Front-End Web Development

Jingjie (Vincent) Zheng

JS

HTML

CSS

In a nutshell …

- HTML adds meaning to text by logically dividing it and identifying the role that texts plays on the page.
- CSS introduces styles and layouts to provide beautiful feels and looks.
- JS offers great dynamics for making and handling change of the document.

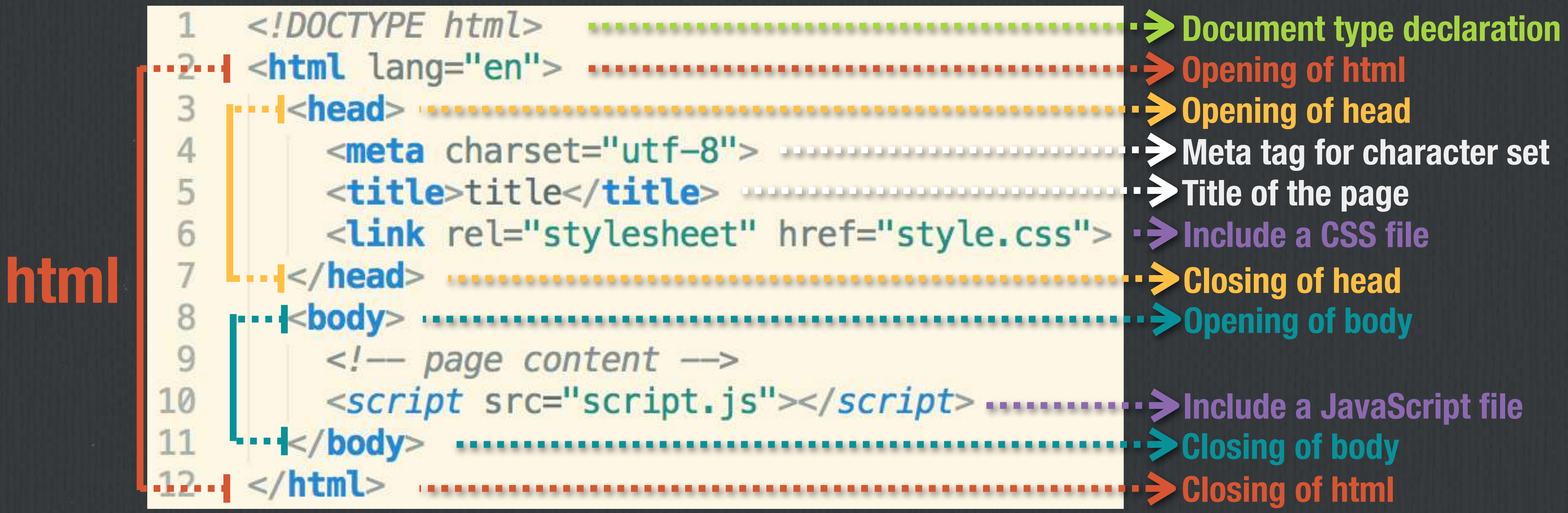
HTML

Think Structure

HTML

- Definition:
  - HyperText Markup Language
- Syntax:
  - <tag attribute="value"> content </tag>
  - Nested tags
  - Each tag has a number of attributes
- Internal model:
  - Document Object Model

HTML5 Document



```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>title</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <!-- page content -->
  <script src="script.js"></script>
</body>
</html>
```

HTML5 Tags

- Headers
  <h1>Header 1</h1>
  <h2>Header 2</h2>
  <h3>Header 3</h3>

- Paragraph
  <p>A paragraph of texts … </p>

- Horizontal Lines
  <hr>

- Line Breaks
  <br>

HTML5 Tags

- Images
  ```
  <img src=“images/apple.png”, alt=“Apple Image”>
  ```
- Tables
  ```
  <table>
  <tr><td></td><td></td></tr>
  <tr><td></td><td></td></tr>
  </table>
  ```
- Links
  ```
  <a href=“http://www.example.com”>Example</a>
  <a href=“#biography”>Go to Biography</a>
  ```

HTML5 Tags

- Lists
  - Ordered List
    <ol>
    <li>First, eat</li> <li>Second, sleep</li>
    <li>Repeat the first</li>
    </ol>
  - Unordered List
    <ul>
    <li>Eat</li> <li>Sleep</li>
    </ul>

HTML5 Tags

- Generic containers
  - <div></div>
    - block element, i.e., line breaks before and after it
    - container for virtually anything
  - <span></span>
    - inline element
    - container for text

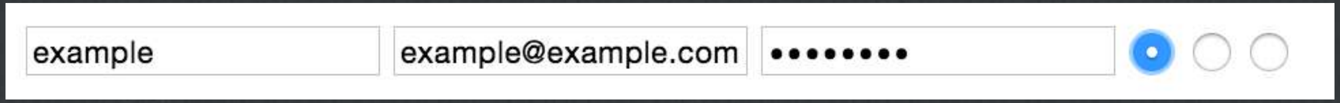
HTML5 Tags

- Common structuring tags that essentially are divs
  - <header></header>
  - <footer></footer>
  - <aside></aside>
  - <section></section>

