Make and deploy a website

https://material.io/guidelines/layout/responsive-ui.html#responsive-ui-breakpoints

<!DOCTYPE html>
<html>
<head>
<title>Ollie Bike Sharing</title>
<meta charset="utf-8"/>
<link rel="stylesheet" type="text/css" href="main.css">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"/>

</head>
<body>
</bod>
</html>

Languages

- **html**: stands for *hypertext markup language*, and is used to give a webpage structure.

- **css**: stands for *cascading style sheets*, and is used to style HTML elements.

HTML Elements

- *h1 - h6*: indicate text headings on a webpage. h1 is the largest heading; h6 is the smallest.
  <h1>Heading</h1>
-

- *p*: used for non-heading text, such as the bodies of articles or company descriptions.
  Description of company here.
-

- *a*: short for anchor and used to add links to other webpages. Anchor elements typically have an href attribute:
  Click here to learn how to make a website!
-

- *img*: used to add an image to a webpage. Image elements are *self-closing*

and do not require a closing tag:
<img src="https://images.com/favorite.png">

- *video*: used to add videos to a webpage, and uses multiple attributes and a nested source element:
<video width="320" height="240" controls>
  <source src="https://movies.io/great-clip.mp4" type="video/mp4">
</video>

- *unordered list*: used to create lists on a webpage and requires li elements inside a ul:

  list item
  another item
  yet another

- *div*: used to organize HTML elements into different groups, which can be given a class attribute:

  <h2>Subheading!</h2>

- *metadata tags*: provide metadata about a webpage.

Web Concepts

- **parent/child elements**: used to describe HTML elements that enclose or are enclosed by other elements. For example, below the ul is the parent and the li items are children:

  ...
  ...
  ...

Click Up Next to start learning about CSS!

The HTML ***video* element** can add video to a webpage.

<video width="320" **height**="240" **controls**>

```
  <source src="video-url.mp4" type="video/mp4">
</video>
```

The video element uses a number of attributes. Let's take a look at them:

1. width and height: Set the size of the screen that displays the video.

2. controls: Adds play, pause and volume control.

3. source src: Sets the URL of the video to play.

4. type: Specifies different video formats.

CSS:

1. *rule*: a list of CSS instructions for how to style a specific HTML element or group of HTML elements.

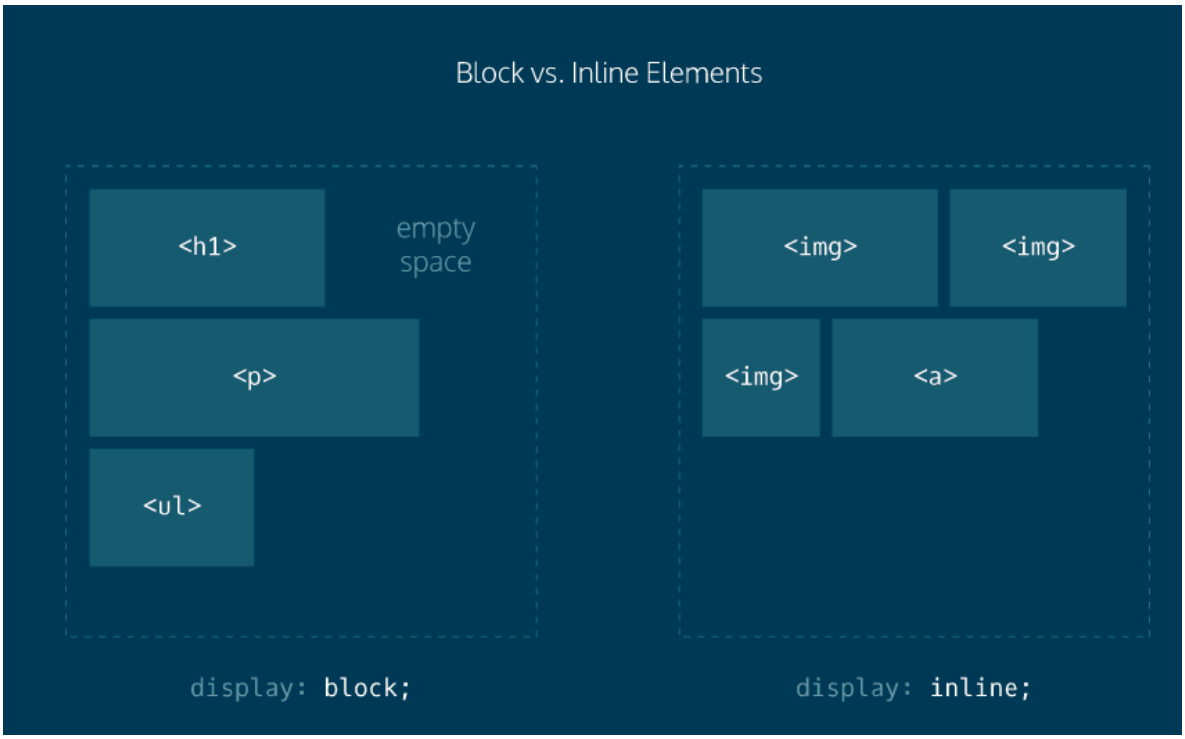
2. *selector*: specifies exactly which HTML elements to style. Here h1 is the selector.

