Intro to HTML + CSS

School of Visual Concepts
Scott Thiessen, instructor

http://movingobjects.io/svc/
scott@movingobjects.io

Week 4

- Review
- CSS Box Model
- CSS Positioning
- Flexbox
- Coding from a Design file

Review

Block tags

`<h1>` `<h2>` `<h3>`

`<p>`

`<ul>` `<ol>`

`<li>`

`<main>` `<nav>`

`<header>` `<footer>`

`<section>`

`<article>`

`<div>`

Inline tags

`<strong>` `<em>` `<span>`

`<a>`

`<img>`

CSS Review

```
h1 {
    font-size: 32px;
}
```

```
body {
    background-color: black;
    font-family: serif;
    color: white;
}
```

CSS Review

Typography

color
font-family
font-size
line-height
letter-spacing

font-style
font-weight

text-align
text-transform
text-decoration

Sizing

width
height

Spacing

margin
padding

Backgrounds

background-color
background-image
background-repeat
background-size

Borders

border
border-radius

Special Selectors

:hover
:active

```
<p id="intro">
    The only intro paragraph.
</p>
```

```
<p class=“highlighted”>
    One of many possible
    highlighted paragraphs.
</p>
```

Select HTML Element

Use the element (tag) name

```
<body>
```

```
body { }
```

```
<h1>
```

```
h1 { }
```

Select by ID

Use the # sign

```
<div id="top">
```

```
div#top { }
#top { }
```

```
<ul id="nav">
```

```
ul#nav { }
#nav { }
```

Select by Class

Use a . (dot)

```
<a class="selected">
```

```
a.selected { }
.selected { }
```

```
<h1 class="special">
```

```
h1.special { }
.special { }
```

Select "descendent"

Use space between two selectors

```
<ul>                                ul li { }
  <li>Item</li>
</ul>

<h1>                                h1 em { }
  <em>Hi</em> there!
</h1>
```

CSS Selectors

Mix & Match

```
ul#nav li a { }
```

```
<ul id="nav">
  <li>
    <a href="#">Link</a>
  </li>
</ul>
```

Centering Your Website

HTML

```
<body>
  <main>
    <p>Content</p>
    ...
  </main>
</body>
```

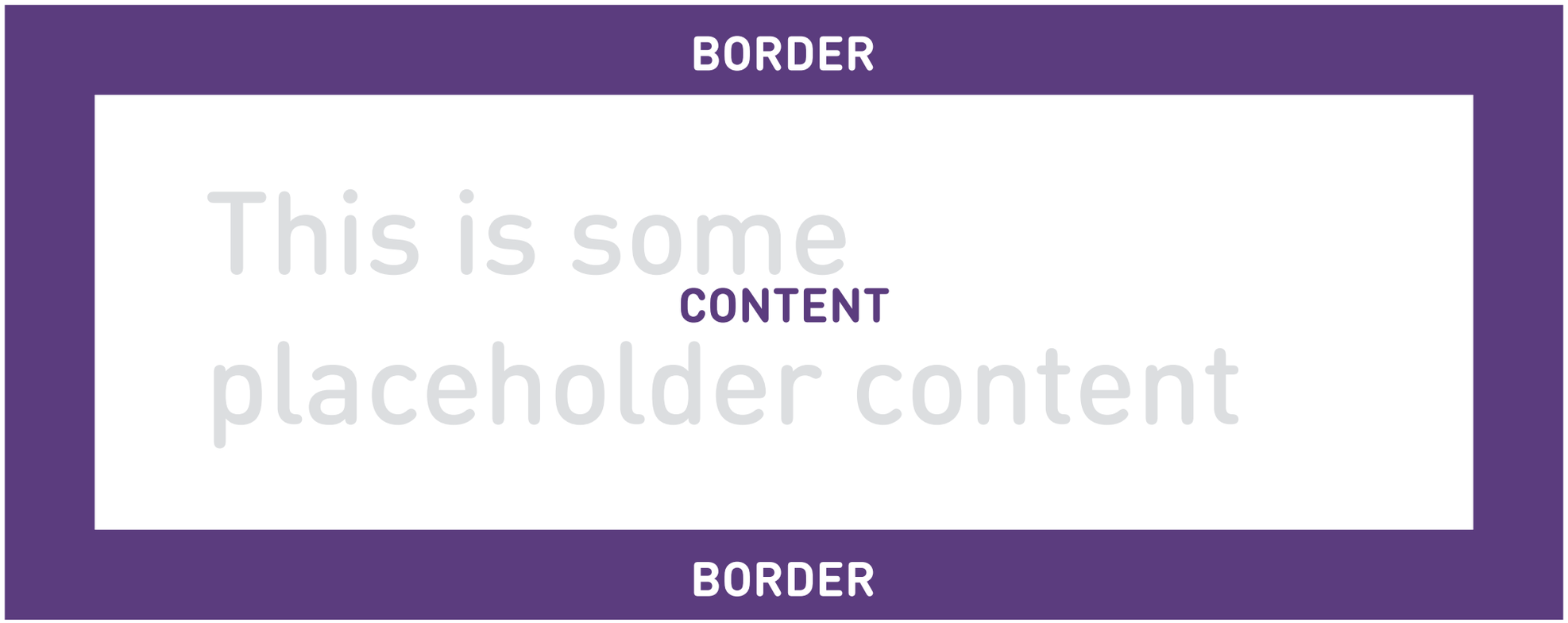
CSS

```
main {
  width: 800px;
  margin-left: auto;
  margin-right: auto;
}
```

