font-size`: به واحد پیکسل
- `border`: اندازه، نوع (یکنواخت، خط‌چین، غیره)، رنگ
- بیاید بخشی از آنچه را که تا کنون یادگرفتیم در راستای بهبود جدول اقیانوس‌های موجود در بالا مورد استفاده قرار دهیم. برای آغاز کار، یک کد HTML در ادامه آورده شده است:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Nicer Table</title>
</head>
<body>
  <table>
    <thead>
      <th>Ocean</th>
      <th>Average Depth</th>
      <th>Maximum Depth</th>
    </thead>
```

```

<tbody>
  <tr>
    <td>Pacific</td>
    <td>4280 m</td>
    <td>10911 m</td>
  </tr>
  <tr>
    <td>Atlantic</td>
    <td>3646 m</td>
    <td>8486 m</td>
  </tr>
</tbody>
</table>
</body>
</html>

```



| Ocean | Average Depth | Maximum Depth |
|----------|---------------|---------------|
| Pacific | 4280 m | 10911 m |
| Atlantic | 3646 m | 8486 m |

این HTML بسیار شبیه به همان محتوایی است که پیشتر دیدیم، اما اکنون در المان `head` و با استفاده از تگ `style` یا انجام `link` به یک برگه‌ی سبک، CSS زیر را اضافه می‌کنیم:

```

table {
  border: 1px solid black;
  border-collapse: collapse;
}

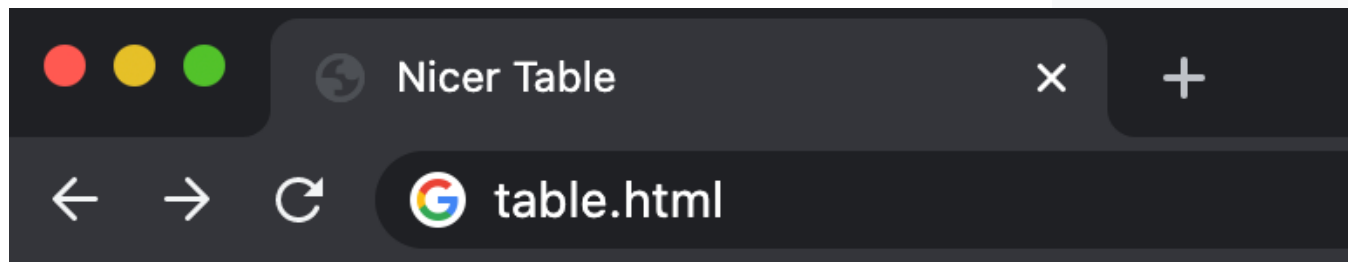
td {
  border: 1px solid black;
  padding: 2px;
}

th {
  border: 1px solid black;

```

```
padding: 2px;
}
```

این خروجی کد ما هستش:



| Ocean | Average Depth | Maximum Depth |
|----------|---------------|---------------|
| Pacific | 4280 m | 10911 m |
| Atlantic | 3646 m | 8486 m |

- احتمالا به این نکته پی برده‌اید که موارد تکراری و غیرضروری زیادی در این CSS وجود دارد، چراکه td و th دارای style هایی مشابه یکدیگر هستند. این تکرارها را می‌توانیم (بهتر است) به صورت کد زیر خلاصه کنیم، با استفاده از کاما مشخص می‌کنیم که این style خاص، بر روی بیش از یک نوع المان اعمال خواهد شد.

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

td, th {
  border: 1px solid black;
  padding: 2px;
}
```

- این مثال مقدمه‌ی خوبی برای موضوع [CSS Selectors](#) یا انتخاب‌گرهای CSS است. روش‌های مختلفی برای تعیین المان‌های HTML دلخواه، جهت اعمال یک style خاص وجود دارد، که در اینجا به برخی از آنها اشاره خواهیم کرد:
- نوع المان: این همان روشی است که تا بدینجا از آن استفاده کرده‌ایم: یعنی اعمال style بر روی المان‌هایی از یک نوع.
- id: روش دیگر آن است که ابتدا یک شناسه‌ی id را به المان‌های html و به صورت first="h1 id" <h1/>!Hello <"header" header اختصاص دهیم، سپس با استفاده از {...}first-header# سبک مورد

نظر را اعمال نماییم، در اینجا علامت # بیانگر آن است که جستجو از طریق id صورت می گیرد. هیچ دو المانی نمی توانند از یک id یکسان برخوردار باشند، و هیچ المانی نمی تواند بیش از یک id را اختیار کند.

- class: کلاس نیز مانند id است، با این تفاوت که از یک کلاس برای چندین المان می توانیم بهره بگیریم و متقابلاً هر المان واحد نیز می تواند به بیش از یک کلاس تعلق داشته باشد. برای افزودن کلاس به HTML به اینصورت عمل می کنیم : `<h1 class="page-text muted">Hello</h1>` (توجه داشته باشید که دو کلاس به المان خود اضافه کرده ایم: page-text و muted). سپس براساس کلاس و با بهره گیری از علامت . به جای # سبک مورد نظر را اعمال می کنیم: {...} muted.

