ECS 162
Web Programming

4/8

My office hour

- This week only, after class on Wds, not today.
- Office hours are for administrative issues. Lab drop-in hour are best for questions about the assignments or material.

Media Queries

- Computer, tablet and phone have different layouts.
- Use media queries to switch. Like an “if” statement.

```
/* if width is > 480px, layout horizontally */
@media (min-width: 480px) {
        div.bird {
                display: flex;
                flex-direction: row;
        }
}
```

“Mobile First”

- Css file usually first contains properties for mobile view.
- Below that, media queries for other views contain all properties that overwrite phone values.
- Why is it organized in this way?

“Mobile First”

- Css file usually first contains properties for mobile view.
- Below that, media queries for other views contain all properties that overwrite phone values.
- Why is it organized in this way?

Because on a phone things load slowly. Get the phone css there as fast as possible.

Setup details

```
<meta name="viewport" content="width=device-width”>
```

- Tell browser content should fill viewport (the part of the window available to display stuff). You’ll notice problems in the Chrome emulator if this is missing. Goes in html <head>

```
body {    min-height: 100vh; }
```

- Needed to make sure body goes all the way down to bottom of viewport. Goes in css.

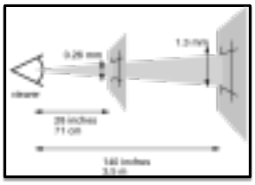
Lengths in CSS

- □ What's a pixel?

  On a device, one of the colored dots making up the image.
- □ We'll call this a device pixel; it's a piece of hardware.
- □ This phone has the same width in device pixels as my laptop.
- □ Also, you hold it close to your face.
- □ So is 2px on a phone a lot smaller than 2px on my laptop?

Lengths in CSS

- □ px in CSS refers to reference pixels.
- □ According to the W3C, a reference pixel is the visual angle subtended by a pixel on a 96dpi device held 28 inches away from the eye.



Every device has it's own dpi and expected distance from the eye. Browser automatically sets pixels size to equal this angle.

Targeting viewport size

- □ Rule of thumb: let the browser handle sizing!
  - ◘ Specify image sizes in reference pixels, not inches or cm.
- □ Let browser handle font sizes
  - ◘ The browser's default font size should be readable from the expected viewing distance.
  - ◘ In css, this is "medium"
  - ◘ Can also use "xx-small" to "xx-large"
  - ◘ 1.5 em is 1.5 times larger than the medium font
  - ◘ Use css font-size property to fine tune font sizes.

Dynamic Web pages

- □ So far we respond to change in viewport width, but we don't do anything when the user pushes the "Read more" button.
- □ Could always request a new Web page from the server when button is pushed. What is the problem with that?

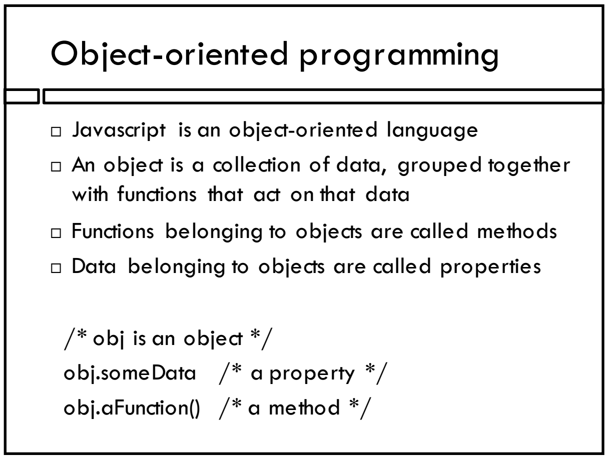
Dynamic Web pages

- □ So far we respond to change in viewport width, but we don't do anything when the user pushes the "Read more" button.
- □ Could always request a new Web page from the server when button is pushed. What is the problem with that?
  - ◘ it is slow, and
  - ◘ it makes a flash.
- □ Instead, we make dynamic pages with Javascript.

Onclick functions

- □ We ask the browser to call a Javascript function when an element is clicked by giving it an onclick attribute.
- □ Like other attributes (src, id, class) we provide it in the start tag.