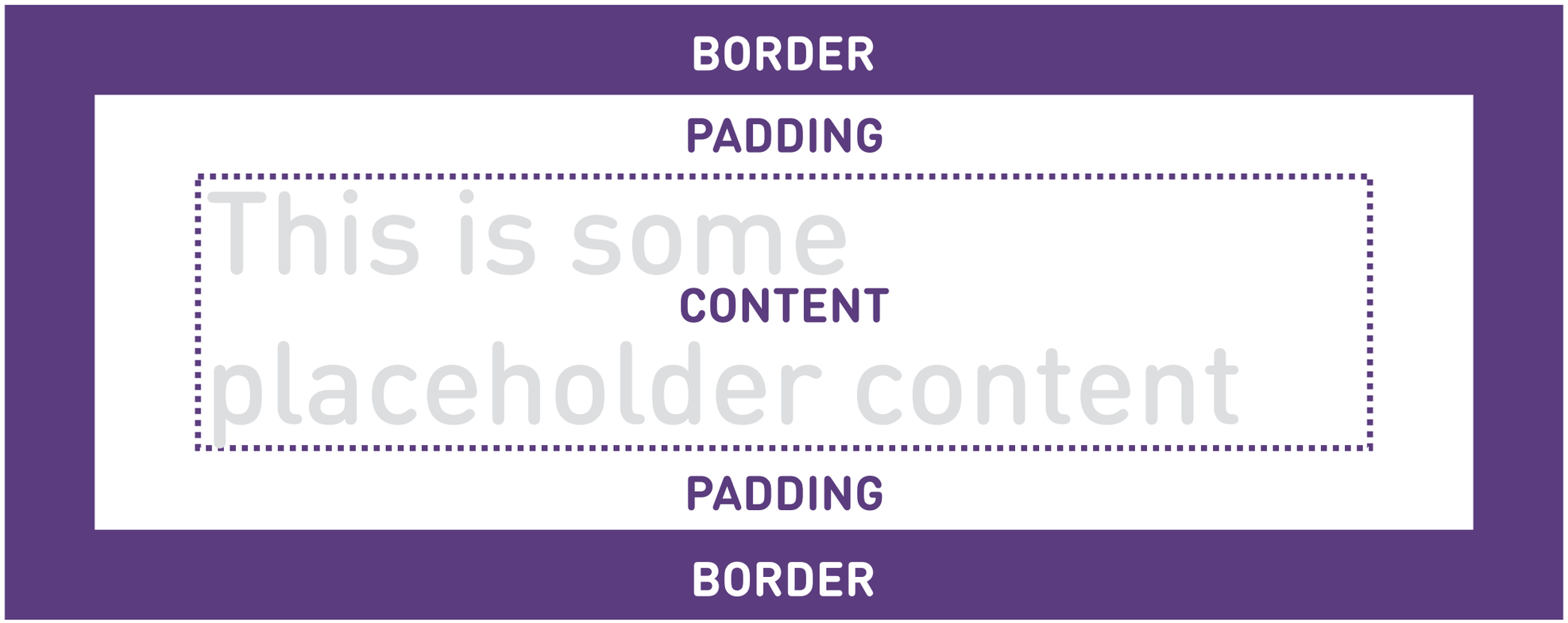
CSS
Box Model

This is some placeholder content

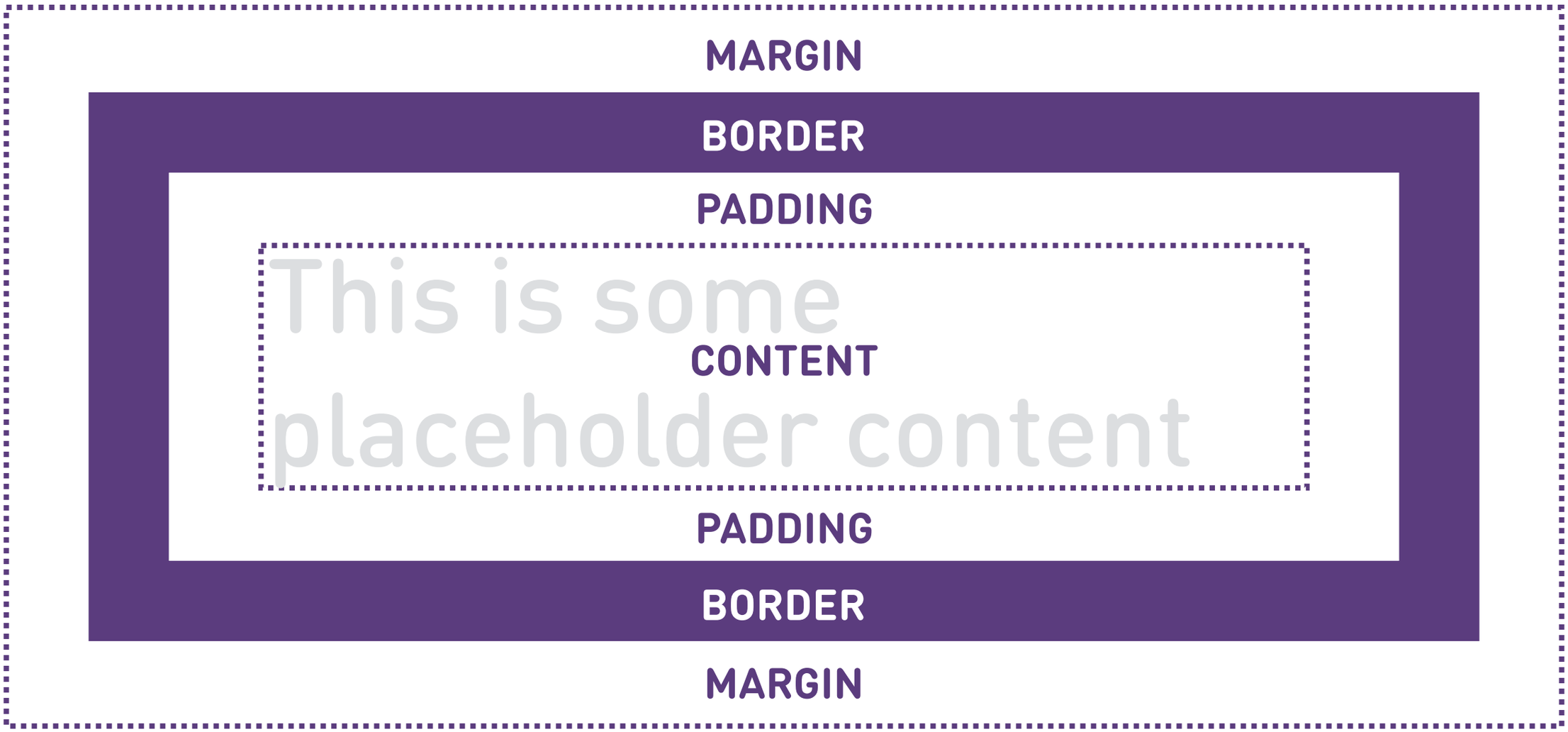