- اکنون باید به مسئله ی بروز تداخلات احتمالی در CSS پردازیم. چنانچه المان header براساس کلاس آن قرمز و براساس id آن آبی تعریف شود، چه اتفاقی رخ خواهد داد؟ در رابطه با این مسئله، CSS از یک ترتیب ویژگی برخوردار است که از قرار زیر است:

- style سازی طی خطوط
- id
- class
- نوع المان

برای تعیین المان های مورد نظر جهت اعمال style، علاوه بر کاما روش های دیگری نیز وجود دارد. جدول زیر برخی از این روش ها را لیست کرده است که در ادامه تعدادی از آنها را توضیح خواهیم داد:

| | |
|-------|---------------------------|
| a, b | Multiple Element Selector |
| a b | Descendant Selector |
| a > b | Child Selector |
| a + b | Adjacent Sibling Selector |
| [a=b] | Attribute Selector |
| a:b | Pseudoclass Selector |
| a::b | Pseudoelement Selector |

انتخاب گر نسل (Descendant Selector): در اینجا با استفاده از انتخاب گر نسل، style را تنها بر روی آیتم های لیستی اعمال می کنیم که در درون یک لیست نامرتب قرار دارند:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Using Selectors</title>
    <style>
      ul li {
        color: blue;
      }
    </style>
```

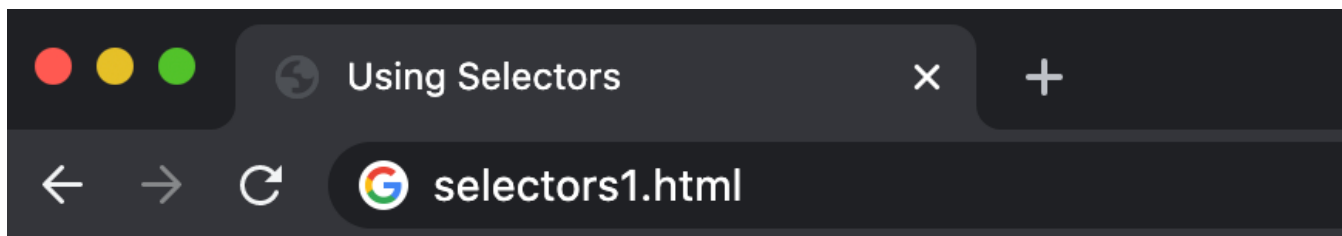
```
</head>
<body>
  <ol>
    <li>foo</li>
    <li> bar
      <ul>
        <li>hello</li>
        <li>goodbye</li>
        <li>hello</li>
      </ul>
    </li>
    <li>baz</li>
  </ol>

</body>
<html>
```

مشخصه‌ها به عنوان انتخاب‌گرها (**Attributes as Selectors**): براساس خصیصه‌هایی که با استفاده از براکت‌ها به المان‌های HTML اختصاص می‌دهیم، گزینه‌های خود را بیش از این نیز می‌توانیم محدودتر نماییم. برای مثال، در لیست لینک‌های زیر، تعیین می‌کنیم که تنها لینک موجود به سایت آمازون، به رنگ قرمز تغییر نماید:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Using Selectors</title>
    <style>
      a[href="https://www.amazon.com/"] {
        color: red;
      }
    </style>
  </head>
  <body>
    <ol>
      <li><a href="https://www.google.com/">Google</a></li>
      <li><a href="https://www.amazon.com/">Amazon</a> </li>
      <li><a href="https://www.facebook.com/">Facebook</a></li>
    </ol>

  </body>
</html>
```



1. [Google](#)
2. [Amazon](#)
3. [Facebook](#)

- با استفاده از CSS علاوه بر تغییر دائمی ظاهر یک المان، تحت شرایط خاص نیز می‌توانیم ظاهر المان را تغییر دهیم. برای مثال، فرض کنید که می‌خواهیم در صورت اشاره‌ی ماوس بر روی یک کلید، رنگ آن کلید تغییر یابد. برای انجام این کار می‌توانیم از یک شبه کلاس CSS بهره بگیریم که سبک‌های اضافی را تحت شرایط خاص فراهم می‌آورد. جهت نوشتن کد مربوطه، بعد از انتخاب‌گر خود از یک علامت کولن استفاده کرده، و در ادامه‌ی آن، شرایط مورد نظر را قید می‌کنیم.
- در رابطه با کلید، به منظور تعیین طراحی مرتبط در زمان اشاره با ماوس، `hover` را به انتخاب‌گر دکمه اضافه می‌کنیم:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Pseudoclasses</title>
    <style>
      button {
        background-color: red;
        width: 200px;
        height: 50px;
        font-size: 24px;
      }

      button:hover {
        background-color: green;
      }
    </style>
  </head>
  <body>
    <button>Button 1</button>
```

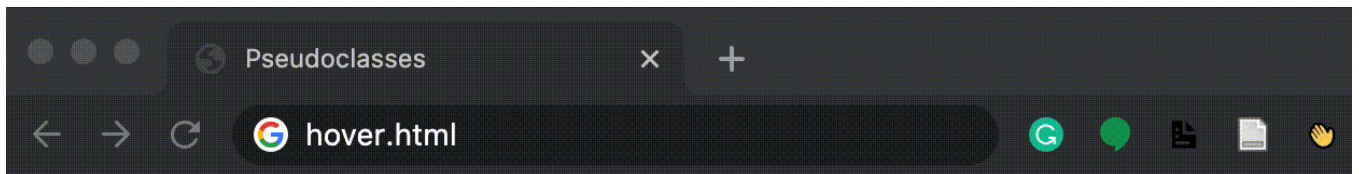