HTML5 Tags

- Forms and inputs

  -
    ```
    <form>
      <input type="text" name="username">
      <input type="email" name="email">
      <input type="password" name="password">
      <input type="radio" name="gender" value="male">
      <input type="radio" name="gender" value="female">
      <input type="radio" name="gender" value="other">
    </form>
    ```



HTML to Forget

- ☐ Skip <table> for page layout!
- ☐ Ditch <font> for controlling the display of text.
- ☐ Don’t use the <b> and <i> tags to emphasize text.
- ☐ Don’t abuse the <br> tag.
- ☐ Skip <table> for page layout!

HTML to Remember

- Use <div> and <span> if no other tags convey the semantics.
- Use <p> for paragraphs of text.
- Use <ul> when you've got a list of several related items.
- Use <ol> to indicate steps in a process or define the order of a set of items.
- Remember to close tags except <br>, <img>, and <input>. (HTML 5)

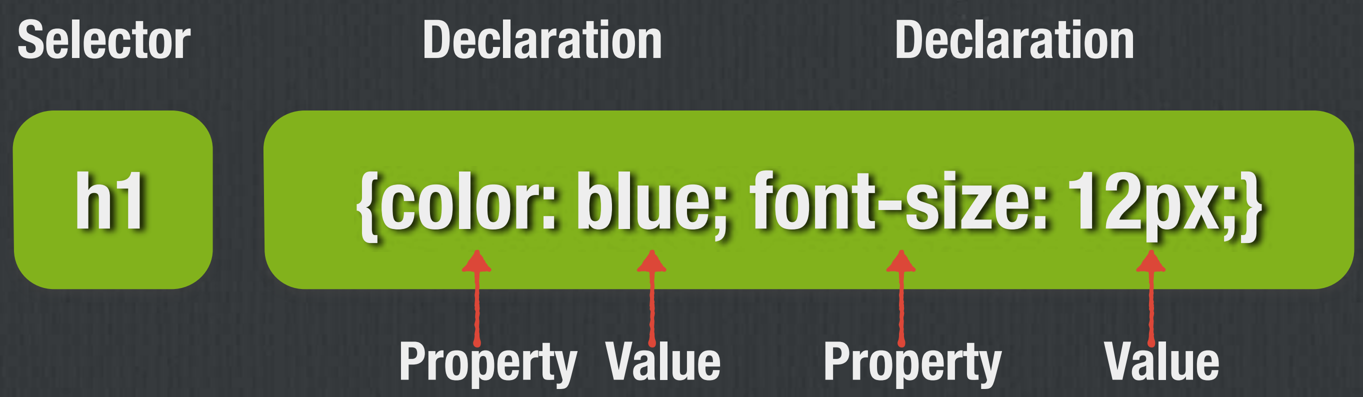
HTML to Remember

- A complete list of what to and what not to use in HTML: http://www.html-5-tutorial.com/all-html-tags.htm

CSS

Think Looks and Feels

Syntax