The Box Model



The Box Model



The Box Model



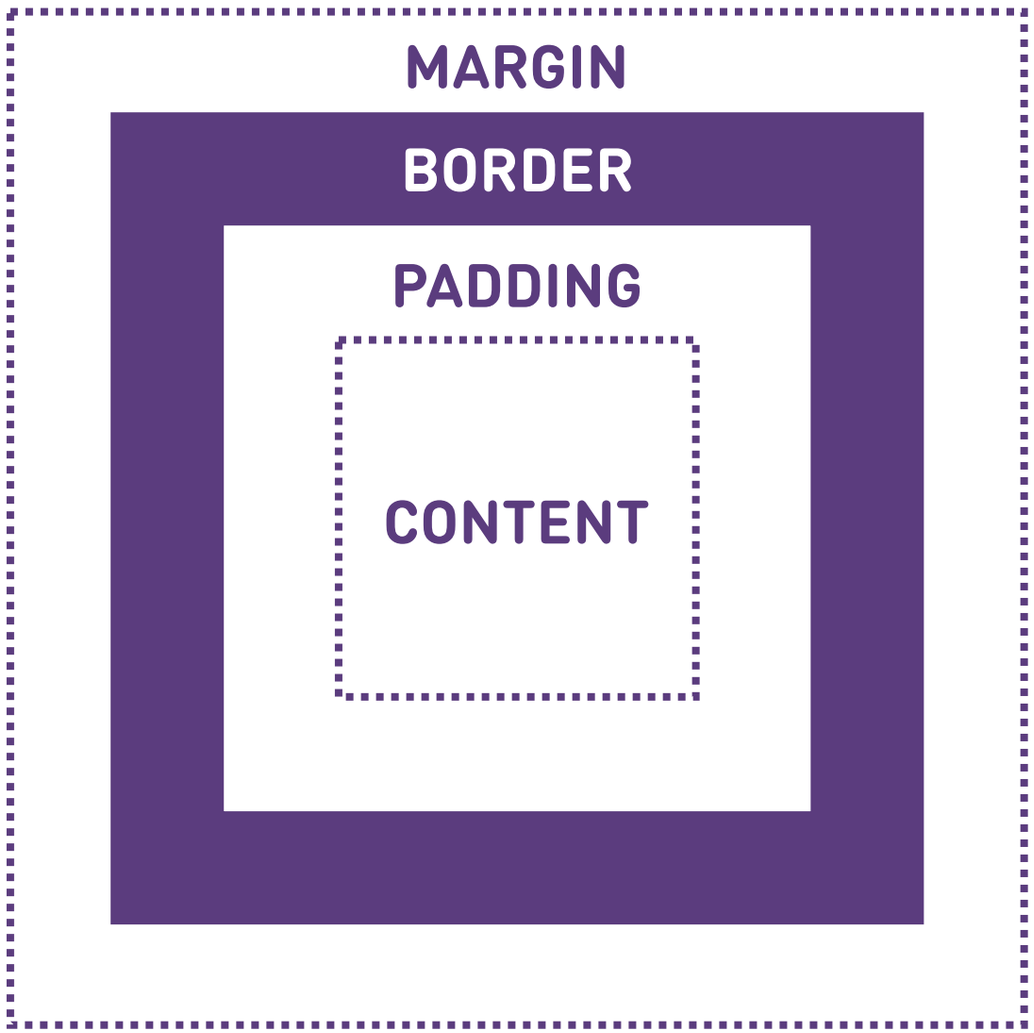
The Box Model

```
p {

    width: 100px;
    height: 100px;

    margin: 10px;
    border: 10px;
    padding: 10px;

}
```