```

<button>Button 2</button>
<button>Button 3</button>

</body>
<html>

```



<https://cs50.harvard.edu/web/2020/notes/0/images/buttons.gif>

طراحی واکنش‌گرا (responsive design)

- امروزه بسیاری از افراد، سایت‌ها را در دستگاه‌هایی به غیر از کامپیوترها، نظیر گوشی‌های هوشمند و تبلت‌ها مشاهده می‌کنند. حصول اطمینان از اینکه وب‌سایت ما بر روی تمامی دستگاه‌ها قابل مشاهده خواهد بود، بسیار حائز اهمیت است.
- یک روش برای انجام این کار، درک مفهوم viewport است. viewport بخشی از صفحه است که در هر زمان خاص برای کاربر قابل رویت می‌باشد. به طور پیش‌فرض بسیاری از صفحات وب (به ویژه سایت‌های قدیمی) مقادیر viewport تمامی دستگاه‌ها را یکسان تلقی می‌کنند، که همین امر برقراری تعامل با سایت را بر روی دستگاه‌های موبایل دشوار می‌سازد.
- یک روش ساده برای تقویت ظاهر یک سایت بر روی دستگاه موبایل آن است که در بخش head فایل‌های HTML از کد زیر استفاده کنیم. این خط کد به دستگاه موبایل اعلام می‌کند که از viewport ای بهره بگیرد که پهنای آن مشابه پهنای دستگاه موبایل باشد و نه یک پهنای بزرگتر.
- روش دیگر جهت مدیریت طریقه‌ی نمایش صفحه وب بر روی دستگاه‌های مختلف آن است که از [Media query](#) ها بهره بگیریم. Media query ها روشی برای تغییر style صفحه و براساس طریقه‌ی نمایش آن است.

```

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

```

- به عنوان مثالی در زمینه‌ی media query، بیایید رنگ صفحه را در صورت کوچک شدن صفحه به یک اندازه‌ی خاص، تغییر دهیم. برای فراخوانی یک media query، عبارت @media را نوشته و سپس نوع کوئری را در درون یک جفت پرانتز مشخص می‌کنیم.

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>Screen Size</title>
<style>
  @media (min-width: 600px) {

```

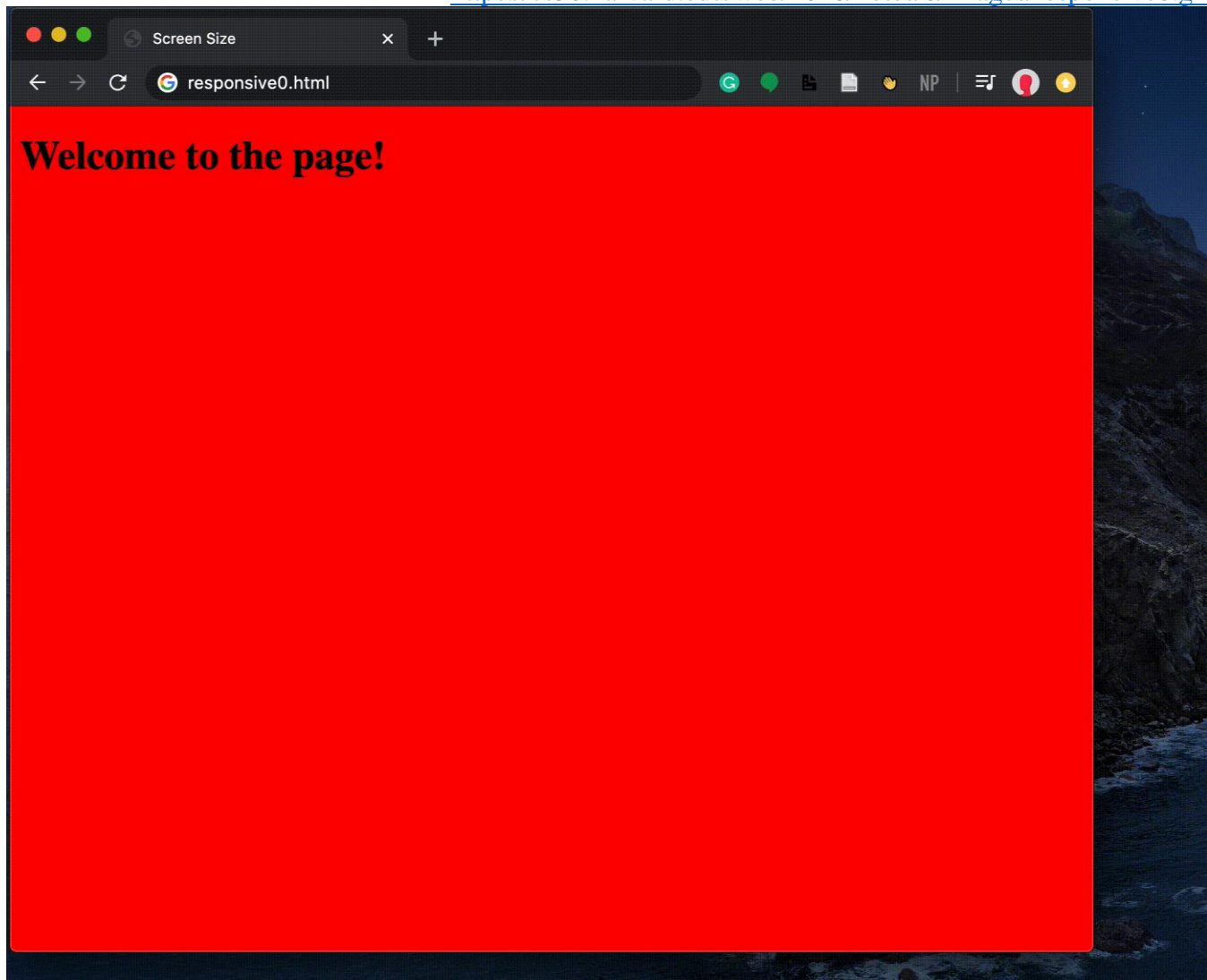
```

body {
  background-color: red;
}

@media (max-width: 599px) {
  body {
    background-color: blue;
  }
}
</style>
</head>
<body>
  <h1>Welcome to the page!</h1>
</body>
</html>

