Feb 7 Lecture Notes

Middle of week 5 😊

Yay!
GetElementById

you can use:
- tag names
- class names & attributes
- get a set of elements
  → the entire class
- get form values from the user

look up & retrieve one or more elements from the DOM
Create new elements and add them to the DOM.

Remember—changes made to the DOM will show up immediately. 😊 That is what you want!
Remove elements from the DOM by taking a PARENT element & removing any of its children. 

Browser is updated immediately with this change.

remove existing elements
You can access the
- text content
  of an element
- attributes of
  an element
  i.e. perhaps change
  the class associated
  with the element.
  cod eh? 😊

Access or create an attribute of an element like `id` or `class`.

```
ul id="list"
  li
```
```javascript
var tempByHour = new Array();
tempByHour[0] = 59.2;
tempByHour[1] = 60.1;
tempByHour[2] = 63;
tempByHour[3] = 65;
tempByHour[4] = 62;
```
Type: Array

- Holds a collection of values
- Each value has an index number that starts with 0!

Here's the variable for the array:

```javascript
var tempByHour = new Array();
tempByHour[0] = 59.2;
tempByHour[1] = 60.1;
tempByHour[2] = 63;
tempByHour[3] = 65;
tempByHour[4] = 62;
```

You can add more values to the array as needed.

For every index in the array, there is a corresponding value.

To add new values to the array, we just reference the index number of the array item, and give it a value.
var tempByHour = new Array()
  tempByHour[0] = 59.2;
  tempByHour[1] = 60.1;
  tempByHour[2] = 63;
  tempByHour[3] = 65;
  tempByHour[4] = 62;

is the same as

var tempByHour = [59.2, 60.1, 63, 65, 62];
var tempByHour = [59.2, 60.1, 63, 65, 62];

tempByHour[5] = 61;

By using a new index, we get a new item in the array.

var message = "The temperature at 5 was " + tempByHour[5];
alert(message);

to access the value of the temperature at index 5, we just reference the array at index 5.

var numItems = tempByHour.length;

Ya gotta know the size of your array!
<doctype html>
<head>
  <title>Phrase-o-matic</title>
  <style>
    body {
      font-family: Verdana, Geneva, Tahoma, sans-serif
    }
  </style>
</head>

<body>
  <h1>Phrase-o-Matic says:</h1>
  <p id="phrase">hi</p>
  <button onclick="makePhrases()">Get Phrase</button>

<script>
  function makePhrases() {
    var words1 = ["24/7", "multi-Tier", "30,000 foot", "B-to-B", "Win-win", "COGS3"];
    var words2 = ["empowered", "value-added", "oriented", "focused", "amazing", "aligned"];
    var words3 = ["process", "solution", "work", "tipping-point", "strategy", "vision"];
    var rand1 = Math.floor(Math.random() * words1.length);
    var rand2 = Math.floor(Math.random() * words2.length);
    var rand3 = Math.floor(Math.random() * words3.length);
    var phrase = words1[rand1] + " " + words2[rand2] + " " + words3[rand3];
    var phraseElement = document.getElementById("phrase");
    phraseElement.innerHTML = phrase;
  }
</script>
</body>
</html>
Q: What exactly is Math, and what do Math.random and Math.floor do?

A: Math is a built-in JavaScript library that has a bunch of math-related functions in it. Math.random generates a random number between 0 and 1. We multiply that by the number of items in the array (which we get using the length property of the array) to get a number between 0 and the length of the array. The result is likely to be a floating point number, like 3.2, so we use Math.floor to make sure we get an integer number that we can use as an index into the array to pick the random word. All Math.floor does is drop the numbers after the decimal point in a floating point number. For example, Math.floor(3.2) is 3.

```javascript
var words1 = ["24/7", "multi-Tier", "30,000 foot", "B-to-B", "win-win", "COGS3"];
var words2 = ["empowered", "value-added", "oriented", "focused", "amazing", "aligned"];
var words3 = ["process", "solution", "work", "tipping-point", "strategy", "vision"];

var rand1 = Math.floor(Math.random() * words1.length);
var rand2 = Math.floor(Math.random() * words2.length);
var rand3 = Math.floor(Math.random() * words3.length);
```
Q: What happens if I try to access an array index that doesn’t exist? Like if I have 5 words stored in myWords and I tried to access myWords[10].

A: You get undefined, which, if you recall, is the value of a variable that hasn’t been assigned a value yet.
I'm not sure I get the difference between a parameter and an argument—are they just two names for the same thing?
When you define a function you can *define* it with one or more *parameters*.

Here we're defining three parameters: degrees, mode and duration.

```javascript
function cook(degrees, mode, duration) {
  // your code here
  var countMe = 1;
  var degree = degree + countMe;
  print("cooking time " duration " for" degree " ");
```
When you define a function you can *define* it with one or more *parameters*.

Here we’re defining three parameters: degrees, mode and duration.

```javascript
function cook(degrees, mode, duration) {
    // your code here
}
```

These are arguments. There are three arguments, a floating point number, a string and an integer.

```javascript
cook(425.0, "bake", 45);
cook(350.0, "broil", 10);
```