3. *properties* and *values*: located inside the { } brackets, properties and values specify what aspect of the selector to style. In the diagram's example, the color property is set to red, which will display all h1 elements in red.

In the diagram, notice:

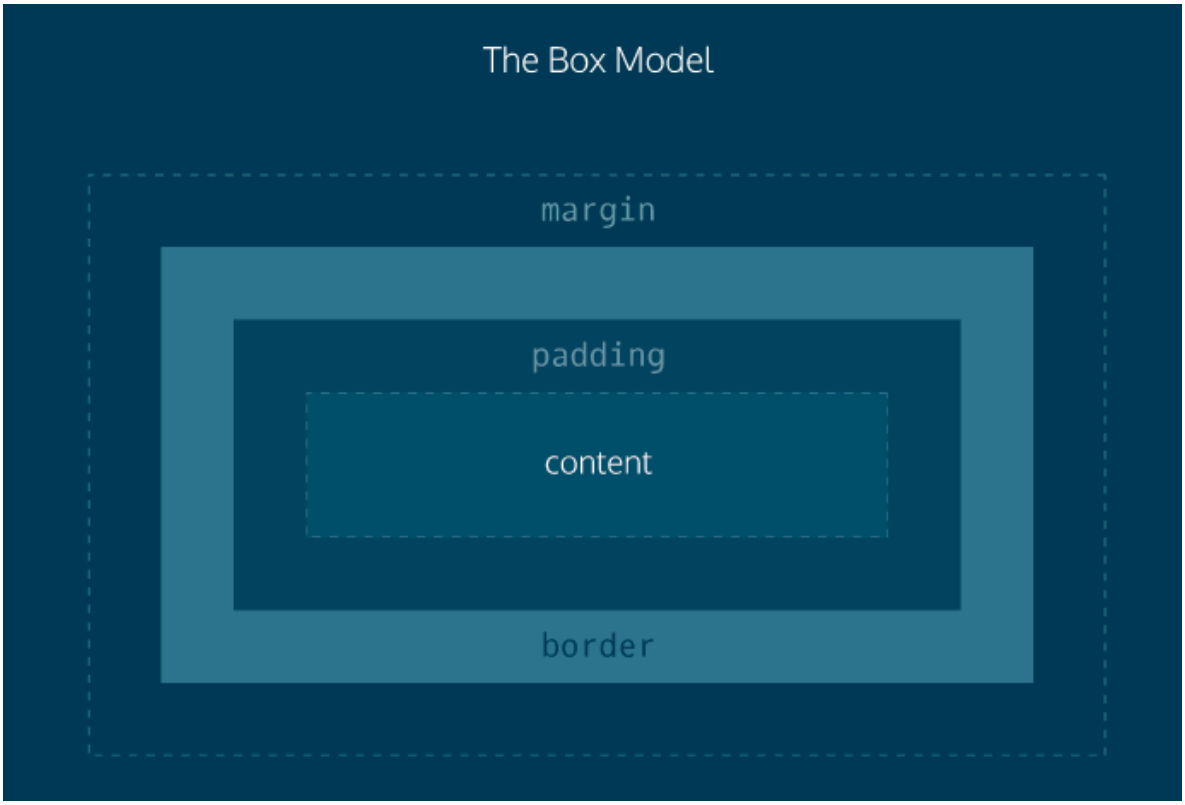
1. The two dotted rectangles represent webpages.

