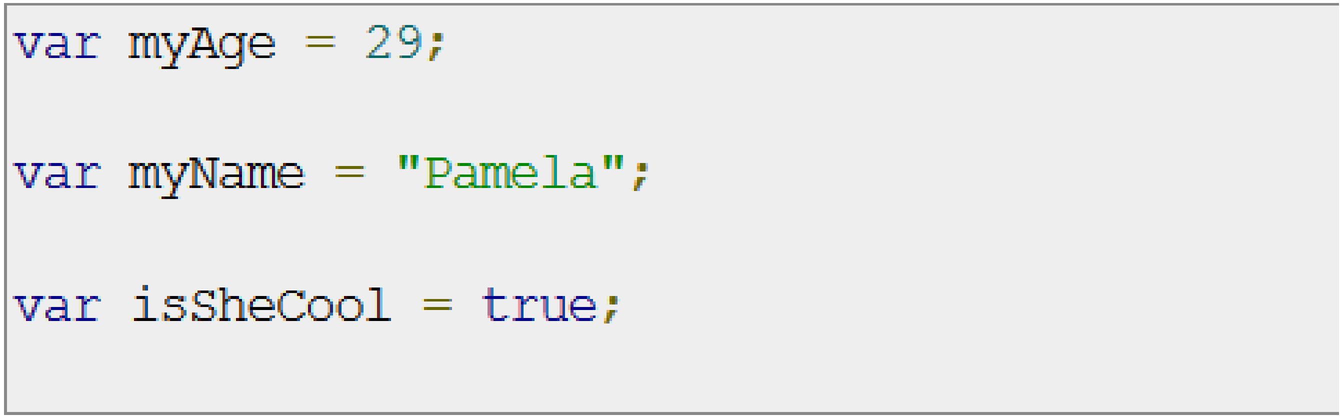
JavaScript Review

JS: Core concepts

- **Variables**: how to declare variables, assign, re-assign, and local vs. global scope.
- **Data types**: numbers, booleans, strings, arrays, and objects.
- **Functions**: how to group code into functions, pass arguments to them, and return values from them.
- **Conditionals**: how to use if/else statements and logical expressions.
- **Loops**: how to use while and for loops to repeat code.

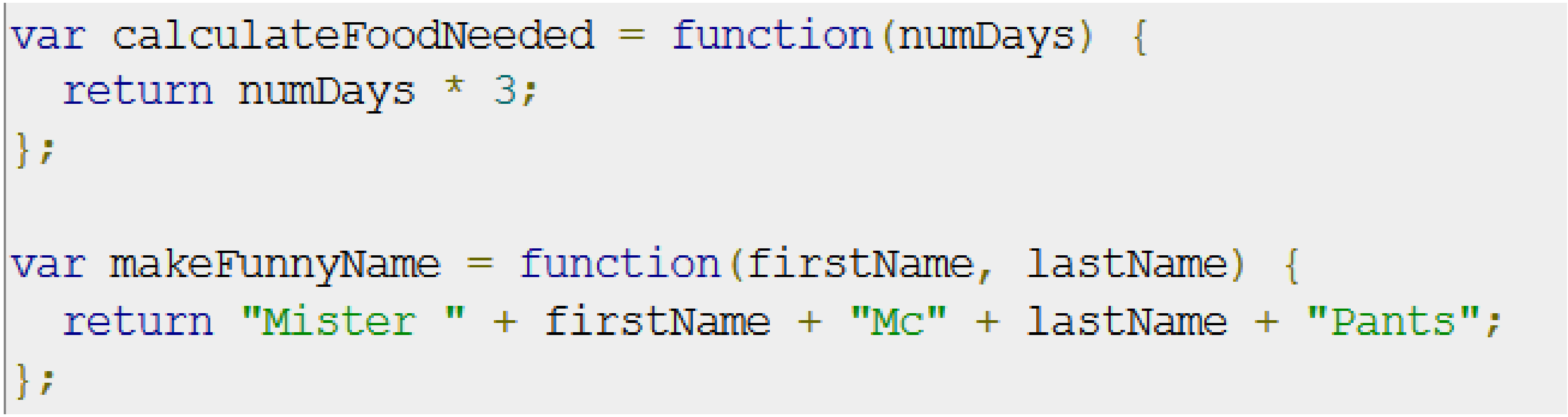
JS: Variables and Data Types



```
var myAge = 29;

var myName = "Pamela";

var isSheCool = true;
```

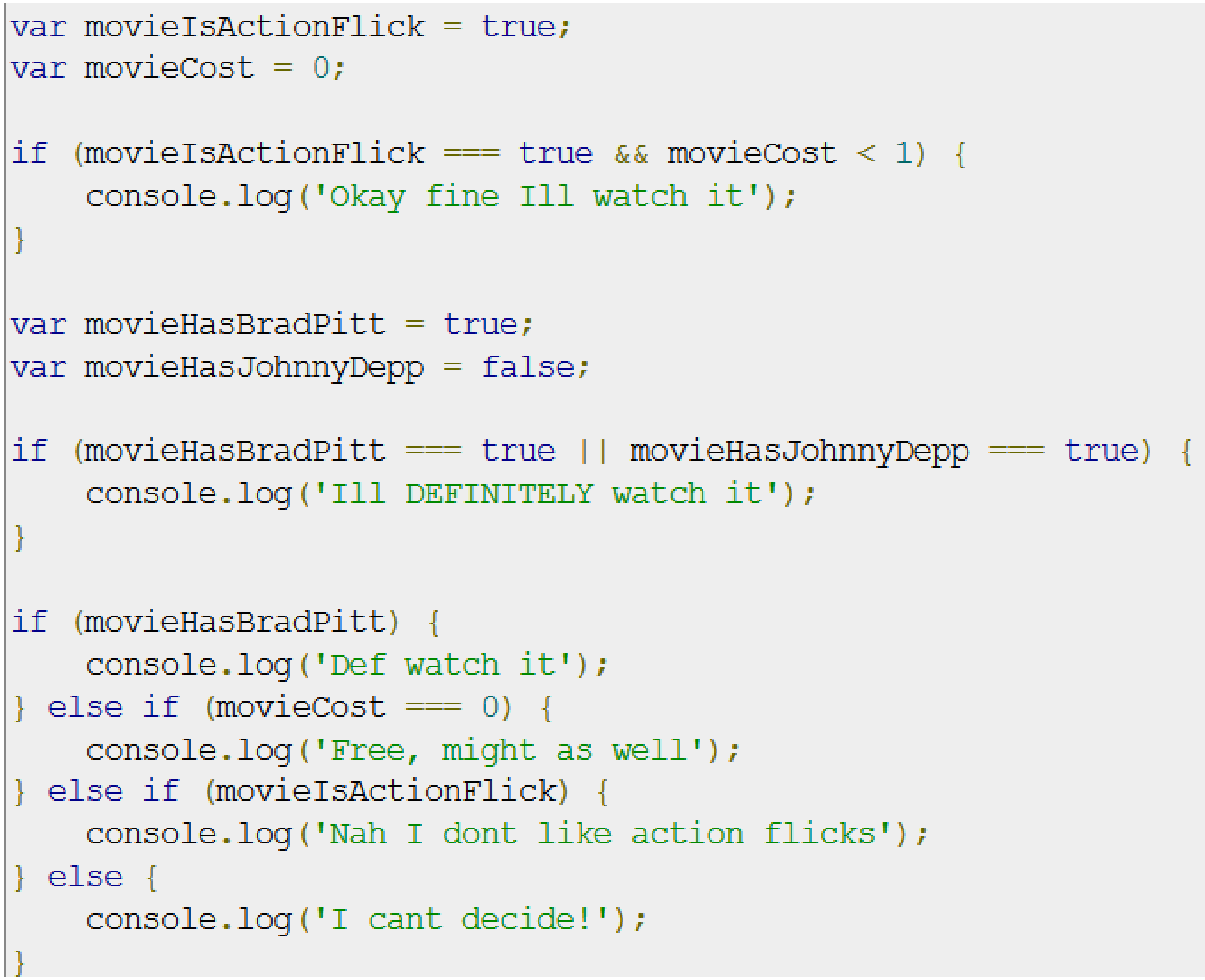
What other data types?