```
h1 {color: blue; font-size: 12px;}
```

Identify Elements

- Use selectors:
  - Tags
    p div span table h1 h2 …
  - Prefix # for selecting with an ID
    #menu #contact-list
  - Prefix . for selecting with a Class
    .contact-name .contact-photo …

Identify Elements

- Tips
  - IDs are unique; Classes are NOT unique.
  - Elements can have both.
  - CSS doesn’t care, JavaScript cares.
  - If you don’t need them, don’t use them.
- Reference: https://css-tricks.com/the-difference-between-id-and-class/

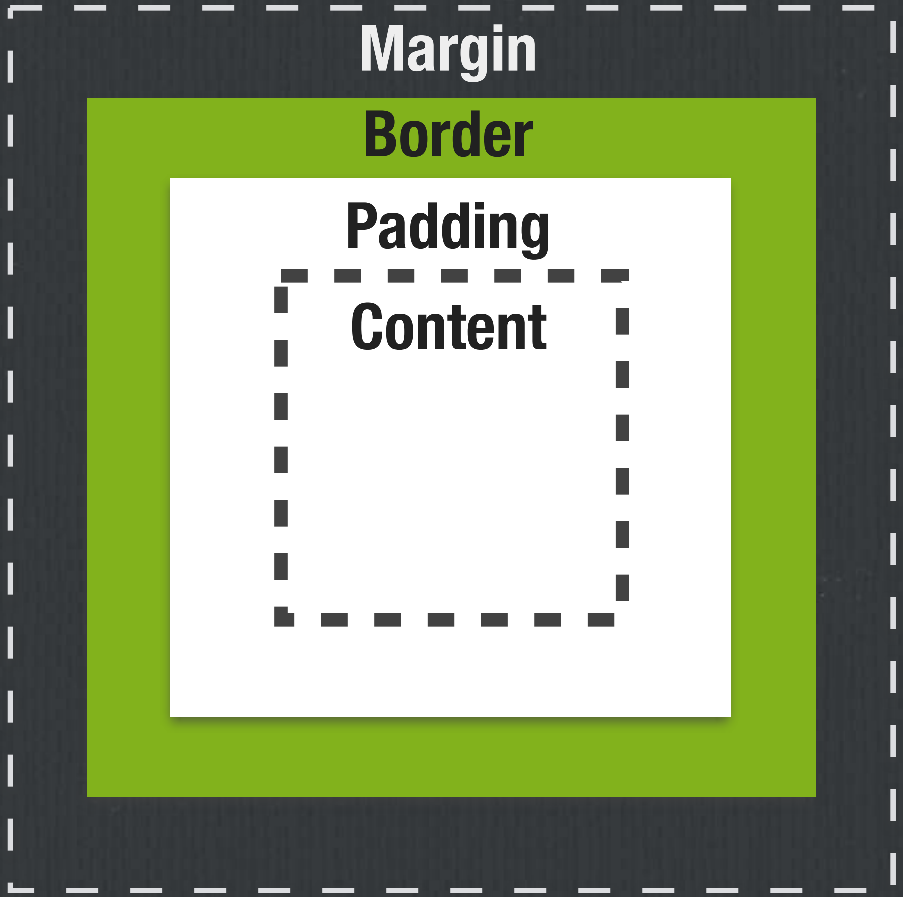
Looks and Feels

- What is CSS really about?
    - Size
    - Position
    - Layout
    - Others: colour, typography, animation, etc.

CSS Reset

- Clear browser default behaviours
- Reference: http://meyerweb.com/eric/tools/css/reset/

Size - Box Model

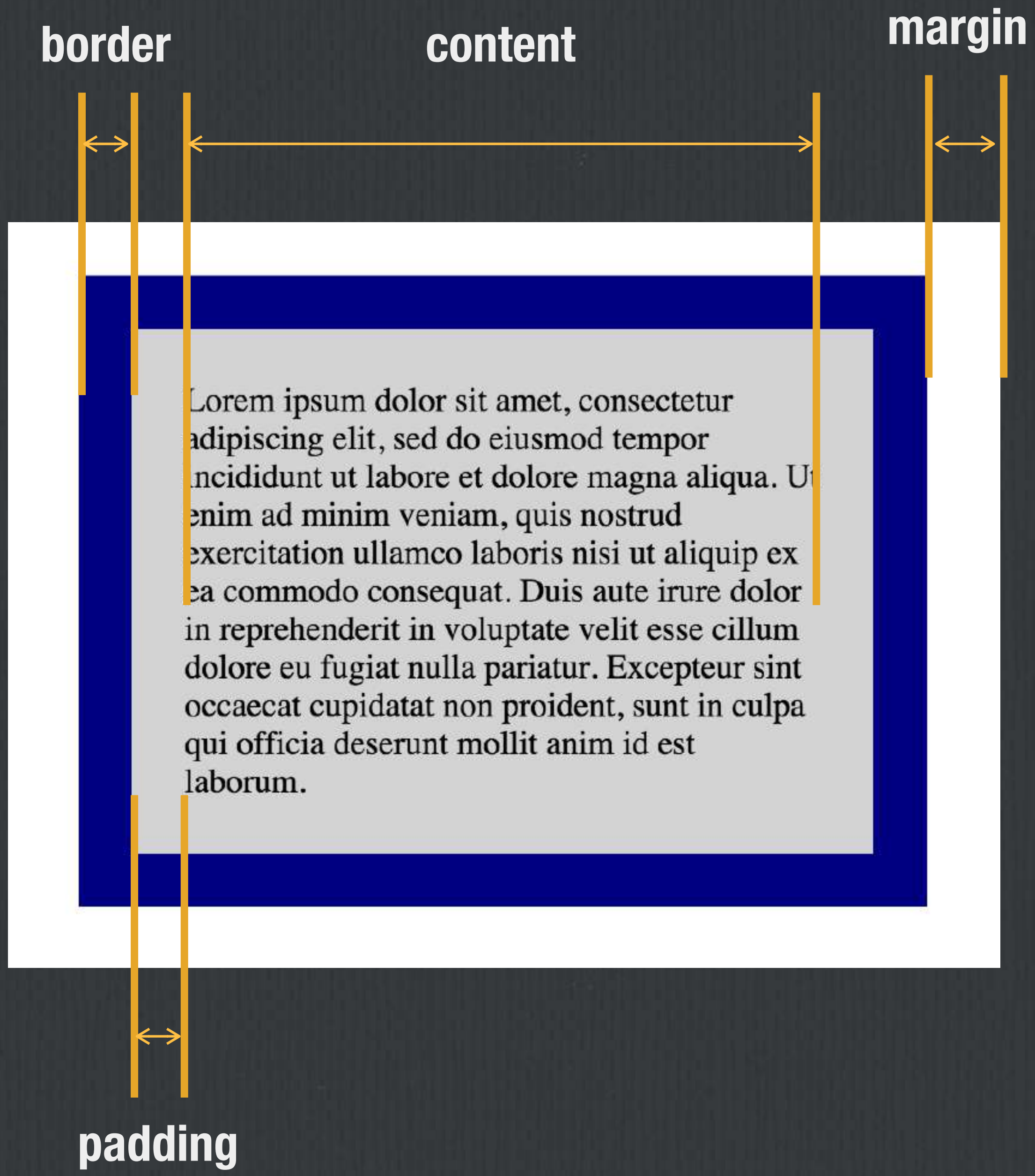


- By default,
  width = width of content
  height = height of content

- Using
  `* { box-sizing: border-box; }`
  sets
    width =
      width of content + padding
    height =
      height of content + padding

Size - Box Model (demo)

```
div {
    background-color: lightgrey;
    width: 300px;
    padding: 25px;
    border: 25px solid navy;
    margin: 25px;
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Size - Measurement Units

- Pixels
  e.g. 5px

- Percent (relative to the containing block)
  e.g. 50%

Position - display (demo)

- display: inline; /* e.g. span */
  - Ignore top & bottom margins and paddings
  - Cannot have a width or height set
  - Allow other elements to sit to their left and right

- display: block; /* e.g. div */
  - Force a line break before and after the element

- display: inline-block; /* A block that does not force line breaks */
  - Respect top & bottom margins and paddings
  - Can have a width and height set
  - Allow other elements to sit to their left and right