2. HTML heading, paragraph, and unordered list elements are block level: each appears on its own line on the webpag



3. e.

4. HTML image and anchor elements are displayed inline: they appear on the same line as their neighboring elements on the webpage.




```
.gallery {
  margin-top: 20px;
```

```
  display: flex;
  flex-wrap: wrap;
  justify-content:  center;
}
```

Web Concepts

- *CSS Box Model*: illustrates the space and boundary properties of an HTML element that can be controlled using CSS.

CSS Skills

- *border*: sets the outline of an HTML page element, like a picture frame that contains the element.

- *padding*: sets the amount of space between an element's content and its border.

- *margin*: sets the amount of space between an HTML element and the next nearest element(s)

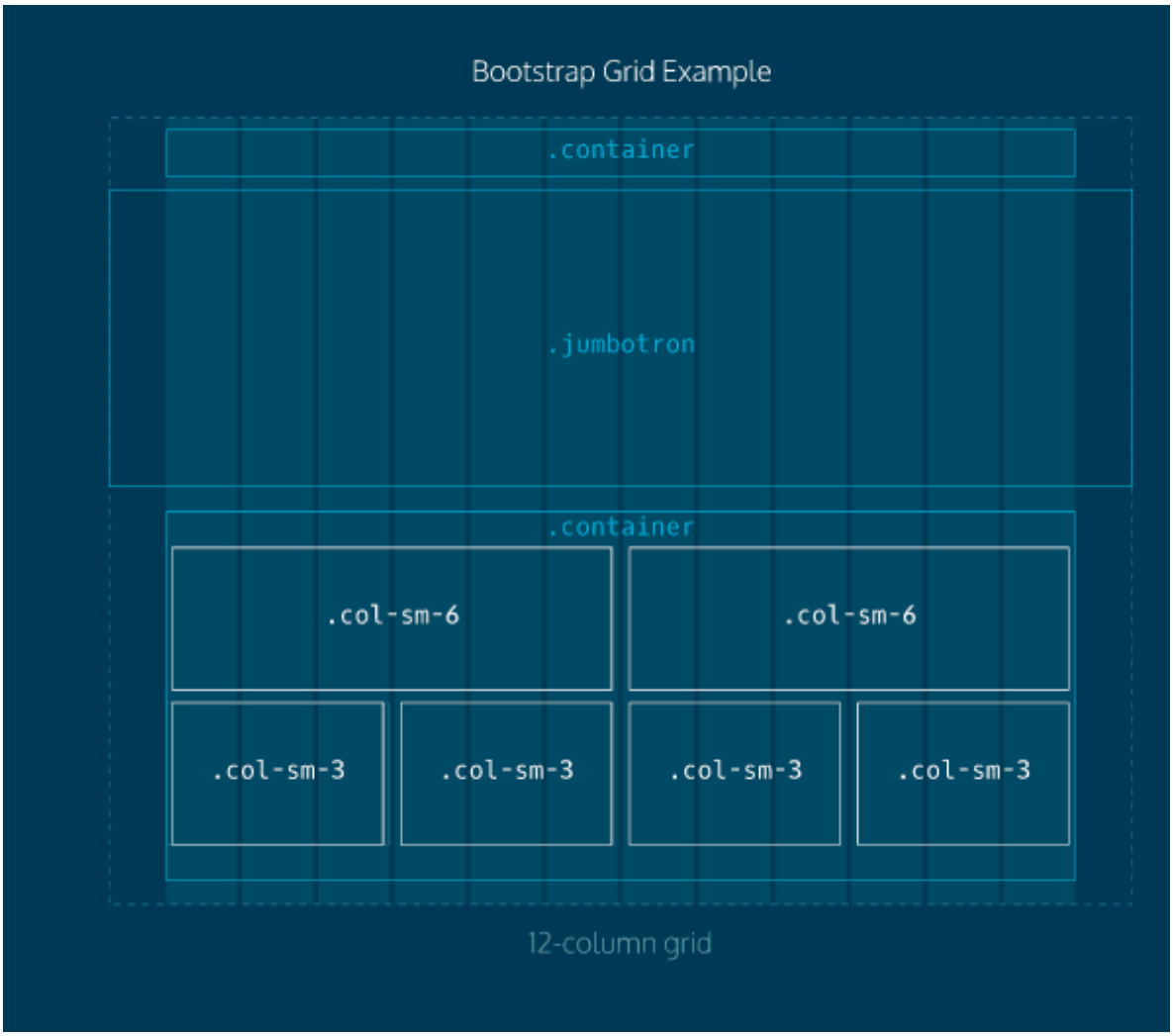
- *display*: property that determines how the selected element will be arranged in relation to other HTML elements on the page.