JS: Functions



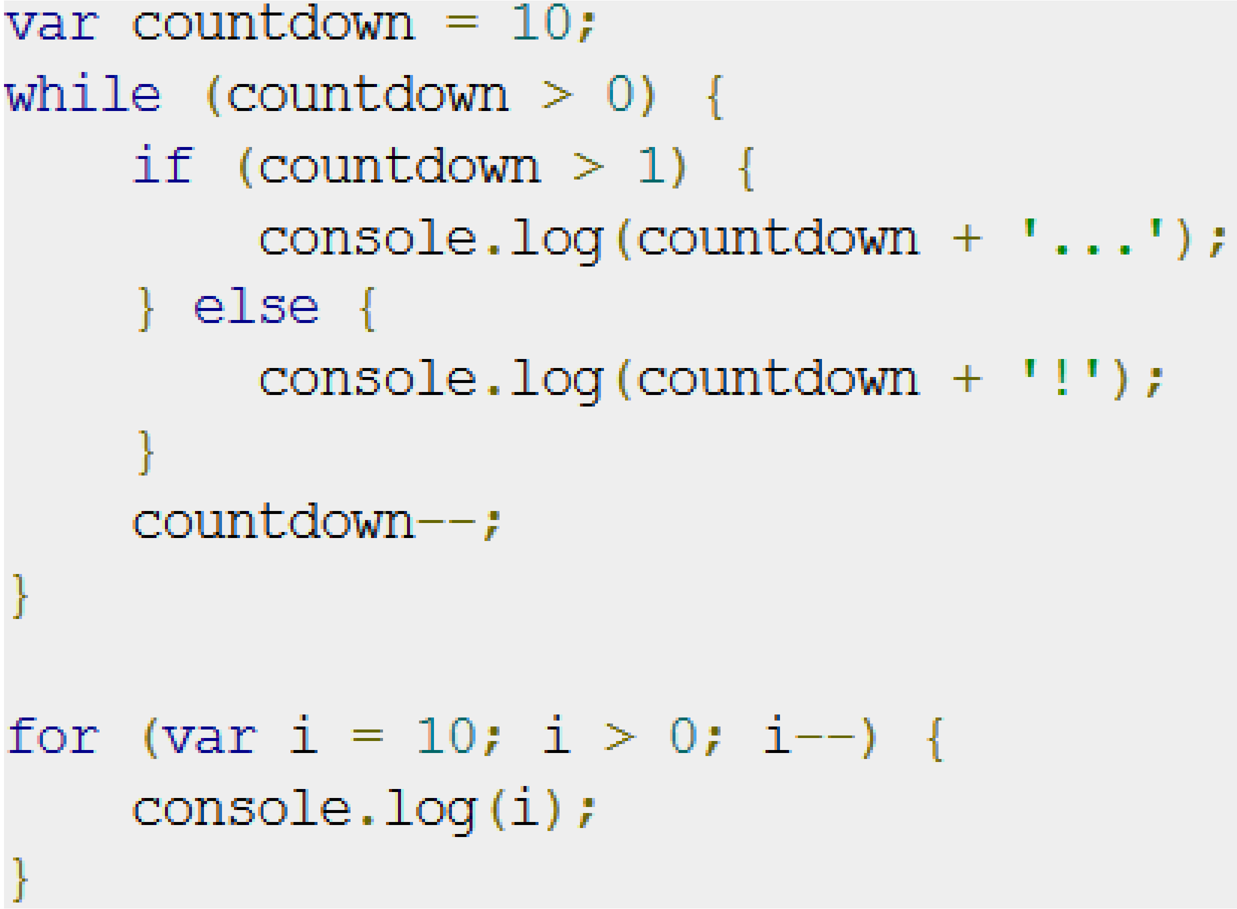
```
var calculateFoodNeeded = function(numDays) {
  return numDays * 3;
};

var makeFunnyName = function(firstName, lastName) {
  return "Mister " + firstName + "Mc" + lastName + "Pants";
};
```

JS: Conditionals



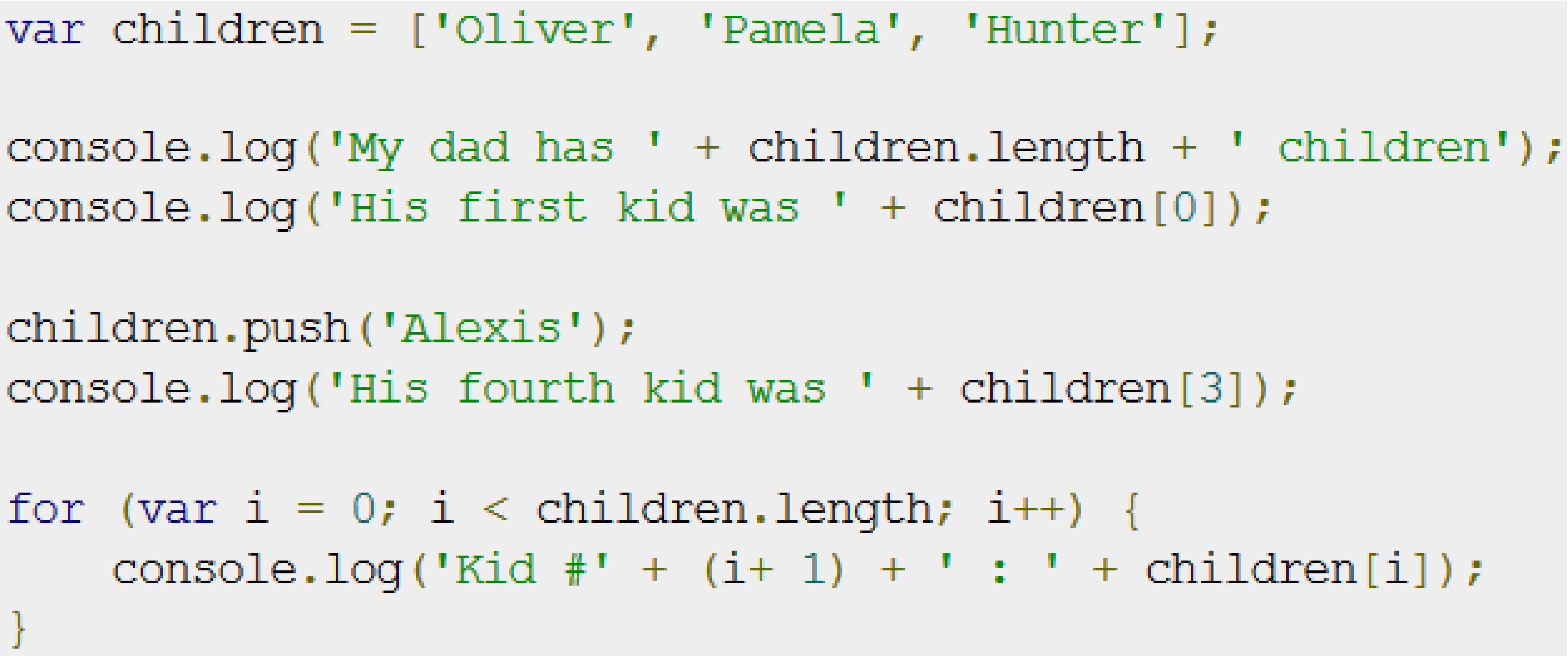
```
var movieIsActionFlick = true;
var movieCost = 0;

if (movieIsActionFlick === true && movieCost < 1) {
    console.log('Okay fine Ill watch it');
}

var movieHasBradPitt = true;
var movieHasJohnnyDepp = false;

if (movieHasBradPitt === true || movieHasJohnnyDepp === true) {
    console.log('Ill DEFINITELY watch it');
}

if (movieHasBradPitt) {
    console.log('Def watch it');
} else if (movieCost === 0) {
    console.log('Free, might as well');
} else if (movieIsActionFlick) {
    console.log('Nah I dont like action flicks');
} else {
    console.log('I cant decide!');
}
```