```

<https://cs50.harvard.edu/web/2020/notes/0/images/responsive0.gif>

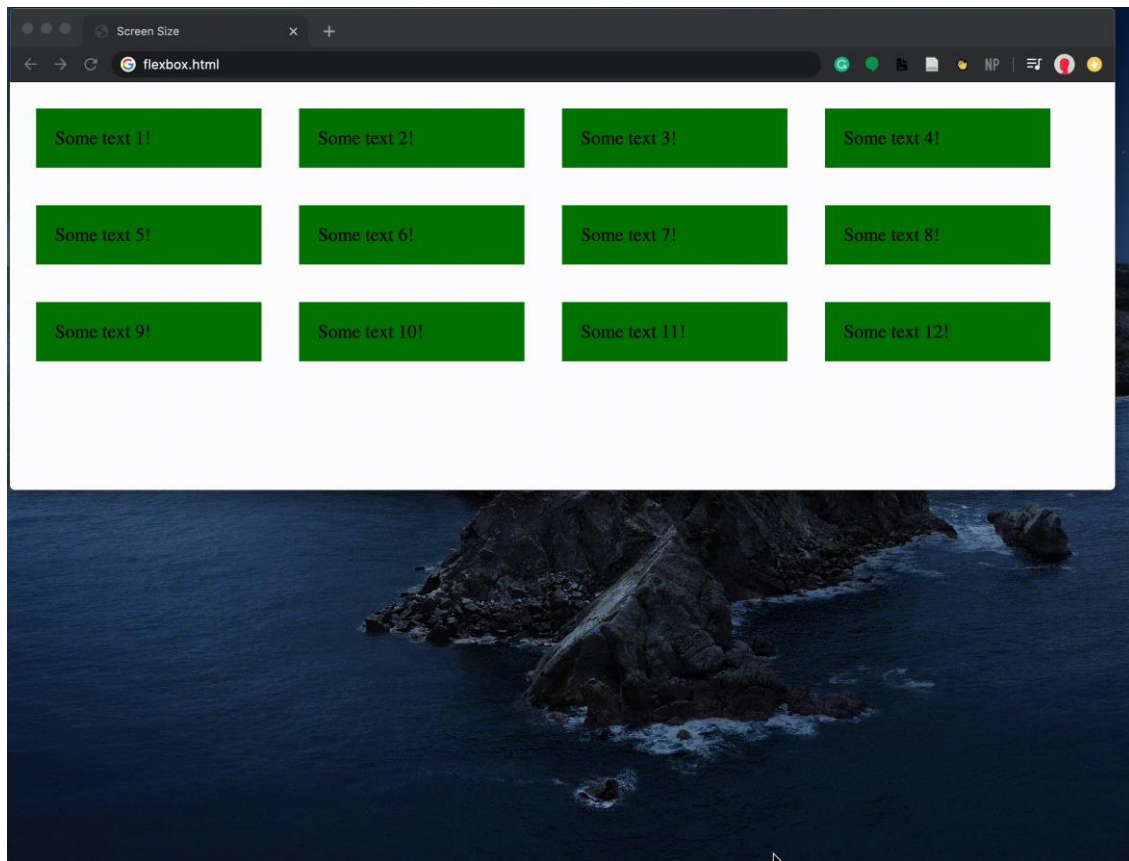


- روش دیگر برای مدیریت اندازه‌های مختلف صفحه‌ی نمایش، بهره‌گیری از مشخصه‌ی جدید CSS به نام flexbox است. با استفاده از این مشخصه چنانچه المان‌ها به صورت افقی در یک صفحه جای‌نگیرند، دچار پیچش شده و در اطراف خط بعدی قرار خواهند گرفت. برای انجام اینکار تمام المان‌ها را در درون یک div قرار می‌دهیم که به آن نگهدارنده می‌گوییم. سپس برای این div یک سبک خاص را اضافه نموده و تعیین می‌کنیم که برای المان‌های

داخل آن می‌خواهیم از display ای با مقدار flexbox بهره بگیریم. در اینجا با هدف نمایش بهتر عملیات پیچش، سبک متفاوتی را برای div های داخلی تعیین می‌کنیم.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Screen Size</title>
    <style>
      #container {
        display: flex;
        flex-wrap: wrap;
      }