Position - display

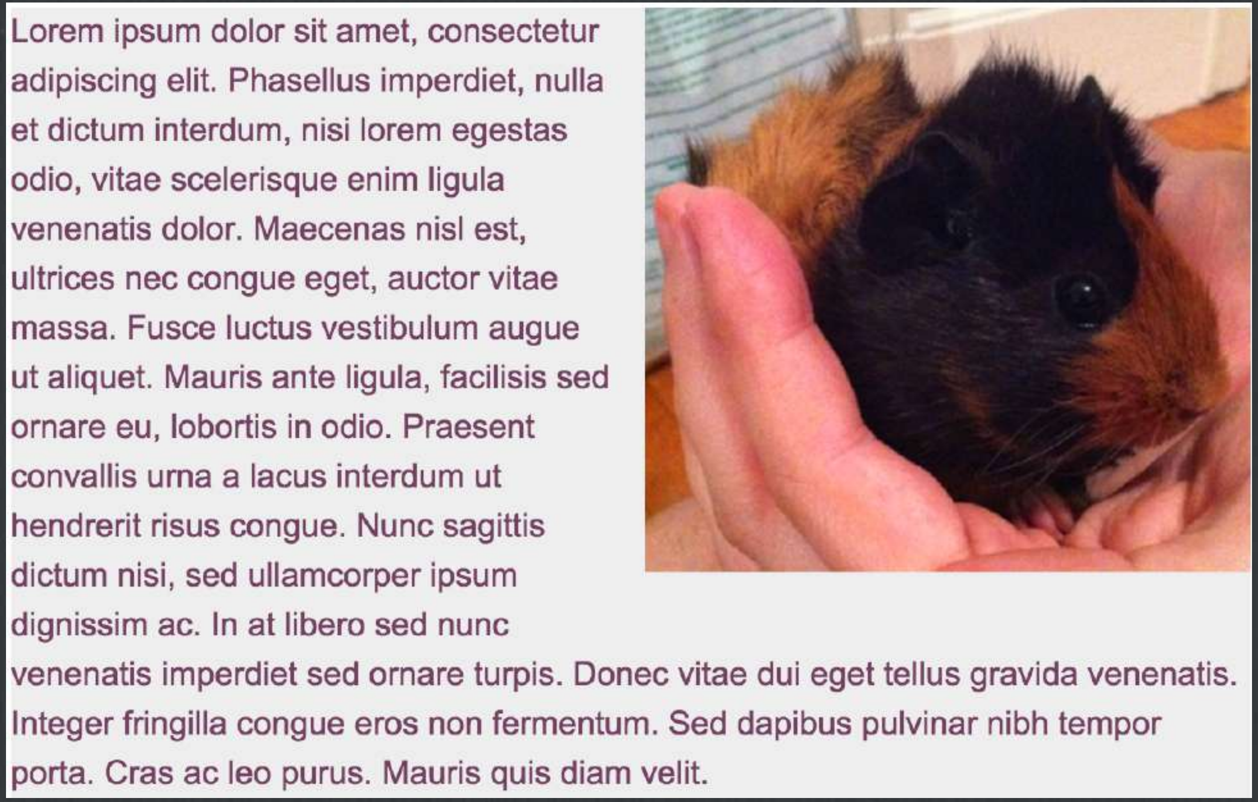
- display: none;   /* renders as if it does not exist */
- visibility: hidden;   /* takes the place, but not showing */

Position - position (demo)

- The “position” property:
  - position: static;
    - Default position
  - position: relative;
    - Relative to its default position
  - position: fixed;
    - Relative to the viewport
  - position: absolute;
    - Relative to its nearest positioned ancestor
      (“positioned” <=> Anything but static)

Position - float

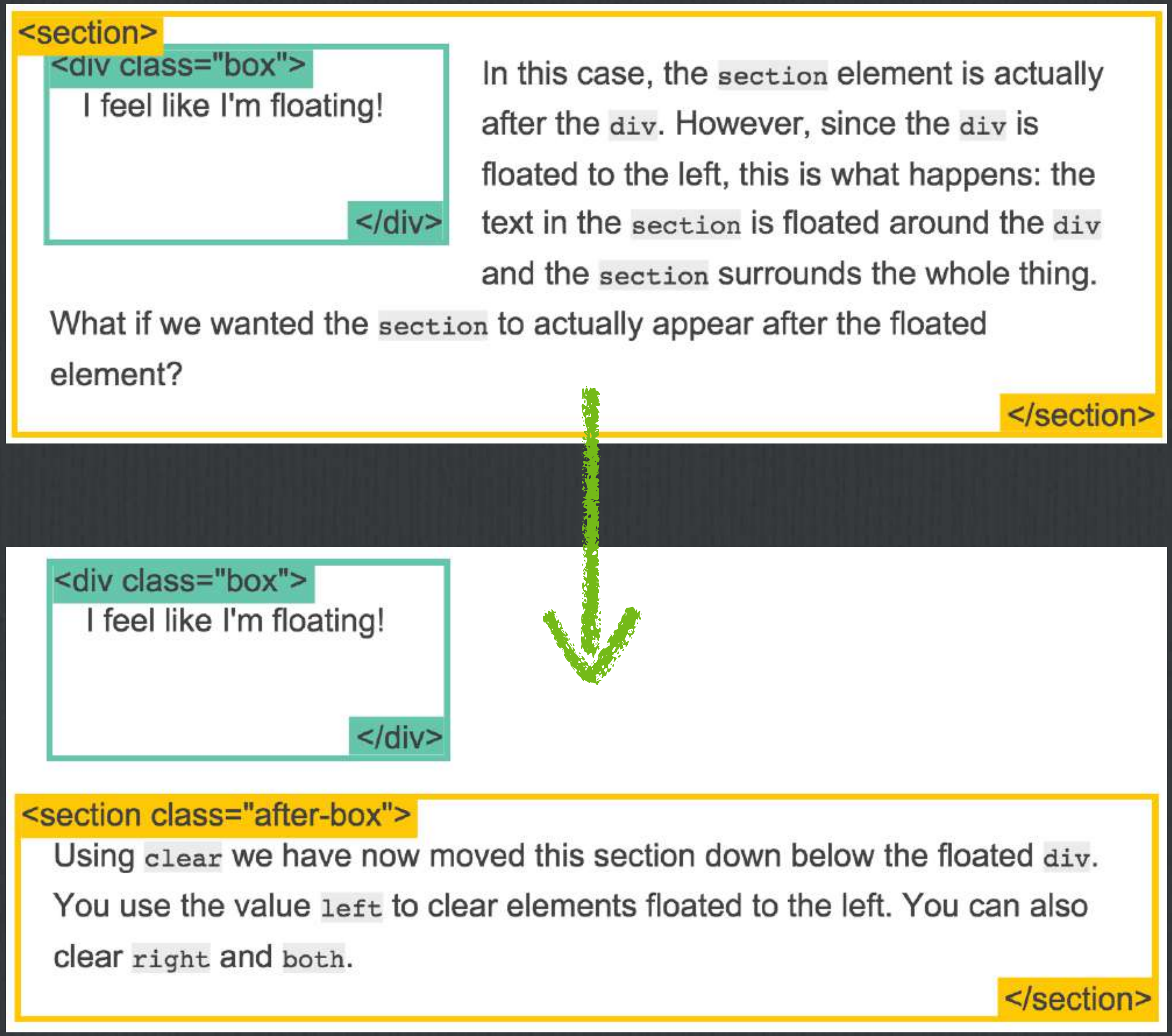
```
img {
  float: right;
  margin: 0 0 1em 1em;
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.

Position - clear