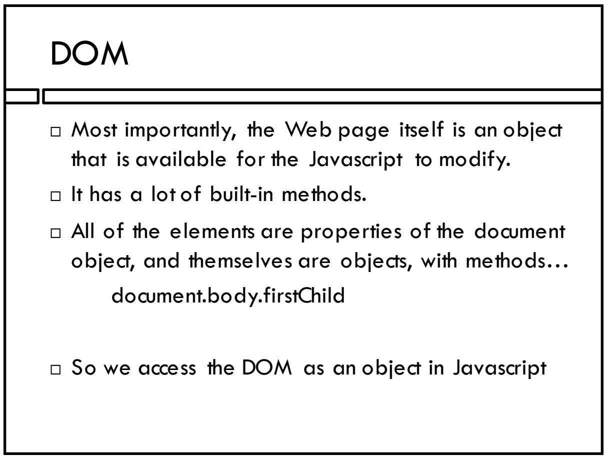
```
<div class="car" onclick="disappear('tesla')"
id="tesla">
```



Object-oriented programming

- Javascript is an object-oriented language
- An object is a collection of data, grouped together with functions that act on that data
- Functions belonging to objects are called methods
- Data belonging to objects are called properties

```
/* obj is an object */
obj.someData    /* a property */
obj.aFunction()   /* a method */
```

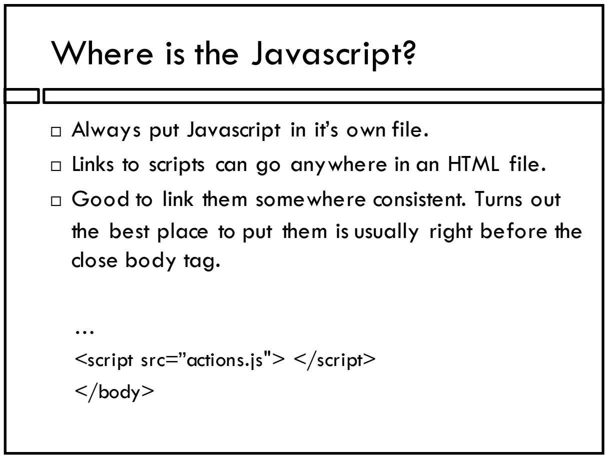


DOM

- Most importantly, the Web page itself is an object that is available for the Javascript to modify.
- It has a lot of built-in methods.
- All of the elements are properties of the document object, and themselves are objects, with methods…

```
    document.body.firstChild
```

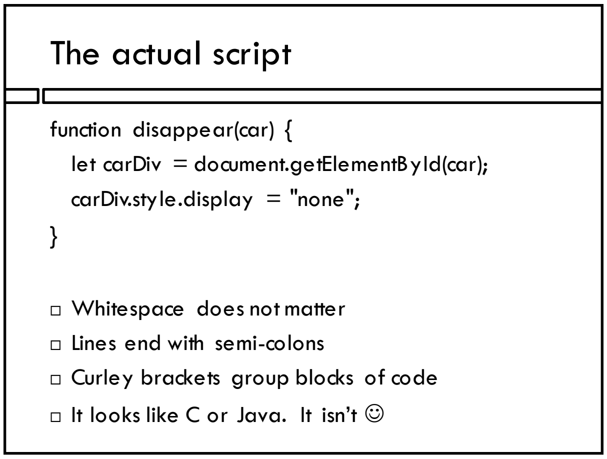
- So we access the DOM as an object in Javascript



Where is the Javascript?

- Always put Javascript in it's own file.
- Links to scripts can go anywhere in an HTML file.
- Good to link them somewhere consistent. Turns out the best place to put them is usually right before the close body tag.

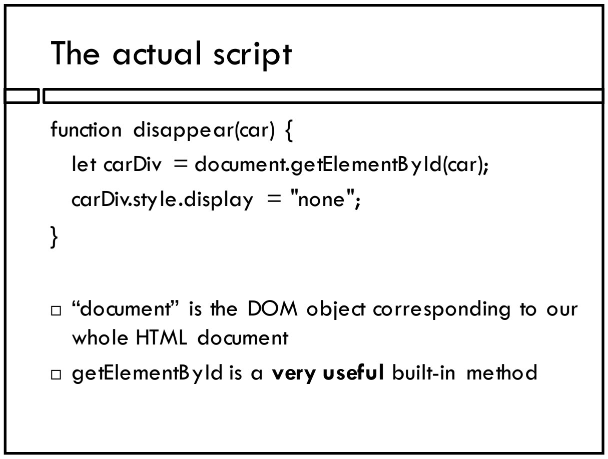
```
…
<script src="actions.js"> </script>
</body>
```



The actual script

```
function disappear(car) {
  let carDiv = document.getElementById(car);
  carDiv.style.display = "none";
}
```

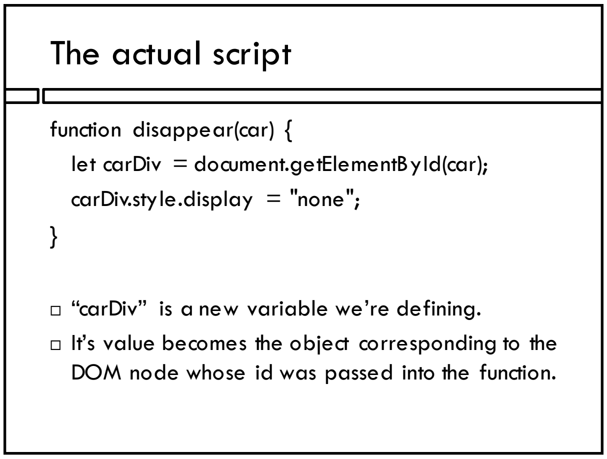
- Whitespace does not matter
- Lines end with semi-colons
- Curley brackets group blocks of code
- It looks like C or Java. It isn't ☺