      #container > div {
        background-color: green;
        font-size: 20px;
        margin: 20px;
        padding: 20px;
        width: 200px;
      }
    </style>
  </head>
  <body>
    <div id="container">
      <div>Some text 1!</div>
      <div>Some text 2!</div>
      <div>Some text 3!</div>
      <div>Some text 4!</div>
      <div>Some text 5!</div>
      <div>Some text 6!</div>
      <div>Some text 7!</div>
      <div>Some text 8!</div>
      <div>Some text 9!</div>
      <div>Some text 10!</div>
      <div>Some text 11!</div>
      <div>Some text 12!</div>
    </div>
  </body>
</html>
```



<https://cs50.harvard.edu/web/2020/notes/0/images/flexbox.gif>

• یک روش متداول دیگر برای تعیین style یک صفحه، بهره‌گیری از HTML grid است. در این گرید مشخصه‌های style را می‌توانیم تعیین کنیم، برای نمونه مشابه آنچه که در زیر نشان داده شده است، پهنای ستون‌ها، و فاصله‌ی میان ستون‌ها و سطرها را می‌توانیم مشخص کنیم. توجه داشته باشید که در زمان تعیین پهنای ستون‌ها، سومین مورد را بر روی auto تنظیم می‌کنیم، بدان معنا که باقی صفحه باید به طور خودکار پر شود.

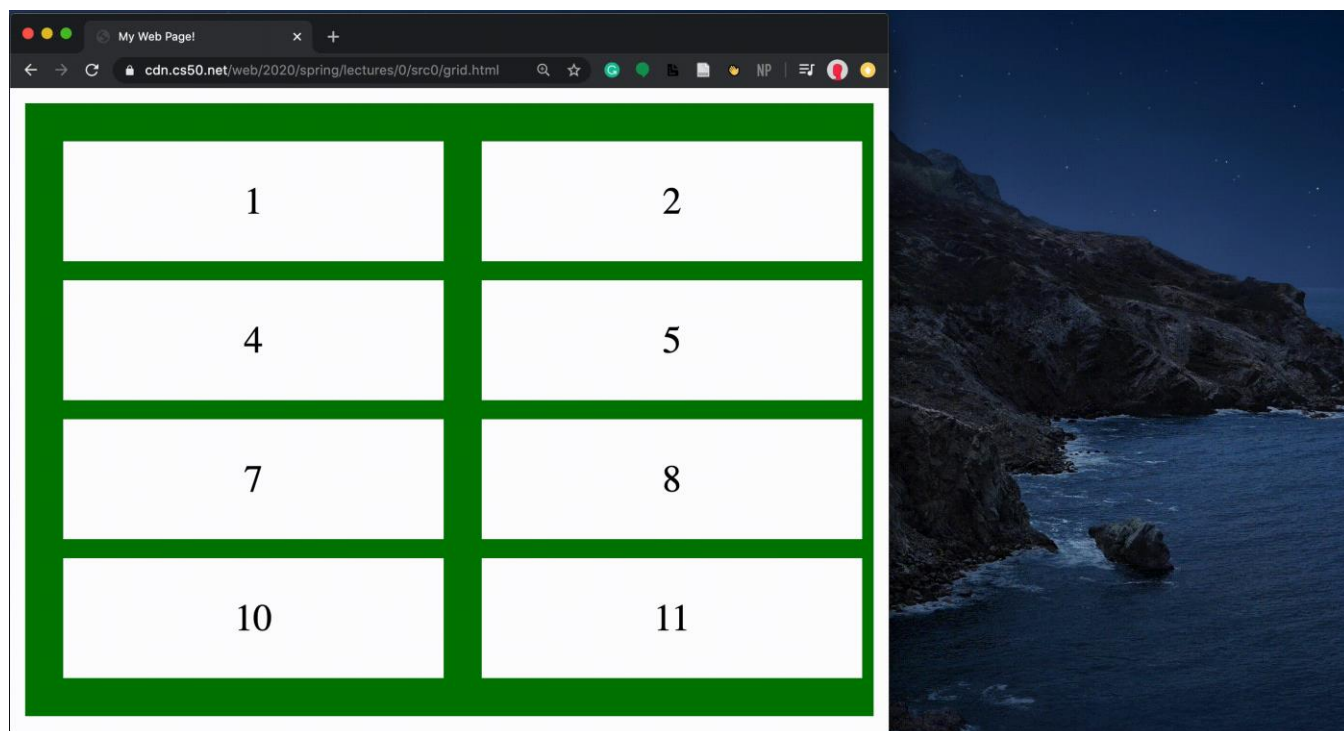
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>My Web Page!</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
      .grid {
        background-color: green;
        display: grid;
        padding: 20px;
        grid-column-gap: 20px;
        grid-row-gap: 10px;
        grid-template-columns: 200px 200px auto;
      }

      .grid-item {
        background-color: white;
        font-size: 20px;
        padding: 20px;
        text-align: center;
      }
    </style>
  </head>
  <body>
```

```

<div class="grid">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
  <div class="grid-item">7</div>
  <div class="grid-item">8</div>
  <div class="grid-item">9</div>
  <div class="grid-item">10</div>
  <div class="grid-item">11</div>
  <div class="grid-item">12</div>
</div>
</body>
</html>