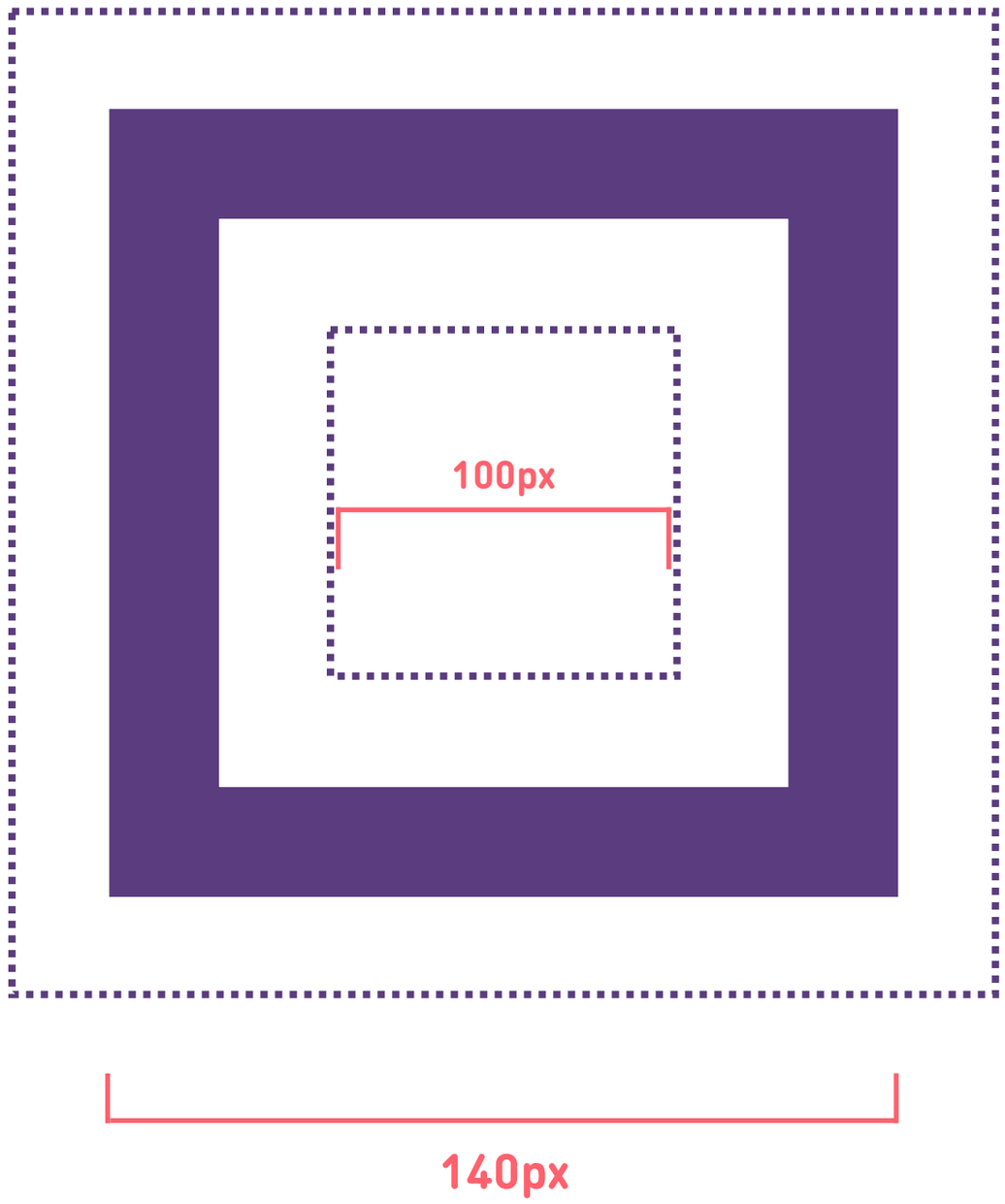
The Box Model

```
p {

    width: 100px;
    height: 100px;

    margin: 10px;
    border: 10px;
    padding: 10px;

}
```



