



SLIDES FOR THIS CLASS: <http://bit.ly/DPLjs3>

Objects

An object can hold both properties and methods (which are functions that are part of an object).

```
var objectName = {  
  property: value,  
  method: function(){  
    //what the method does goes here  
  }  
}
```

Access a property or method using dot notation:

```
var newVar = objectName.property;  
objectName.method();
```

Change, add or delete properties with dot notation:

```
objectName.property = newValue; //change a the value of a property  
objectName.newProperty = value; //add a new property to an object  
delete objectName.property; //delete a property
```

You can also create arrays of objects

```
var newArray = [  
  {property1: value1,  
   property2: value1}  
  {property1: value2,  
   property2: value2}  
];
```

And pass objects as arguments into functions

```
var myObject = {  
  property1: value;  
  property2: value;  
};  
  
function describeMyObject(argName) {  
  console.log('This object is ' + argName.property1 + " and it is " +  
    argName.property2);  
}  
describeMyObject(myObject);
```

WATCH OUT FOR TRAILING COMMAS
leaving a comma after the last item in your object will make your code not work in IE8 and below!

Document Object Model (DOM)

When the browser reads a web page, it builds up a tree structure of objects that represent all the elements on the page. The html in our DOM tree is an object named `document` in JS. One way to refer to any element in our HTML by working our way down the DOM, starting with `document`. Each element is called a node. We can reference the `parentNode` (the level above on the tree), `childNodes` (the level below on the tree), or siblings with `prevSibling` and `nextSibling` (nodes on the same level of the tree).

Finding elements by child or parent takes a lot of work. We can also find elements by ID or class:

in our HTML:

```

```

in our JS:

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

in our HTML:

```
<li class="placename">At home</li>
```

in our JS:

```
var place=
document.getElementsByClassName('placename');
```

Note the plural "Elements" - this makes an array, not a single-value variable!

We can find elements by tag name:

in our HTML:

```
    <li class="placename">At home</li>
    <li class="placename">At work</li>
    <li class="placename">In space</li>
```

in our JS:

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

We can also access the attributes of any node:

in our HTML:

```

```

in our JS:

```
var img = document.getElementById('mainpicture');
img.src = 'images/coffee2.jpg';
```

Every node has an `innerHTML` property, which overwrites the HTML of that node.

```
document.body.innerHTML = '<p>Oops, I just erased everything else on the page!<p>';
```

The document object also gives you ways to add new nodes to the page.

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