JS: Loops



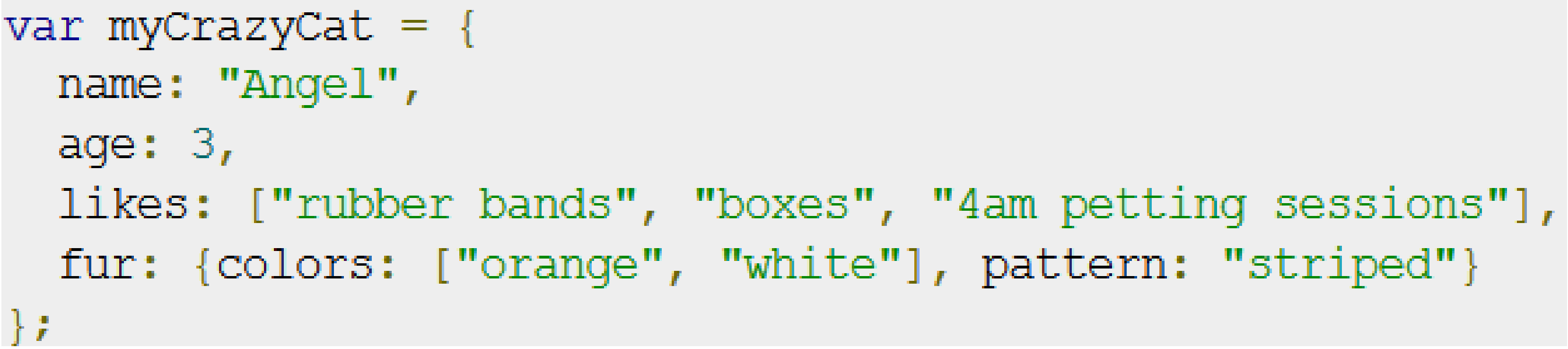
```
var countdown = 10;
while (countdown > 0) {
    if (countdown > 1) {
        console.log(countdown + '...');
    } else {
        console.log(countdown + '!');
    }
    countdown--;
}

for (var i = 10; i > 0; i--) {
    console.log(i);
}
```

JS: Arrays



```
var children = ['Oliver', 'Pamela', 'Hunter'];

console.log('My dad has ' + children.length + ' children');
console.log('His first kid was ' + children[0]);

children.push('Alexis');
console.log('His fourth kid was ' + children[3]);

for (var i = 0; i < children.length; i++) {
    console.log('Kid #' + (i+ 1) + ' : ' + children[i]);
}
```

JS: Objects



```
var myCrazyCat = {
  name: "Angel",
  age: 3,
  likes: ["rubber bands", "boxes", "4am petting sessions"],
  fur: {colors: ["orange", "white"], pattern: "striped"}
};
```

JS: Many Environments

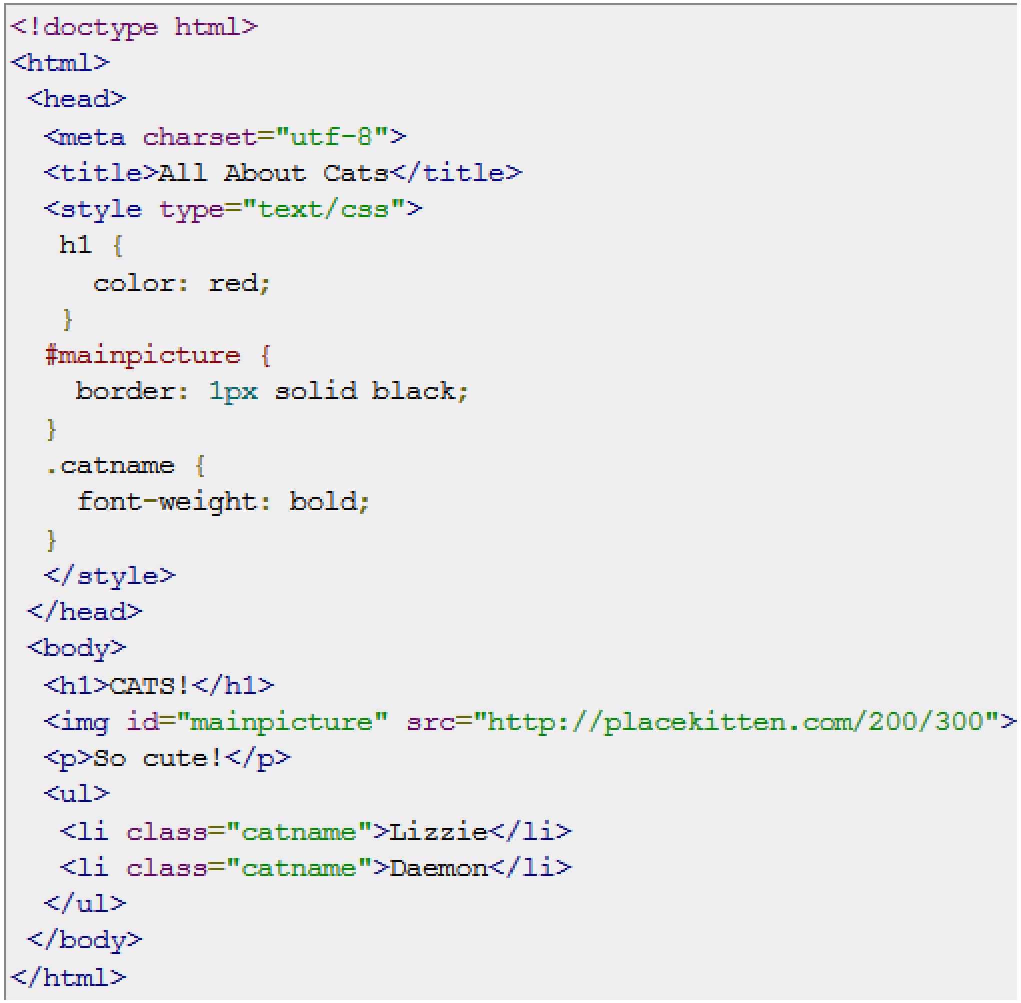
- JS can be used inside many environments for many use cases:
  - **Browser**: To make webpages interactive.
  - **ProcessingJS**: To make drawings and animations.
  - **NodeJS**: To make servers that render webpages and store data.
  - **JohnnyFive**: To control robots and arduinos.
  - **Photoshop**: To write scripts to automate image manipulation.
- Each environment comes with its own set of relevant functionality and global variables.

JS in the Browser

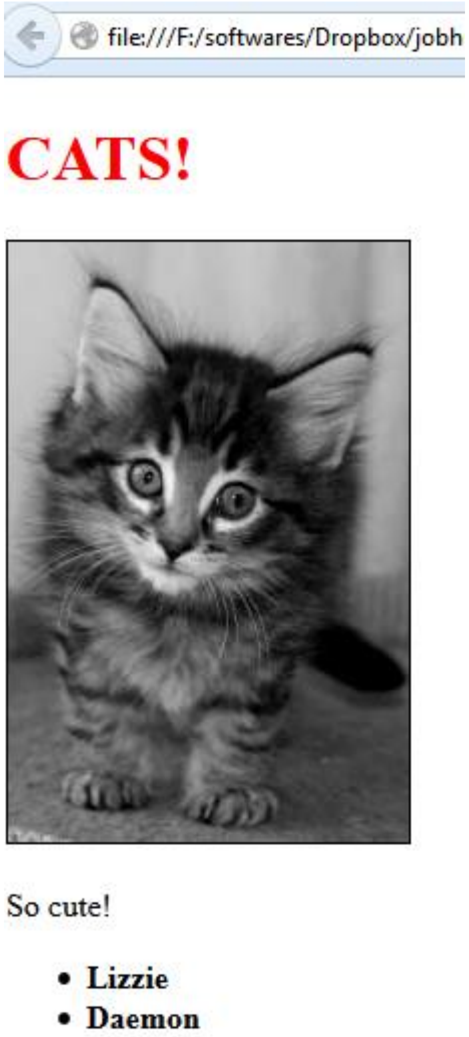
- In this environment, the functions are all for making web pages interactive, like:
  - document.getElementById("main")
  - document.body.innerHTML += "<img src='cat.gif'>";
  - window.setInterval(moveImage, 1000);
  - window.addEventListener("scroll", loadMorePics);