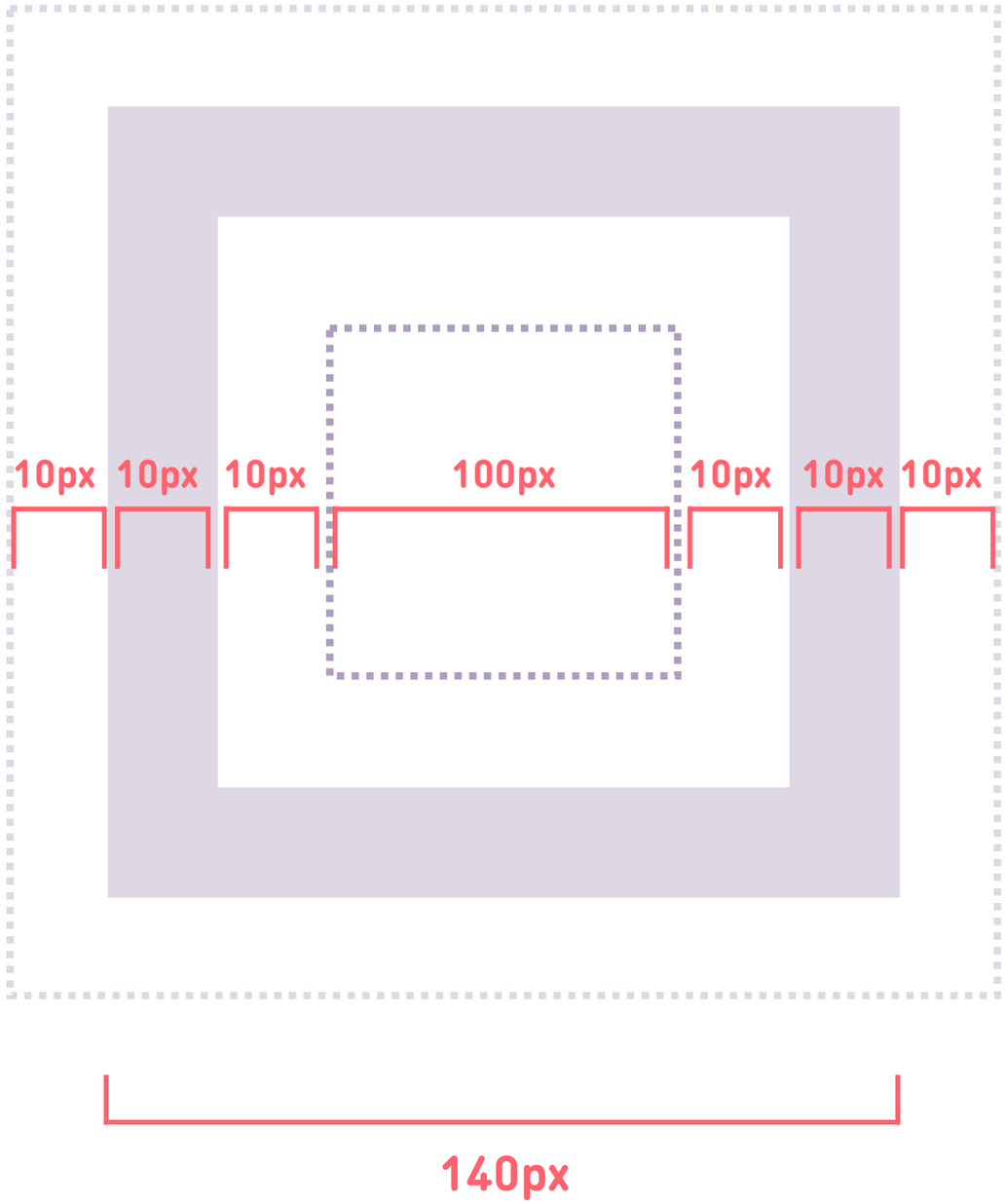
The Box Model

```
p {

    width: 100px;
    height: 100px;

    margin: 10px;
    border: 10px;
    padding: 10px;

}
```