```



<https://cs50.harvard.edu/web/2020/notes/0/images/grid.gif>

Bootstrap

- ظاهراً کتابخانه‌های آماده‌ی بسیار زیادی توسط دیگر افراد نوشته شده است، که سبک‌گذاری صفحات وب را به مراتب آسان‌تر می‌سازند. یکی از این کتابخانه‌ها که در طول این دوره نیز از آن استفاده خواهیم کرد، Bootstrap نام دارد.
- با افزودن یک خط کد به بخش head فایل HTML، به سادگی می‌توانیم bootstrap را به کد خود اضافه کنیم:

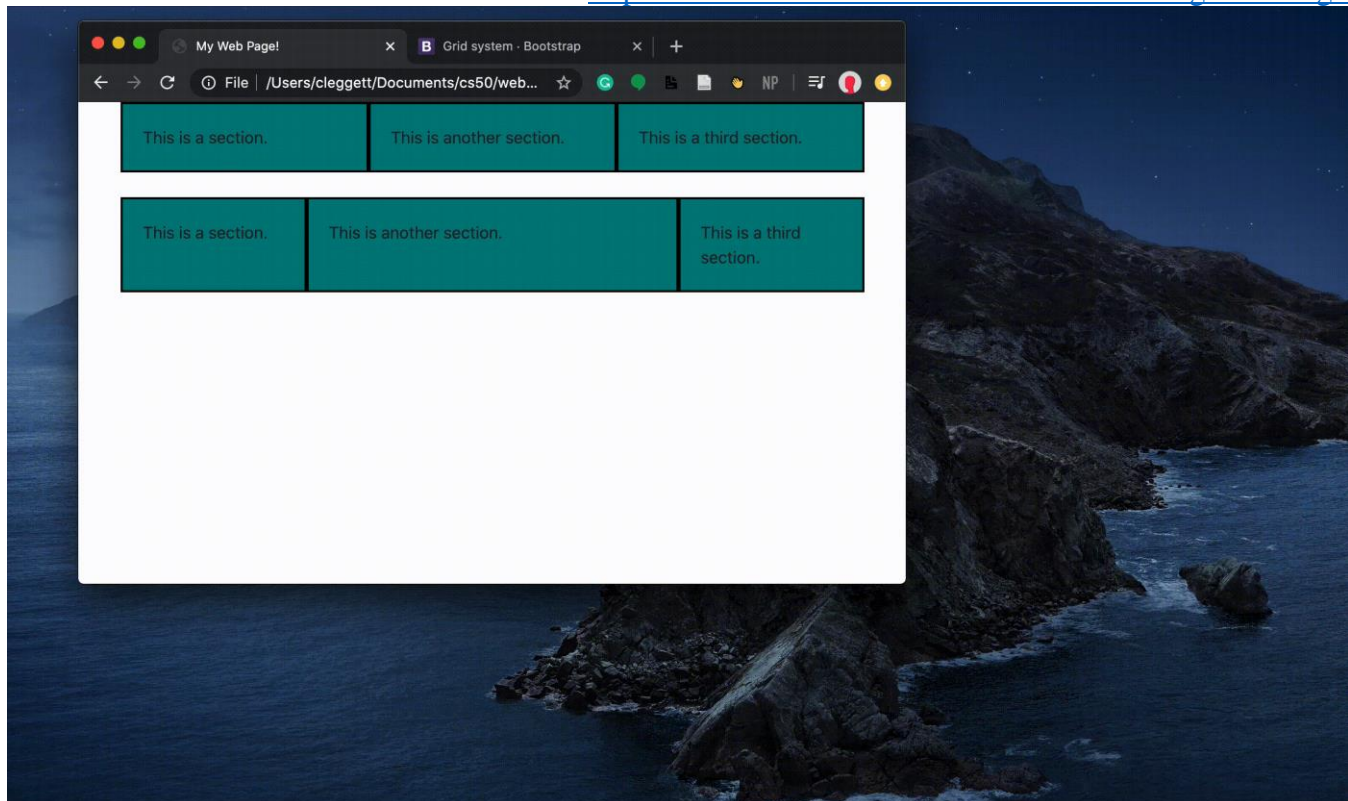
```

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYYxFfc+NcPb1dKGj7Sk"
crossorigin="anonymous">