JavaScript DOM

HTML & CSS: Review



```
<!doctype html>
<html>
 <head>
  <meta charset="utf-8">
  <title>All About Cats</title>
  <style type="text/css">
   h1 {
     color: red;
   }
  #mainpicture {
    border: 1px solid black;
  }
  .catname {
    font-weight: bold;
  }
  </style>
 </head>
 <body>
  <h1>CATS!</h1>
  <img id="mainpicture" src="http://placekitten.com/200/300">
  <p>So cute!</p>
  <ul>
   <li class="catname">Lizzie</li>
   <li class="catname">Daemon</li>
  </ul>
 </body>
</html>
```



file:///F:/softwares/Dropbox/jobh

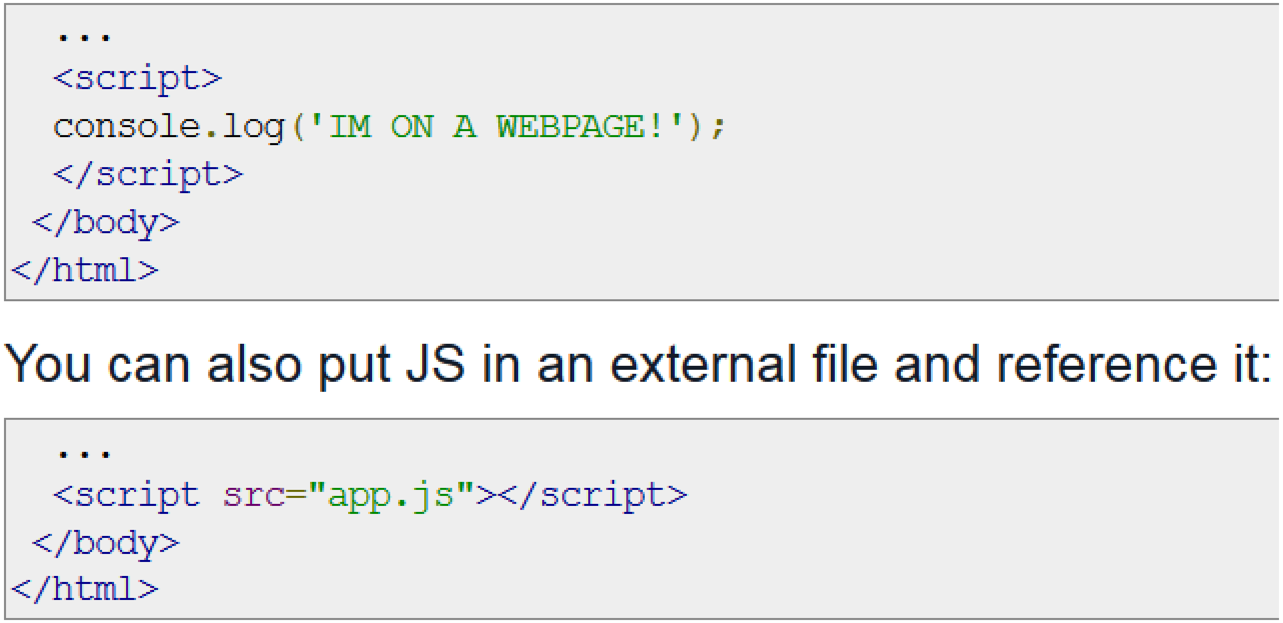
CATS!

So cute!

- **Lizzie**
- **Daemon**

JS in HTML

- You can put JS inside a script tag (commonly at bottom of the page):

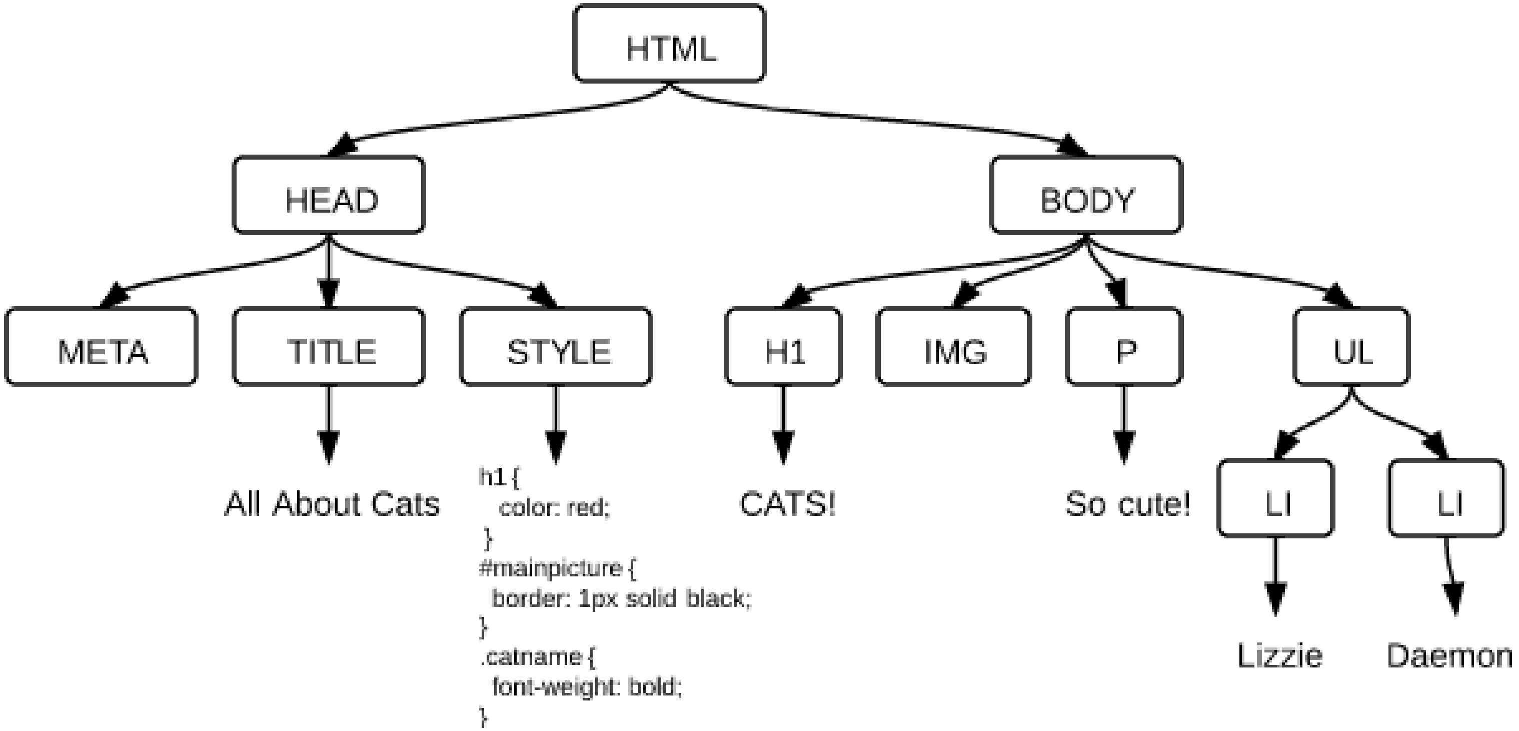


```
  ...
  <script>
  console.log('IM ON A WEBPAGE!');
  </script>
 </body>
</html>
```

You can also put JS in an external file and reference it:

```
  ...
  <script src="app.js"></script>
 </body>
</html>
```

The DOM Tree



HTML

HEAD

BODY

META

TITLE

STYLE

H1

IMG

P

UL

All About Cats

```
h1 {
  color: red;
}
#mainpicture {
  border: 1px solid black;
}
.catname {
  font-weight: bold;
}
```

CATS!

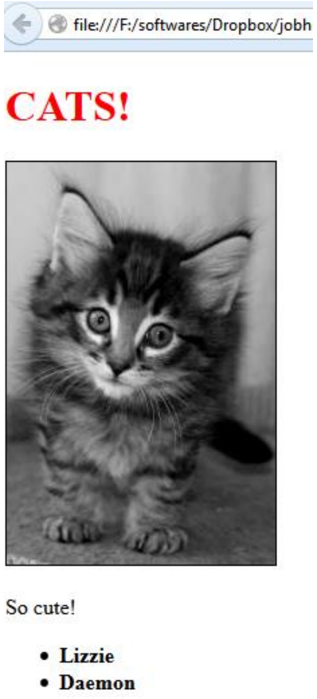
So cute!

LI

LI

Lizzie

Daemon



CATS!

So cute!