```
.after-box {
  clear: left;
}
```



<section>
<div class="box">
I feel like I'm floating!
</div>

In this case, the `section` element is actually after the `div`. However, since the `div` is floated to the left, this is what happens: the text in the `section` is floated around the `div` and the `section` surrounds the whole thing.

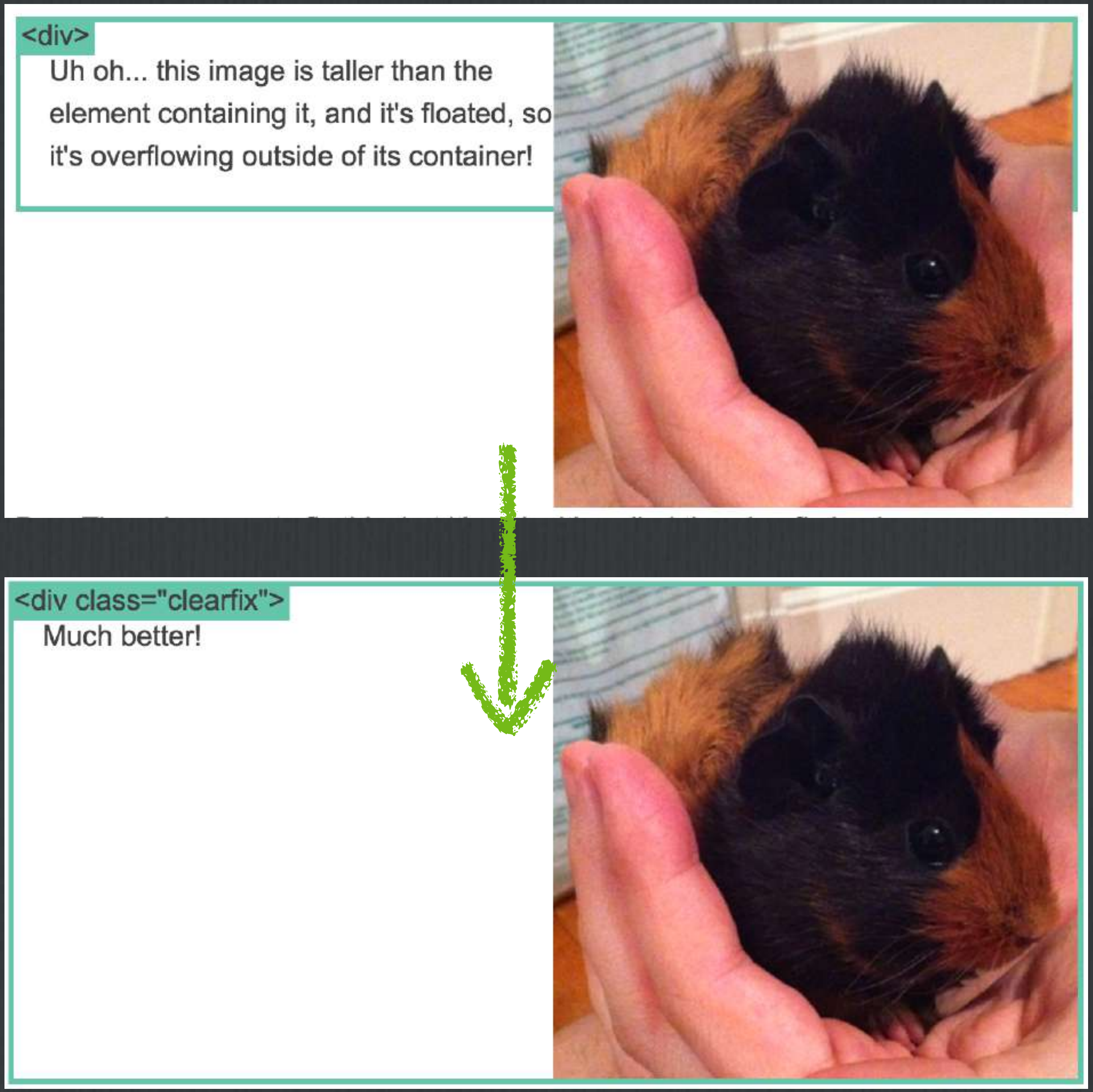
What if we wanted the `section` to actually appear after the floated element?

</section>
<div class="box">
I feel like I'm floating!
</div>
<section class="after-box">

Using `clear` we have now moved this section down below the floated `div`. You use the value `left` to clear elements floated to the left. You can also clear `right` and `both`.

Position - clearfix (demo)