The actual script

```
function disappear(car) {
  let carDiv = document.getElementById(car);
  carDiv.style.display = "none";
}
```

- "document" is the DOM object corresponding to our whole HTML document
- getElementById is a **very useful** built-in method



The actual script

```
function disappear(car) {
  let carDiv = document.getElementById(car);
  carDiv.style.display = "none";
}
```

- "carDiv" is a new variable we're defining.
- It's value becomes the object corresponding to the DOM node whose id was passed into the function.

The actual script

```
function disappear(car) {
   let carDiv = document.getElementById(car);
   carDiv.style.display = "none";
}
```

- Every DOM object has a property “style” which contains its CSS.
- display is one of its properties, whose value might be block, inline, flex, or…
- The none display value means the DOM object does not go into the layout at all.

Display none

- This is useful for elements that move between computer/tablet/phone views.
- If possible, organize the HTML so that flexbox will handle changes in layout
- Sometimes this is not possible (eg. is the green corridor image the sibling of both “read-more” paragraphs, or just the second one?).
- In that case you could have two img elements for the same image, one always with display: none.

The actual script

```
function disappear(car) {
   var carDiv = document.getElementById(car);
   carDiv.style.display = "none";
}
```

- Sadly, sometimes property names in Javascript don’t match the corresponding CSS keywords. Always check!

Javascript runs in the browser

- The Javascript code is run by the browser while executing the HTML file, as soon as it sees the <script> tag.
- But the images don’t disappear when the HTML file is executed – not until one is clicked. Why?

Javascript runs in the browser

- The Javascript code is run by the browser while executing the HTML file, as soon as it sees the <script> tag.
- But the images don’t disappear when the HTML file is executed – not until one is clicked. Why?
- This code just **defines a function;** the function is not run until it is called.
- The function is run when the image is clicked

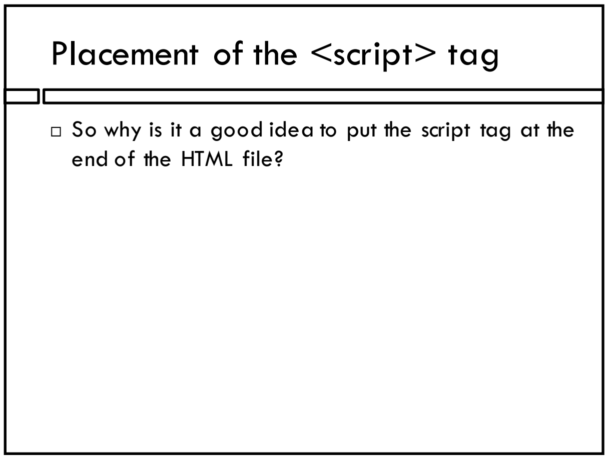
```
    onclick=”disappear(spark)"
```

Javascript outside of functions

- Does get run immediately when the page is loaded!

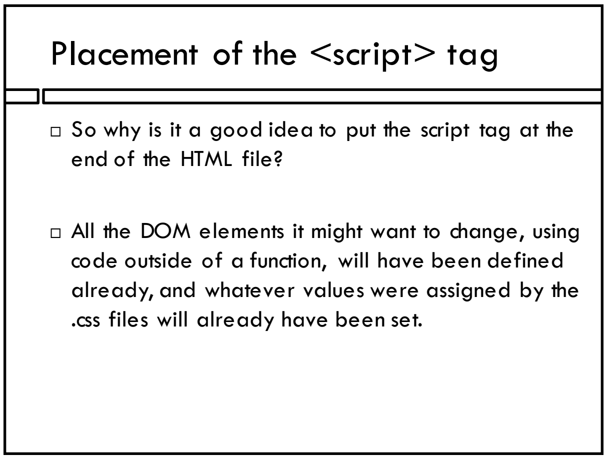
```
var demoPgh = document.getElementById("theText");
demoPgh.style.color = “red”;
```

- demoPgh is a new variable containing a paragraph
- The color property is changed right after the page is loaded