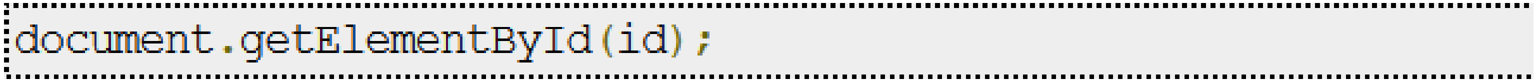
- **Lizzie**
- **Daemon**

DOM Access

- The document object gives us ways of accessing and changing the DOM of the current webpage.
- General strategy for DOM manipulation:
  - **Find the DOM node** using an access method and store it into a variable.
  - **Manipulate the DOM node** by changing its attributes, styles, inner HTML, or appending new nodes to it.

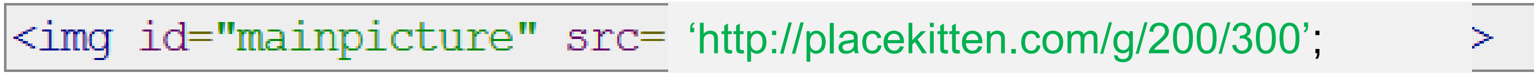
DOM Access: By Id

The method signature:



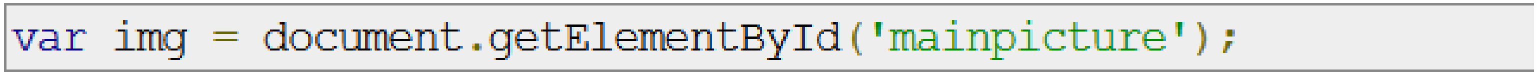
```
document.getElementById(id);
```

If the HTML had:



```
<img id="mainpicture" src= ‘http://placekitten.com/g/200/300’;          >
```

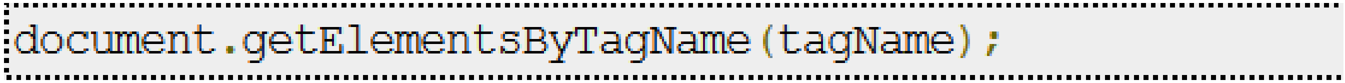
We'd access it this way:



```
var img = document.getElementById('mainpicture');
```

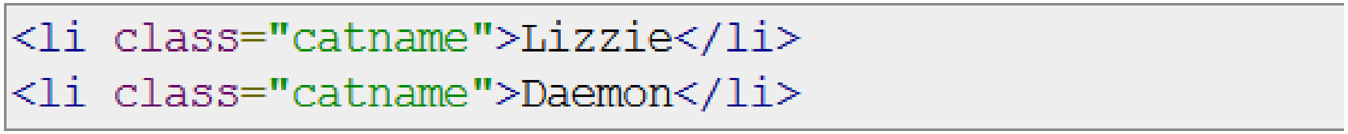
DOM Access: By Tag Name

The method signature:



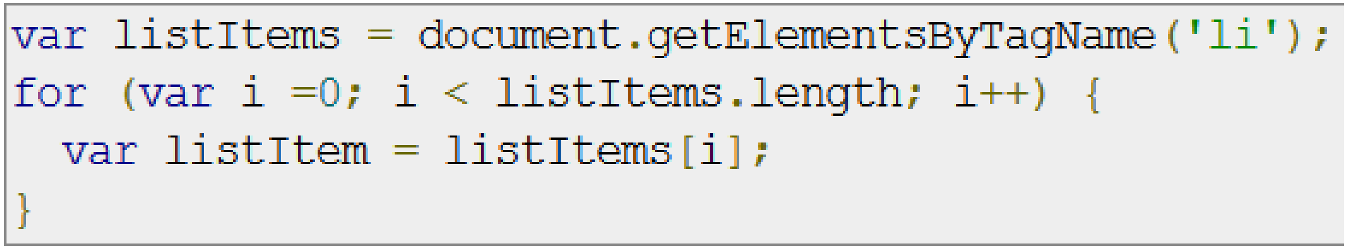
```
document.getElementsByTagName(tagName);
```

If the HTML had:



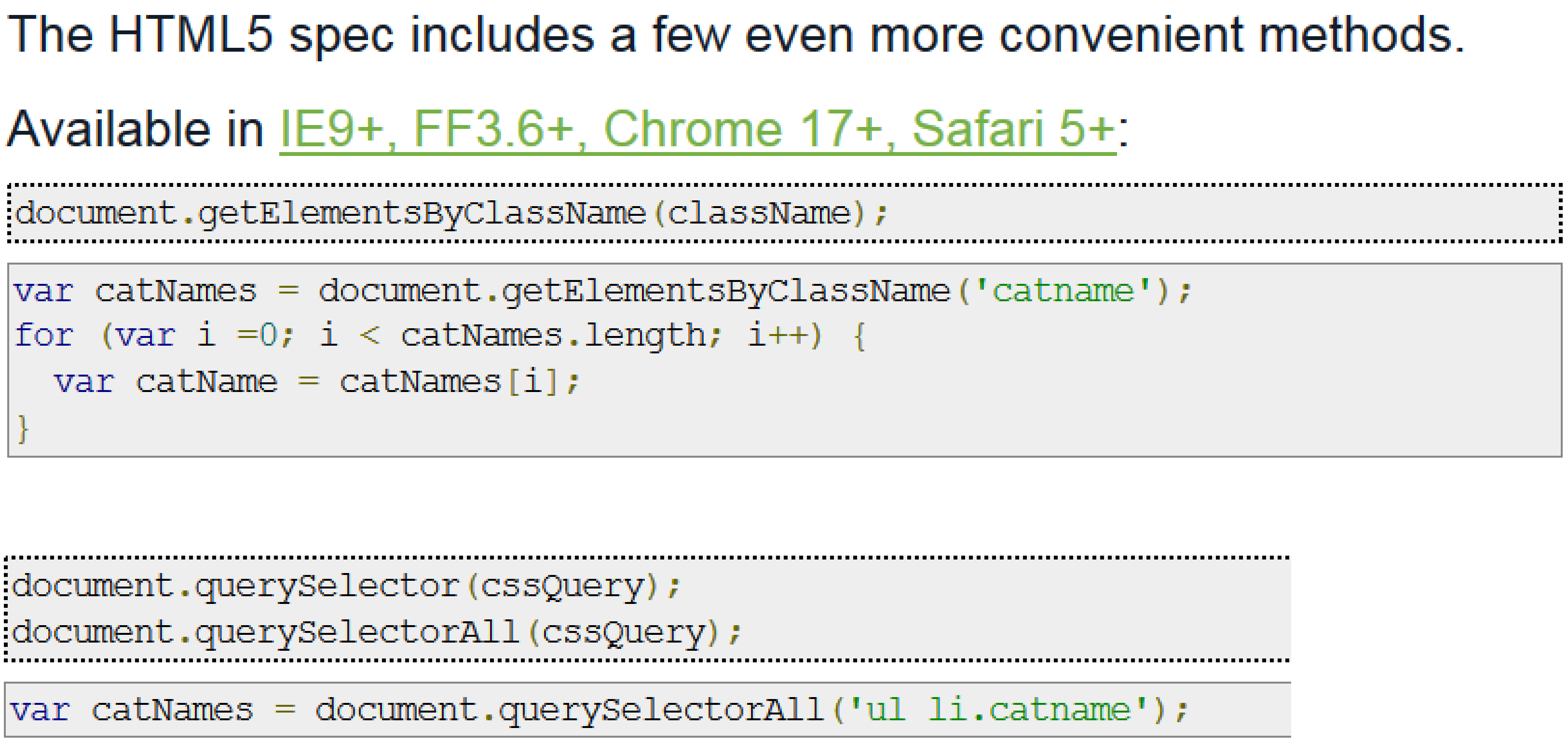
```
<li class="catname">Lizzie</li>
<li class="catname">Daemon</li>
```

We'd access it this way:



```
var listItems = document.getElementsByTagName('li');
for (var i =0; i < listItems.length; i++) {
  var listItem = listItems[i];
}
```

DOM Access: HTML5



The HTML5 spec includes a few even more convenient methods.