```
img {
  float: right;
}
.clearfix {
  overflow: auto;
}
```



Uh oh... this image is taller than the element containing it, and it's floated, so it's overflowing outside of its container!

Much better!

Inline-Block Layout

- Demo

Media Query

- Useful for responsive design
- References:
  http://learnlayout.com/media-queries.html
  https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Media_queries

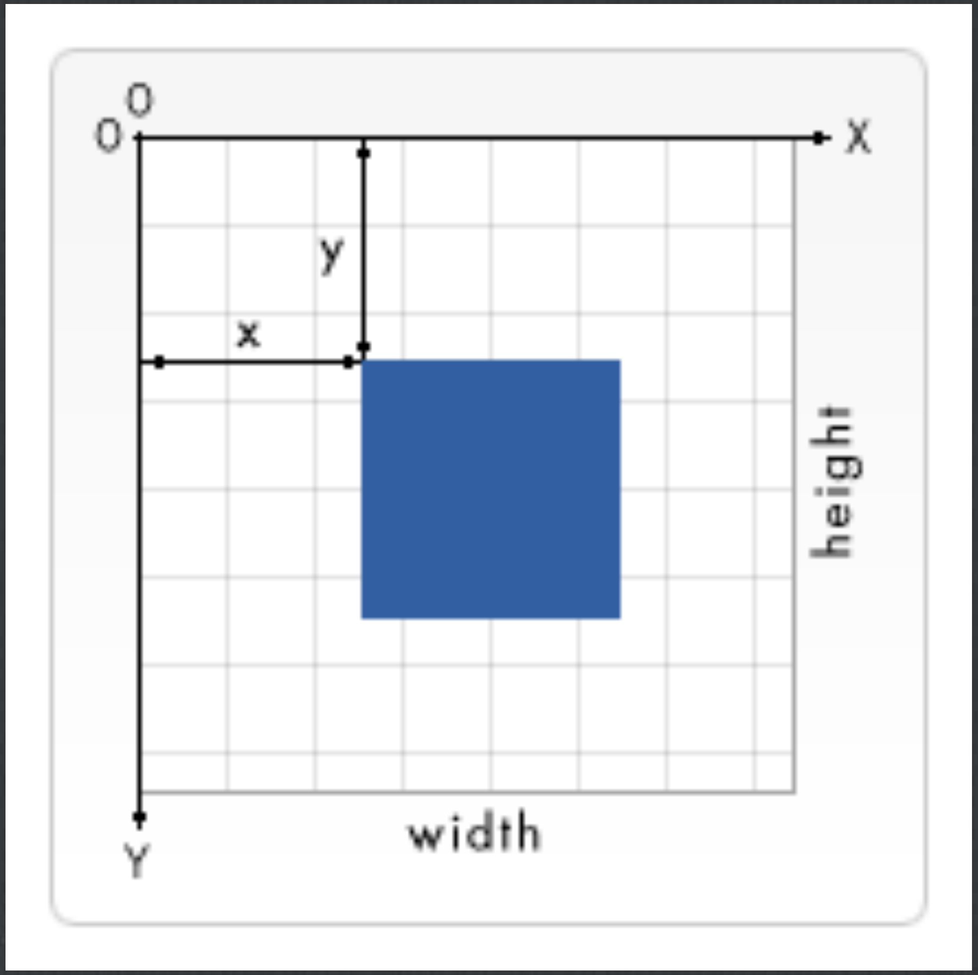
Resources

- http://learnlayout.com/
- http://www.w3schools.com/css/default.asp

JavaScript

Think Interaction

Canvas



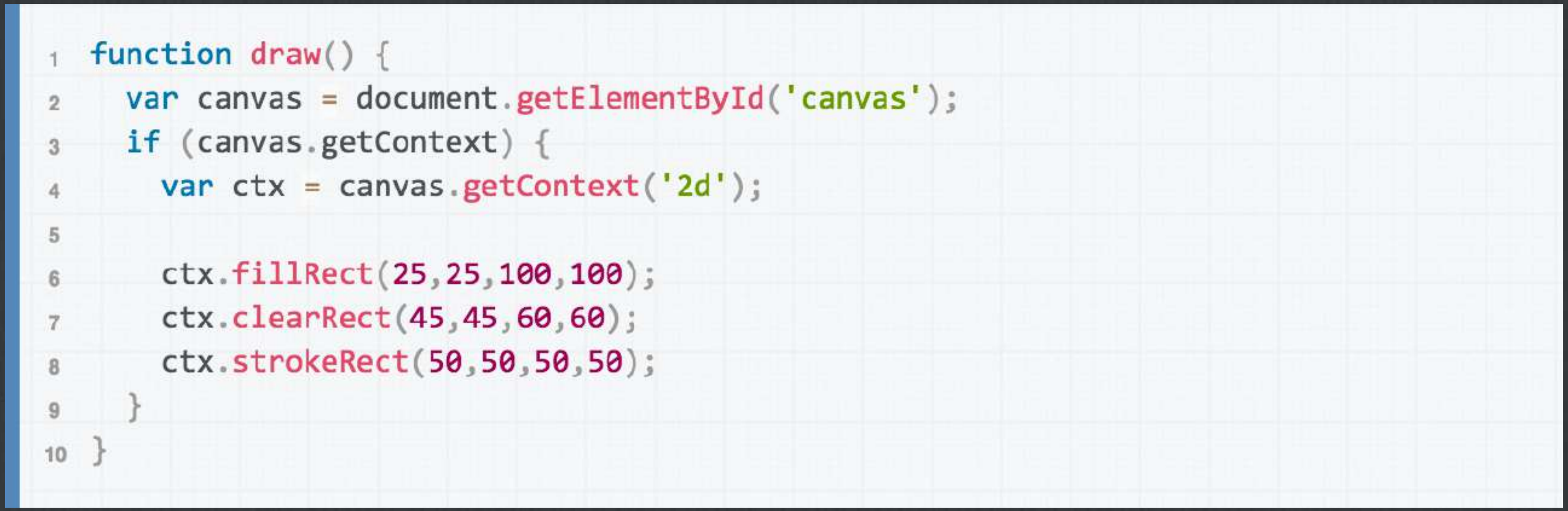
- Canvas is an element where you can draw
- Origin is at top left corner
  Positive x is to the right
  Positive y is to the bottom

Canvas Setup (demo)

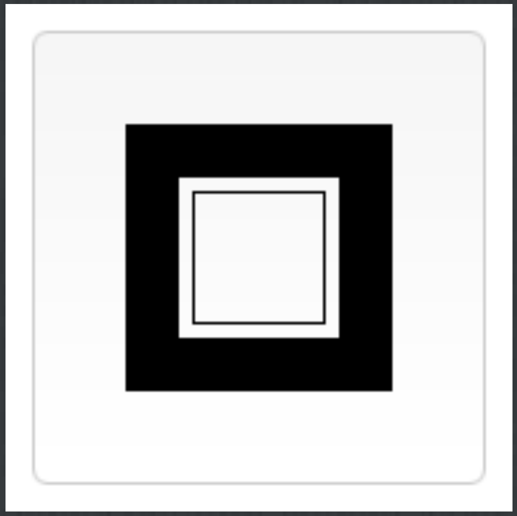
```html
<!DOCTYPE html>
<html>
<head>
<title>Canvas Simple Example</title>
</head>
<body>
<!-- Start your code here -->
  <canvas width="150" height="150" id="canvas"></canvas>
<!-- End your code here -->
</body>
</html>
```