```

- یکی از مشخصه‌های متداول Bootstrap، سیستم grid آن است. Bootstrap به طور پیش فرض 12 ستون را به 12 ستون تقسیم‌بندی می‌کند. با افزودن col-x می‌توان تعیین نمود که هر المان از چند ستون تشکیل شود، مقدار x عددی مابین 1 تا 12 خواهد بود. برای نمونه در صفحه‌ی بعدی، یک سطر وجود دارد که از چندین ستون با پهنای یکسان تشکیل شده است، بعد از آن یک سطر دیگر وجود دارد که ستون مرکزی آن نسبت به ستون‌های دیگر بزرگتر است:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>My Web Page!</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
    integrity="sha384-Vkoo8x4CGs03+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
    crossorigin="anonymous">
    <style>
      .row > div {
        padding: 20px;
        background-color: teal;
        border: 2px solid black;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <div class="row">
        <div class="col-4">
          This is a section.
        </div>
        <div class="col-4">
          This is another section.
        </div>
        <div class="col-4">
          This is a third section.
        </div>
      </div>
    </div>
    <br/>
    <div class="container">
      <div class="row">
        <div class="col-3">
          This is a section.
        </div>
        <div class="col-6">
          This is another section.
        </div>
        <div class="col-3">
          This is a third section.
        </div>
      </div>
    </div>
  </body>
</html>
```

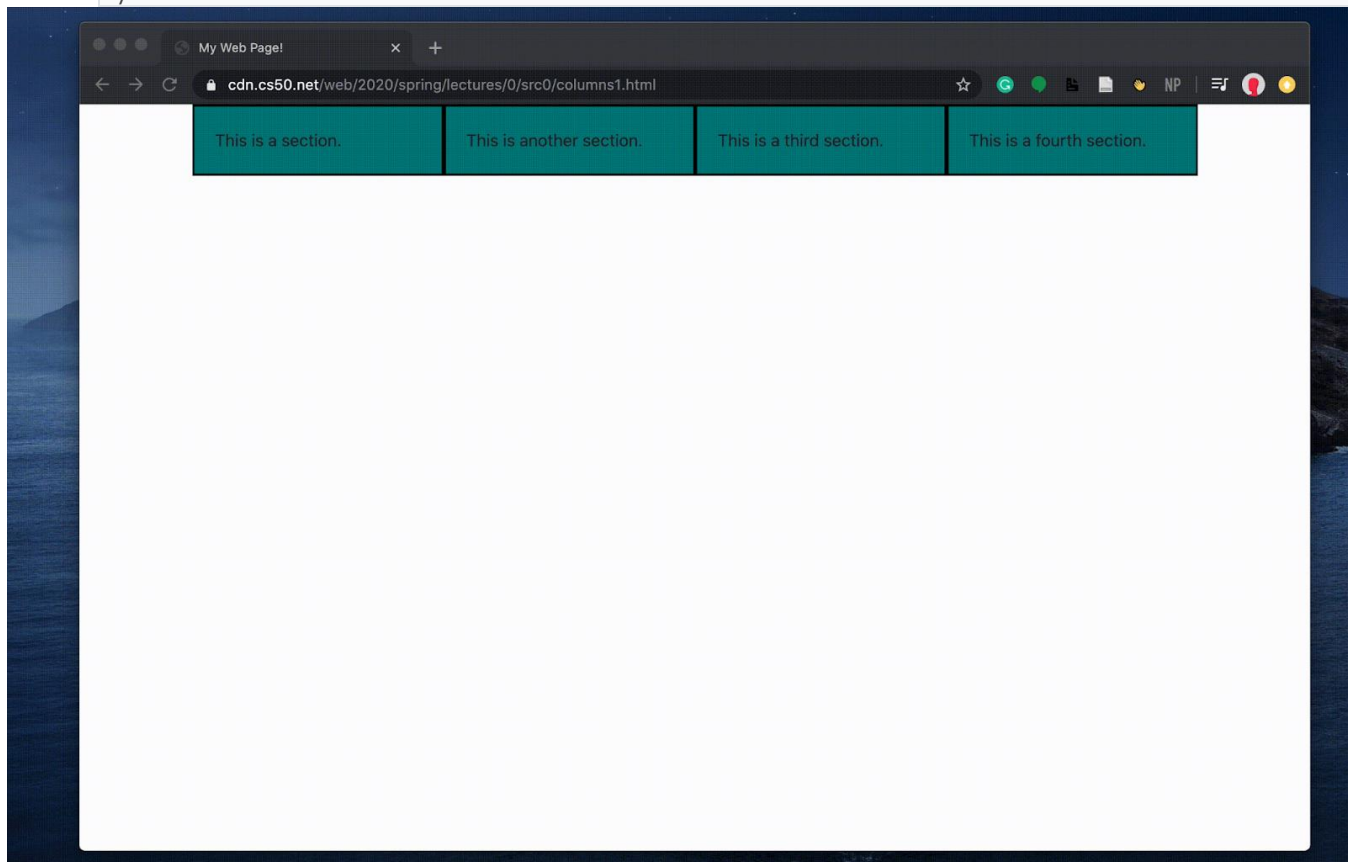
- در راستای ارتقا خاصیت واکنش‌گرا بودن برای موبایل، Bootstrap این امکان را فراهم کرده است که اندازه‌ی ستون را متناسب با اندازه‌ی صفحه تعیین کنیم. در مثال بعدی با بهره‌گیری از `col-lg-3` تعیین می‌کنیم که در یک صفحه با اندازه‌ی بزرگ، المان مورد نظر ما بایستی از 3 ستون تشکیل شود، و با استفاده از دستور `col-sm-6` تعیین می‌کنیم که در صفحاتی با اندازه‌ی کوچک، این المان بایستی دارای 6 ستون باشد:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>My Web Page!</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
    integrity="sha384-Vkoo8x4CGs03+Hhvx8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
    crossorigin="anonymous">
    <style>
      .row > div {
        padding: 20px;
        background-color: teal;
        border: 2px solid black;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <div class="row">
        <div class="col-lg-3 col-sm-6">
          This is a section.
        </div>
        <div class="col-lg-3 col-sm-6">
          This is another section.
        </div>
        <div class="col-lg-3 col-sm-6">
          This is a third section.
        </div>
      </div>
    </div>
  </body>
</html>
```

```

</div>
<div class="col-lg-3 col-sm-6">
  This is a fourth section.
</div>
</div>
</div>
</body>
</html>