Available in IE9+, FF3.6+, Chrome 17+, Safari 5+:

```
document.getElementsByClassName(className);
```

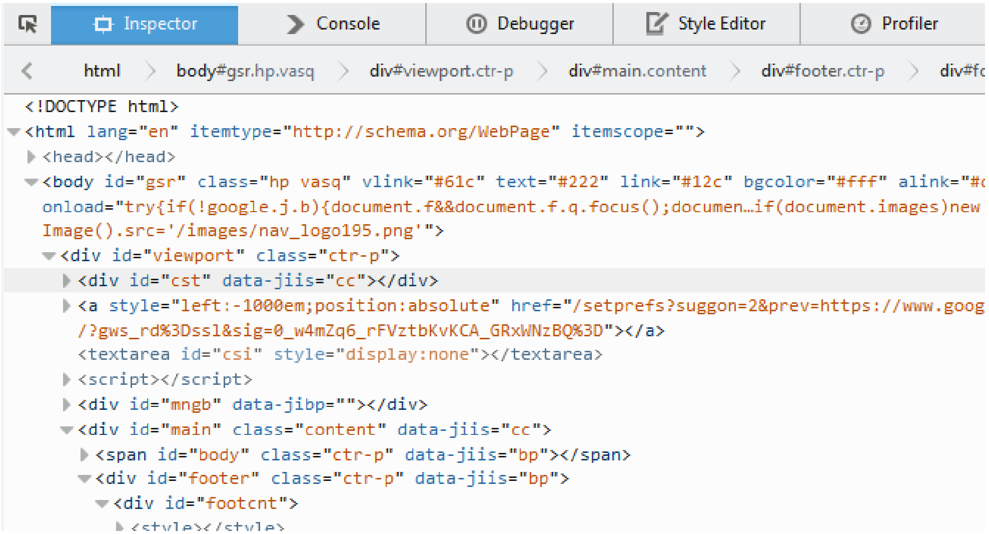
```
var catNames = document.getElementsByClassName('catname');
for (var i =0; i < catNames.length; i++) {
  var catName = catNames[i];
}
```

```
document.querySelector(cssQuery);
document.querySelectorAll(cssQuery);
```

```
var catNames = document.querySelectorAll('ul li.catname');
```

Exercise: DOM Access

- **DOM Inspecting**
  - Chrome: Right-click -> "Inspect Element"
  - Firefox: Right-click -> "Inspect Element" -> "HTML"
  - IE: Open Tools -> Developer Tools



```
<!DOCTYPE html>
<html lang="en" itemtype="http://schema.org/WebPage" itemscope="">
 <head></head>
 <body id="gsr" class="hp vasq" vlink="#61c" text="#222" link="#12c" bgcolor="#fff" alink="#c
 onload="try{if(!google.j.b){document.f&&document.f.q.focus();documen…if(document.images)new
 Image().src='/images/nav_logo195.png'">
  <div id="viewport" class="ctr-p">
   <div id="cst" data-jiis="cc"></div>
   <a style="left:-1000em;position:absolute" href="/setprefs?suggon=2&prev=https://www.goog
   /?gws_rd%3Dssl&sig=0_w4mZq6_rFVztbKvKCA_GRxWNzBQ%3D"></a>
   <textarea id="csi" style="display:none"></textarea>
   <script></script>
   <div id="mngb" data-jibp=""></div>
   <div id="main" class="content" data-jiis="cc">
    <span id="body" class="ctr-p" data-jiis="bp"></span>
    <div id="footer" class="ctr-p" data-jiis="bp">
     <div id="footcnt">
      <style></style>
```

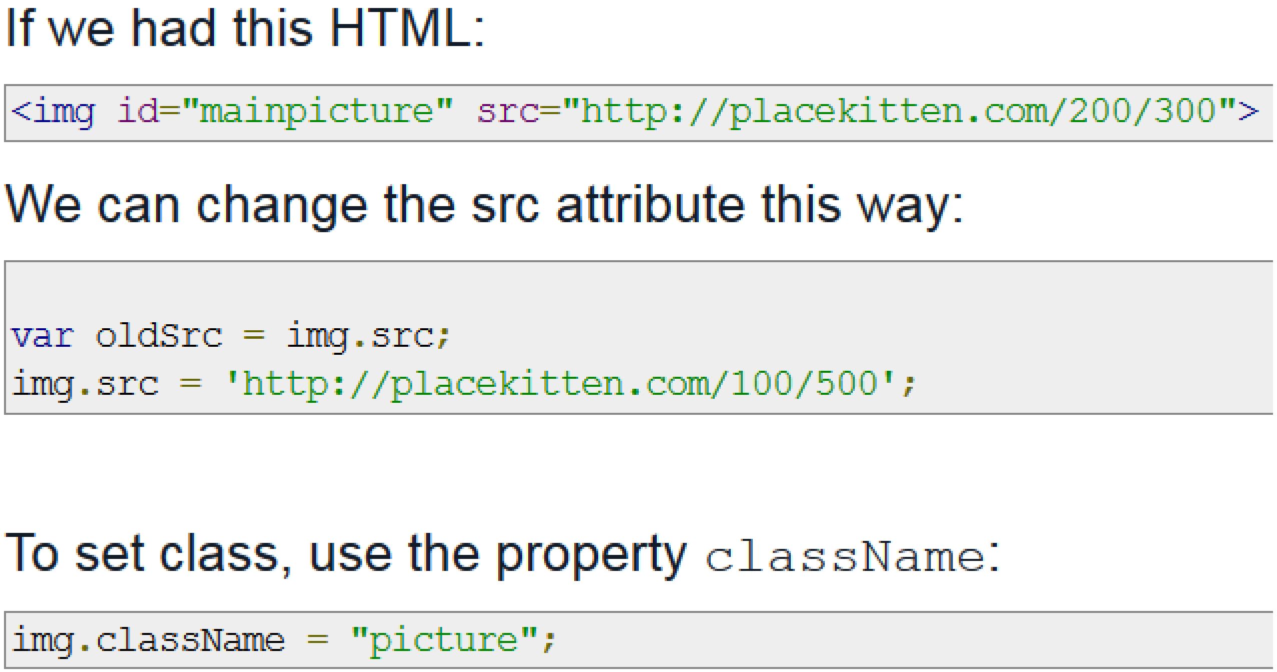
Exercise: DOM Access

Use DOM access methods to select

- Open up http://www.csc.lsu.edu/~qywang/CS6210/HTMLExercise/exercise_solution-6.html in Chrome/Firefox, and open up the console.
- Use DOM access methods to find the following parts of the page:
  - Every image on the page (getElementsByTagName)
  - The bottom left-hand twitter link (getElementById)
  - The location/city names in the table (getElementsByClassName)
  - The first location/city name in the table (getElementsByClassName)
  - All of the city names under the “The location of our collaborators” (querySelectorAll)

DOM Nodes: Attributes

You can access and change attributes of DOM nodes using dot notation.



If we had this HTML:

```
<img id="mainpicture" src="http://placekitten.com/200/300">
```

We can change the src attribute this way:

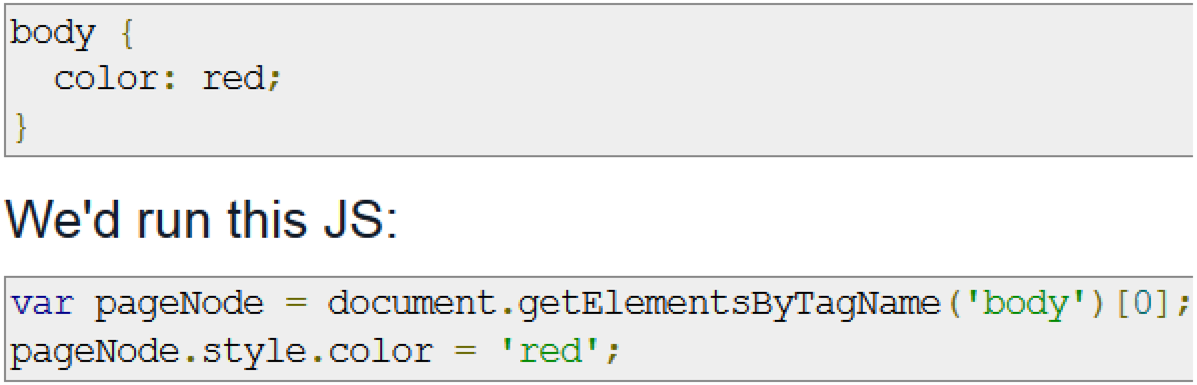
```
var oldSrc = img.src;
img.src = 'http://placekitten.com/100/500';
```