```javascript
window.addEventListener("load", function() {
  function draw() {
      var canvas = document.getElementById("canvas");
      if (canvas.getContext) {
        var ctx = canvas.getContext("2d");

        ctx.fillStyle = "rgb(200,0,0)";
        ctx.fillRect (10, 10, 55, 50);

        ctx.fillStyle = "rgba(0, 0, 200, 0.5)";
        ctx.fillRect (30, 30, 55, 50);
      }
  }
  draw();
});
```

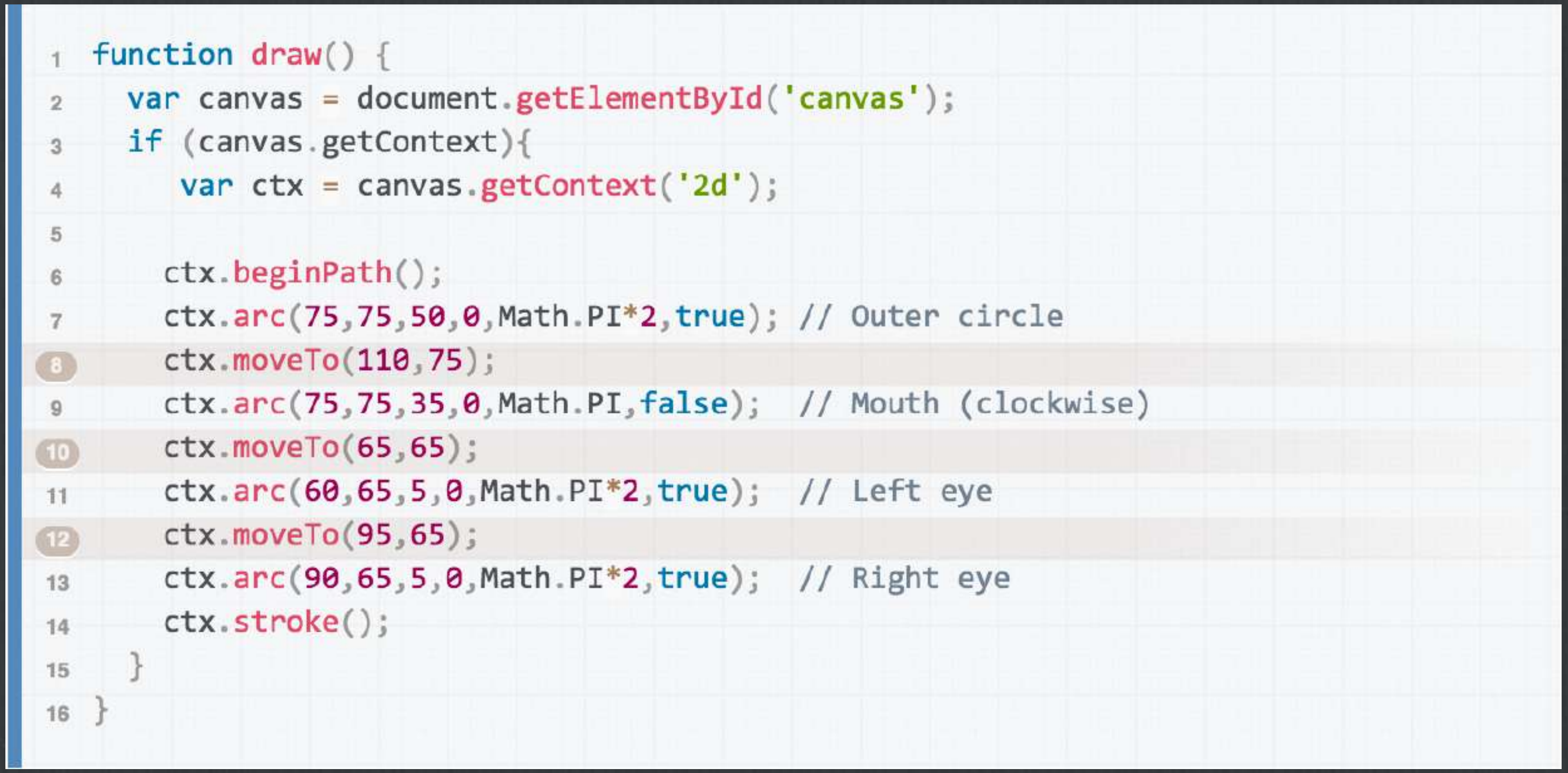
Canvas Example (1/3)



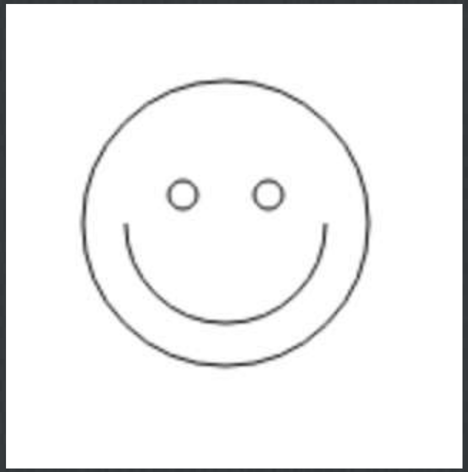
```
function draw() {
  var canvas = document.getElementById('canvas');
  if (canvas.getContext) {
    var ctx = canvas.getContext('2d');