Placement of the <script> tag

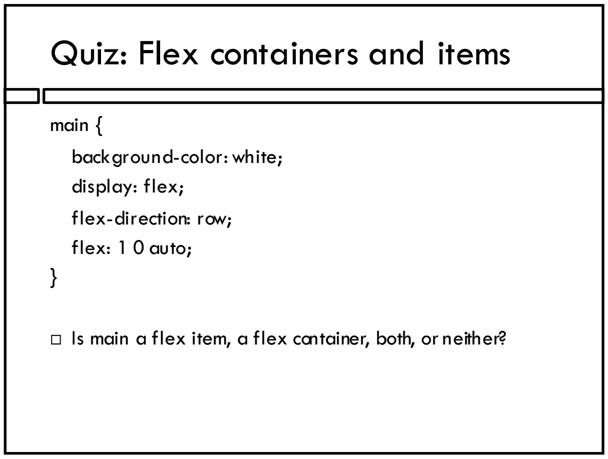
- So why is it a good idea to put the script tag at the end of the HTML file?



Placement of the <script> tag

- So why is it a good idea to put the script tag at the end of the HTML file?

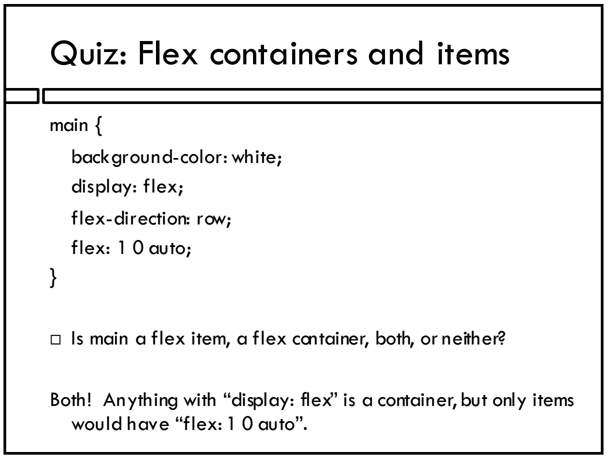
- All the DOM elements it might want to change, using code outside of a function, will have been defined already, and whatever values were assigned by the .css files will already have been set.



Quiz: Flex containers and items

```
main {
   background-color: white;
   display: flex;
   flex-direction: row;
   flex: 1 0 auto;
}
```

- Is main a flex item, a flex container, both, or neither?

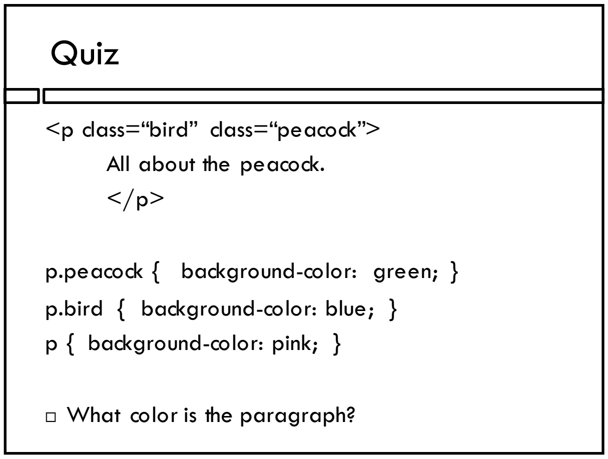


Quiz: Flex containers and items

```
main {
   background-color: white;
   display: flex;
   flex-direction: row;
   flex: 1 0 auto;
}
```

- Is main a flex item, a flex container, both, or neither?

Both! Anything with “display: flex” is a container, but only items would have “flex: 1 0 auto”.

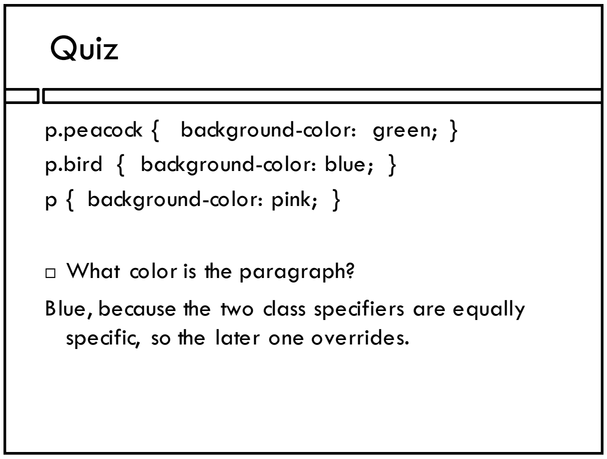


Quiz

```
<p class=“bird”  class=“peacock”>
        All about the peacock.
        </p>

p.peacock {   background-color:  green; }
p.bird  {  background-color: blue;  }
p {  background-color: pink;  }
```

- What color is the paragraph?



Quiz

```
p.peacock {   background-color:  green; }
p.bird  {  background-color: blue;  }
p {  background-color: pink;  }
```

- What color is the paragraph?

Blue, because the two class specifiers are equally specific, so the later one overrides.