The Box Model

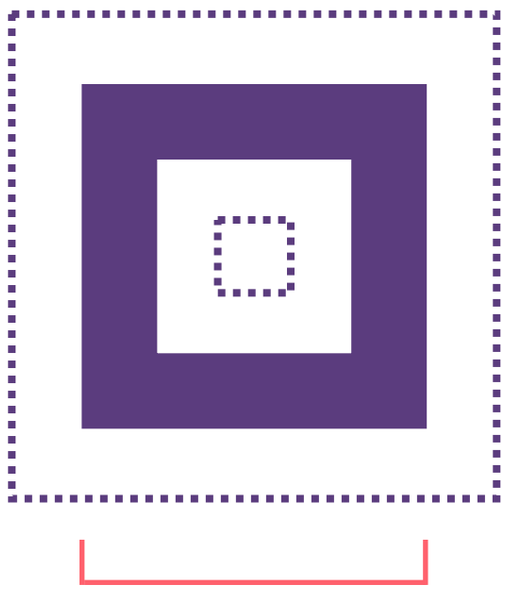
```
p {

    box-sizing: border-box;

    width: 100px;
    height: 100px;

    margin: 10px;
    border: 10px;
    padding: 10px;

}
```



Make boxes easier to size

Add this code to your CSS file:

```
* {
    box-sizing: border-box;
}
```

Positioning

What does it do?

Allows us to place elements in specific locations on the page.

Positioning methods

- `relative`: adjust placement within the flow of the page
- `absolute`: place in a specific place outside of the flow of the page
- `fixed`: place relative to the browser window, unaffected by scrolling

How it works

You set (1) the positioning method, and (2) the distance from the left/right and top/bottom in CSS.

```
div {
    position: absolute;
    left: 25px;
    top: 25px;
}
```

```
div {
    position: relative;
    left: 25px;
    top: 25px;
}
```

Repositions `div` 25px to the right, and 25px down from where it was in the flow of the page.

```
div {
    position: absolute;
    left: 25px;
    top: 25px;
}
```

Pulls `div` out of the flow of the page and positions it 25px from the left side, and 25px from the top of the page.

```
div {
    position: fixed;
    left: 25px;
    top: 25px;
}
```

Anchors `div` to 25px from the left side, and 25px from the top of the browser window, unaffected by scrolling.

Relative to container

If an element positioned with `absolute` is contained inside another element that is positioned, it will be positioned relative to that container instead of the page.

Arrange in front/behind

Set z-index to a whole number to move elements in front or behind each other. Higher numbers overlap lower numbers.

```
div {
    z-index: 5;
}
```

Flexbox

What does it do?

Allows us to set up a system for flexible columns of content on our page.

How does it work?

You define the rules for a container HTML element and the elements within it (child elements).

The browser follows the rules you set to lay everything out and adapt when the browser changes size.

The best guide/reference

CSS Tricks Guide to Flexbox is an invaluable resource:

https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Coding
based on a design

Process Overview

1. Put content in HTML
2. Add structural HTML elements
3. Style elements in CSS, one chunk at a time, from top to bottom of the page; adjust HTML as needed

1. Put Content in HTML

- Use standard base HTML code to start
- Add text and images using appropriate HTML elements: headings, paragraphs, images, lists

2. Add Structure to HTML

- Group elements that go together in structural HTML elements (`main`, `header`, `footer`, `section`, `div`)
- Add `id`s and `class`es where it seems appropriate.

3. Style Elements in CSS

- Select one element at a time. Make sure selector works (e.g., add obvious background-color) before adding rules.
- Apply one CSS property at a time to bring it closer to the design
- Compare to design, edit, repeat

Week 4
Project

Week 4 Project

Master Your Layout

Take the layout & positioning strategies we've learned this week and implement them into your website

Ideas

- Create a multi-column layout
- Create an image gallery (in a columned grid format) using Flexbox
- Find an interesting way to position your header/navigation (`fixed` position?)
- Allow elements on the page to overlap

Questions? Comments?

- Visit http://movingobjects.io/svc for class slides, code samples, resources
- Email me: scott@movingobjects.io