    ctx.fillRect(25,25,100,100);
    ctx.clearRect(45,45,60,60);
    ctx.strokeRect(50,50,50,50);
  }
}
```



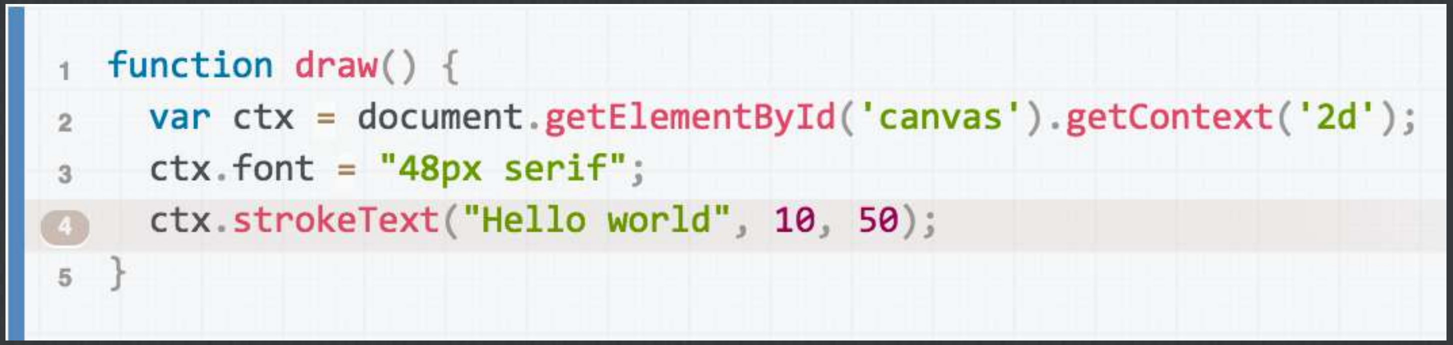
Canvas Example (2/3)



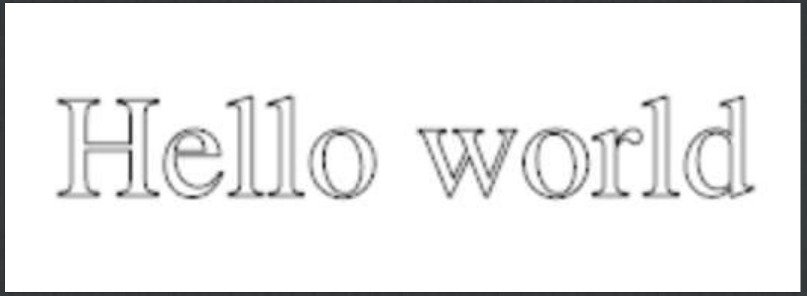
```
function draw() {
  var canvas = document.getElementById('canvas');
  if (canvas.getContext){
     var ctx = canvas.getContext('2d');

    ctx.beginPath();
    ctx.arc(75,75,50,0,Math.PI*2,true); // Outer circle
    ctx.moveTo(110,75);
    ctx.arc(75,75,35,0,Math.PI,false);  // Mouth (clockwise)
    ctx.moveTo(65,65);
    ctx.arc(60,65,5,0,Math.PI*2,true);  // Left eye
    ctx.moveTo(95,65);
    ctx.arc(90,65,5,0,Math.PI*2,true);  // Right eye
    ctx.stroke();
  }
}
```



Canvas Example (3/3)



```
function draw() {
  var ctx = document.getElementById('canvas').getContext('2d');
  ctx.font = "48px serif";
  ctx.strokeText("Hello world", 10, 50);
}
```



Canvas

- Reference: https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial

JavaScript

- Reference: https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_re-introduction_to_JavaScript

JavaScript

- Todo List Demo

Other Things That Matter

- Chrome Developer Tool
- Editors

Chrome Developer Tool

- option+command+i or Right Click + Inspect Element.
- https://developer.chrome.com/devtools.
- Tips
  - Use device mode to do responsive design.
  - Use hard refresh to get around cache.

Editors for Web Development

- Sublime Text
  http://www.sublimetext.com/

- Atom
  https://atom.io/

- Visual Studio Code
  https://code.visualstudio.com/

- WebStorm
  https://www.jetbrains.com/webstorm/