To set class, use the property `className`:

```
img.className = "picture";
```

DOM Nodes: Styles

- You can change styles on DOM nodes via the style property.
- If we had this CSS:



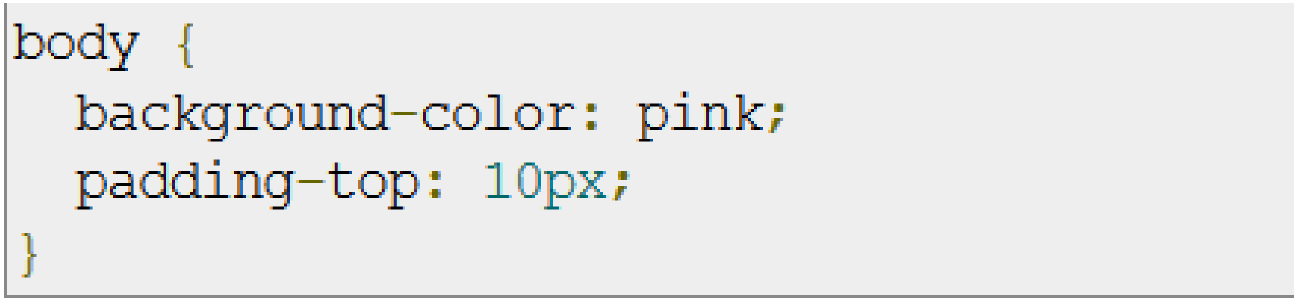
```
body {
  color: red;
}
```

We'd run this JS:

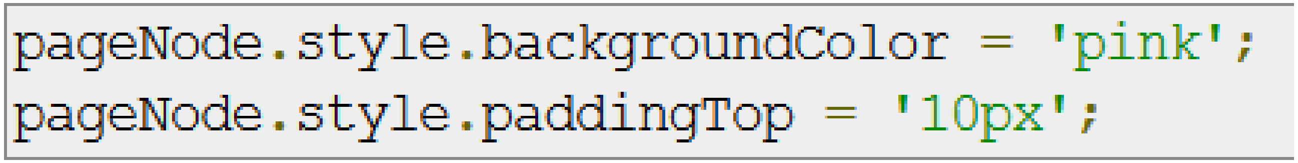
```
var pageNode = document.getElementsByTagName('body')[0];
pageNode.style.color = 'red';
```

DOM Nodes: Styles (cont’)

CSS property names with a "-" must be camelCased and number properties must have a unit:

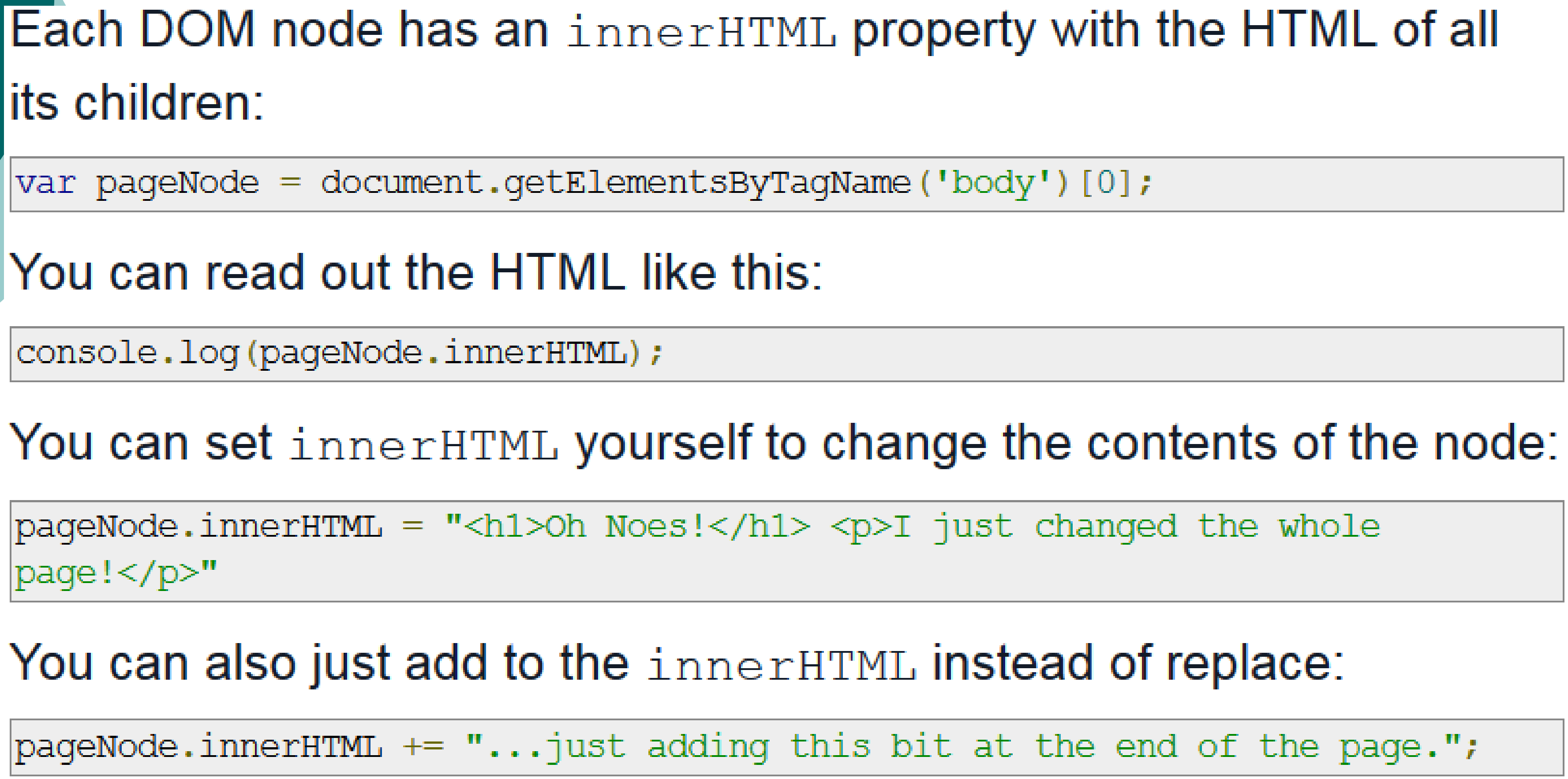


```
body {
  background-color: pink;
  padding-top: 10px;
}
```



```
pageNode.style.backgroundColor = 'pink';
pageNode.style.paddingTop = '10px';
```

DOM innerHTML



Each DOM node has an `innerHTML` property with the HTML of all its children:

```
var pageNode = document.getElementsByTagName('body')[0];
```

You can read out the HTML like this:

```
console.log(pageNode.innerHTML);
```

You can set `innerHTML` yourself to change the contents of the node:

```
pageNode.innerHTML = "<h1>Oh Noes!</h1> <p>I just changed the whole
page!</p>"
```

You can also just add to the `innerHTML` instead of replace:

```
pageNode.innerHTML += "...just adding this bit at the end of the page.";
```