- *inline*: display value used to arrange HTML elements on the same line as neighboring elements.

- ***flex*: display value that allows us to easily align multiple page elements vertically or horizontally.**

- *float*: property used to float HTML elements left or right of neighboring elements.

- *position*: property used to position HTML elements in exact locations on a webpage.



In the diagram, observe the following:

1. Bootstrap's grid columns are represented by 12 vertical bars. The boxes represent HTML elements.

2. The words "container", "jumbotron", "col-sm-6" and "col-sm-3" refer to Bootstrap classes.

3. The element with class "jumbotron" spans the entire width of the webpage, beyond the borders of the grid.

4. Elements inside the second "container", such as **"col-sm-6**" and "**col-sm-3**" are contained within the grid columns.

5. Elements labeled "col-sm-3" take up three grid columns; elements labeled "col-sm-6" take up six grid colu

Many websites have a supporting content area. Supporting content can be

arranged using Bootstrap's grid. Below is an example implementation of a supporting content area.
First, an HTML section element with the container class is used:

```
<section class="container">

</section>
```

Next, div elements with the row class are added:

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

Finally, the rows are divided by using divs with Bootstrap's col-sm-... class.

```
<section class="container">
  <div class="row">
    <div class="col-sm-6">
          <figure class="col-sm-6">
    </figure>
    </div>
    <div class="col-sm-6">
     ...
    </div>
  </div>
  <div class="row">
    <div class="col-sm-6">
     ...
    </div>
    <div class="col-sm-6">
     ...
    </div>
  </div>
</section>
```

Above, two rows are divided into two equal parts. Each part takes up 6 of bootstrap's 12 columns. Using the col-sm-6 class ensures that this layout will appear when the user's screen is the width of a tablet device(768 pixels). On narrower screens, such as an iPhone, only one image per row will appear.
Let's create a supporting content area for our webpage!

```
  <footer class="container">
<div class="row">
    </div>
    <footer class="container">
  <div class="row">
    <p class="col-sm-4">&copy; 2016 Skillfair</p>
    <ul class="col-sm-8">
<li class="col-sm-1">
```

```
  <img src="https://s3.amazonaws.com/codecademy-content/projects/make-a-website/lesson-4/twitter.svg">
</li>

  <li class="col-sm-1">
  <img src="https://s3.amazonaws.com/codecademy-content/projects/make-a-website/lesson-4/facebook.svg">
</li>

  <li class="col-sm-1">
  <img src="https://s3.amazonaws.com/codecademy-content/projects/make-a-website/lesson-4/instagram.svg">
</li>
<li class="col-md-1">
  <img src="https://s3.amazonaws.com/codecademy-content/projects/make-a-website/lesson-4/medium.svg">
</li>

    </ul>
  </div>
</footer>
```