```



<https://cs50.harvard.edu/web/2020/notes/0/images/cols2.gif>

Sass

- به منظور حذف افزونگی در CSS، تا کنون از روش‌های متعددی نظیر انتقال بخش‌های زائد به فایل‌های جداگانه و یا استفاده از کتابخانه‌ی Bootstrap بهره‌گرفتیم، با اینحال موارد دیگری نیز وجود دارند که می‌توانند بیشتر از این بهبود یابند. برای نمونه، فرض کنید که در چندین المان مختلف می‌خواهیم از style های متفاوتی بهره بگیریم، با این حال تمایل داریم که این المان‌ها از رنگ یکسانی نیز برخوردار باشند. چنانچه بعداً تصمیم بگیریم که رنگ المان‌ها تغییر دهیم، در اینصورت مجبور خواهیم بود که تغییر رنگ را در درون المان‌های مختلف به انجام برسانیم.
- Sass زبانی است که کدنویسی کارآمدتر CSS را به روش‌های مختلفی فراهم می‌آورد، یکی از این روش‌ها بهره‌گیری از متغیرها می‌باشد که نمونه‌ای از آن در مثال بعدی آورده شده است.
- در زمان کدنویسی در Sass، یک فایل با افزونه‌ی filename.scss ایجاد می‌کنیم. در این فایل با افزودن \$ پیش از یک نام و سپس استفاده از علامت کولن و تعیین یک مقدار می‌توانیم متغیرهای جدیدی را تعریف کنیم. برای مثال با نوشتن color: red\$ متغیر color را با مقدار red تنظیم می‌کنیم. سپس با استفاده از color\$ می‌توانیم به این متغیر دسترسی داشته باشیم. در اینجا نمونه‌ای از فایل variables.scss آورده شده است:

\$color


```
$color: red;

ul {
  font-size: 14px;
  color: $color;
}

ol {
  font-size: 18px;
  color: $color;
}
```

- اکنون به منظور لینک نمودن این نوع سبک‌گذاری به فایل HTML خود، نمی‌توانیم صرفاً عملیات لینک به فایل SCSS را به انجام برسانیم. چراکه اکثر مرورگرها تنها فایل‌هایی با پسوند CSS را تشخیص می‌دهند. برای حل این مشکل، ابتدا بایستی یک برنامه با نام Sass را بر روی کامپیوتر خود دانلود نمایید. سپس در ترمینال خود این خط فرمان را تایپ کنید: `Sass variables.scss:variables.css`. این خط فرمان، یک فایل با پسوند SCSS و نام `variables.scss` را به فایلی با پسوند CSS و نام `variables.css` کامپایل خواهد کرد، که یک لینک به صفحه‌ی HTML خود را می‌توانید به آن اضافه کنید.
- جهت تسریع این روند، می‌توانید از فرمان `sass--watch variables.scss:variables.css` بهره بگیرید، این فرمان در هر بار تشخیص تغییر در فایل SCSS، تغییرات مربوطه را به طور خودکار در فایل CSS نیز اعمال خواهد کرد.
- در زمان بهره‌گیری از Sass، به جای آنکه مثل سابق از انتخاب‌گرهای CSS استفاده کنیم، به طور فیزیکی می‌توانیم style مورد نظر خود را پیاده‌سازی کنیم. برای نمونه، چنانچه بخواهیم style خاصی را به پاراگراف‌ها یا لیست‌های نامرتب موجود در درون یک div اعمال کنیم، به سادگی می‌توانیم از دستورات زیر بهره بگیریم:

```
div {
  font-size: 18px;

  p {
    color: blue;
  }

  ul {
    color: green;
  }
}
```

به محض اتمام عملیات کامپایل به فایل CSS، فایلی به شکل زیر خواهیم داشت:

```
div {
  font-size: 18px;
}

div p {
  color: blue;
}

div ul {
  color: green;
}
```

- مشخصه‌ی دیگری که Sass ارائه می‌کند، inheritance یا وراثت نام دارد. با استفاده از این مشخصه می‌توانیم یک مجموعه‌ی پایه از سبک‌گذاری‌ها را ایجاد نموده و آنها را در میان چندین المان مختلف به اشتراک بگذاریم. برای انجام آن ابتدا پیش از نام کلاس بایستی از علامت % استفاده کنیم، سپس style مورد نظر خود را اضافه نموده و بعد از آن دستور @extend %classname را به ابتدای یک سبک‌گذاری اضافه می‌کنیم. برای مثال، کد آورده شده در زیر style موجود در درون کلاس message را به هر یک از کلاس‌های مختلف زیر اعمال می‌کند، که در نتیجه‌ی آن یک صفحه‌ی وب، مشابه صفحه‌ی زیر ایجاد خواهد شد.

```
%message {  
  font-family: sans-serif;  
  font-size: 18px;  
  font-weight: bold;  
  border: 1px solid black;  
  padding: 20px;  
  margin: 20px;  
}  
  
.success {  
  @extend %message;  
  background-color: green;  
}  
  
.warning {  
  @extend %message;  
  background-color: orange;  
}  
  
.error {  
  @extend %message;  
  background-color: red;  
}
```

This is a success message.

This is a warning message.

This is an error message.