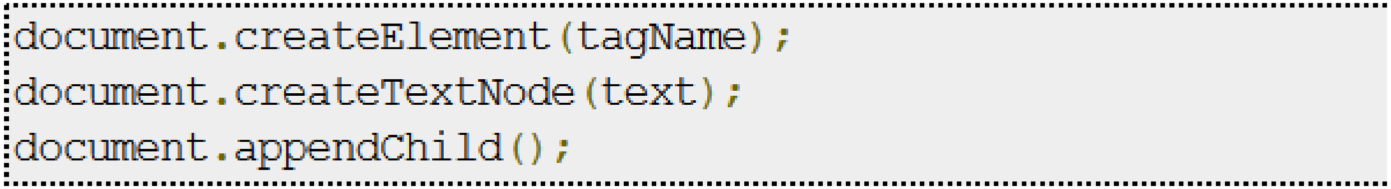
Exercises: The DOM

The Logo Hijack

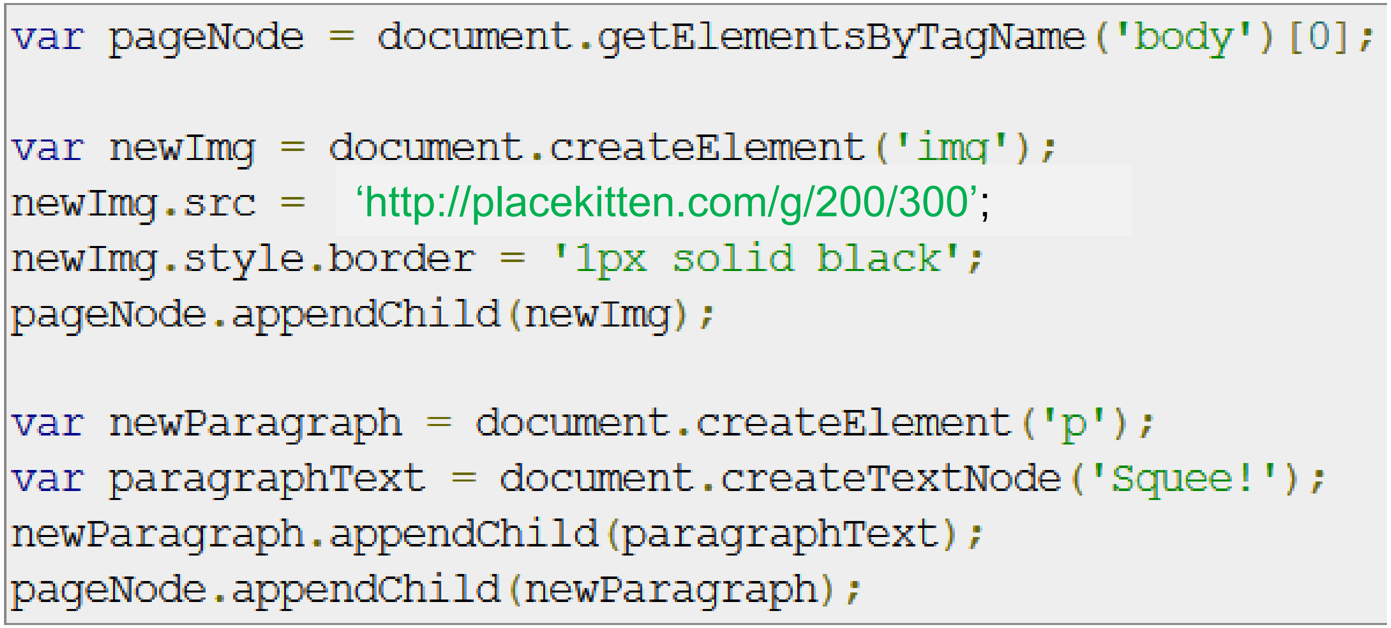
- *No homepage is safe from the logo bandit!*
  - Open up www.google.com in Chrome or Firefox, and open up the console.
  - Find the Google logo and store it in a variable.
  - Modify the source of the logo IMG so that it's a Yahoo logo instead.
  - Find the Google search button and store it in a variable.
  - Modify the text of the button so that it says "Yahooo!" instead.

DOM Modifying

The document object also provides ways to create nodes from scratch:



```
document.createElement(tagName);
document.createTextNode(text);
document.appendChild();
```

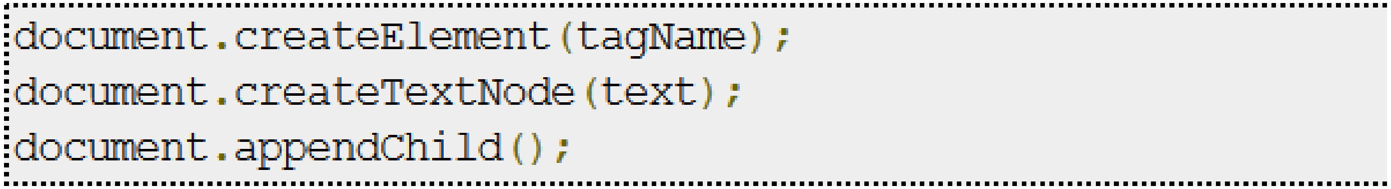


```
var pageNode = document.getElementsByTagName('body')[0];

var newImg = document.createElement('img');
newImg.src =  ‘http://placekitten.com/g/200/300’;
newImg.style.border = '1px solid black';
pageNode.appendChild(newImg);

var newParagraph = document.createElement('p');
var paragraphText = document.createTextNode('Squee!');
newParagraph.appendChild(paragraphText);
pageNode.appendChild(newParagraph);
```

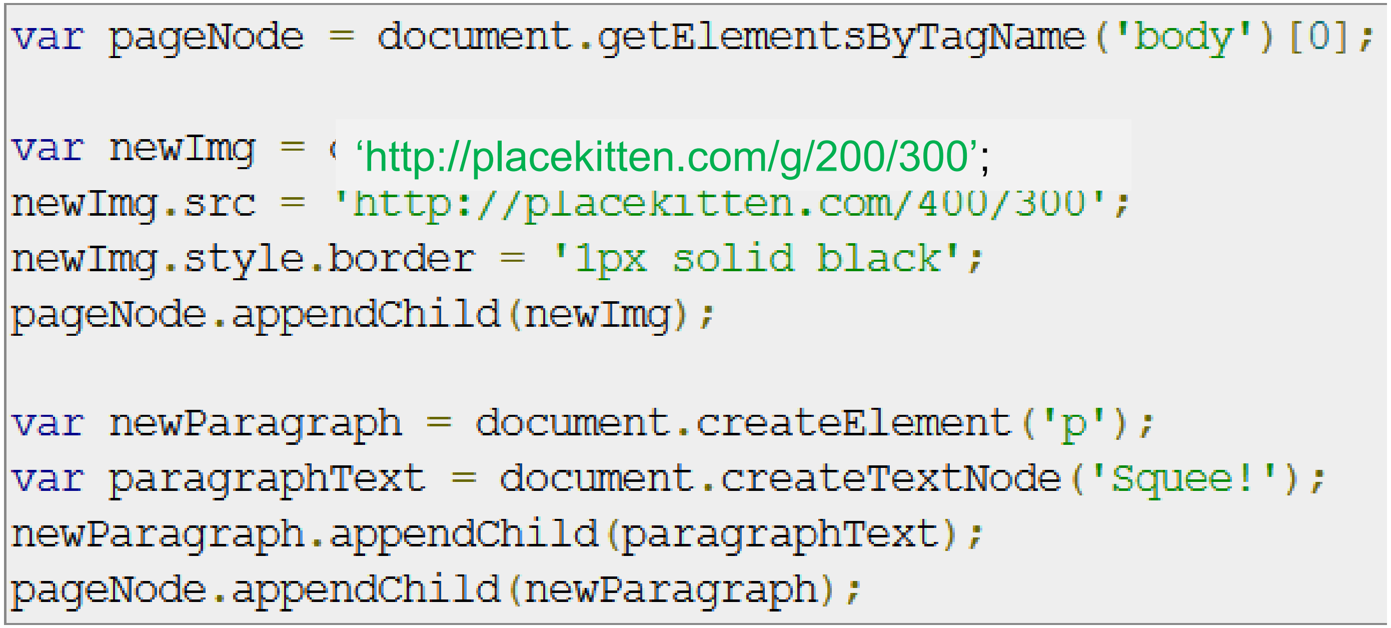
Exercise: DOM Modifying

Create nodes from scratch: open the link:

http://www.csc.lsu.edu/~qywang/CS6210/HTMLExercise/exercise-DOM-Modify.html



```
document.createElement(tagName);
document.createTextNode(text);
document.appendChild();
```



```
var pageNode = document.getElementsByTagName('body')[0];

var newImg = 
newImg.src = 'http://placekitten.com/400/300';
newImg.style.border = '1px solid black';
pageNode.appendChild(newImg);

var newParagraph = document.createElement('p');
var paragraphText = document.createTextNode('Squee!');
newParagraph.appendChild(paragraphText);
pageNode.appendChild(newParagraph);
```

‘http://placekitten.com/g/200/300’;