SchoolNova

IT101

JavaScript: Functions, Loops and Conditions

Functions and Variable Scope

- A program often needs to do the same thing in different places. Repeating all the necessary statements every time is tedious and error-prone. It would be better to put them in one place, and have the program take a detour through there whenever necessary. This is what functions were invented for: They are **reusable** code that a program can go through whenever it wants.
- Functions can accept arguments. Example:

```
function add(a, b) {
  return a + b;
}
alert(add(4,7));
```

- The variables in the function's local environment (inside the curly braces) are only **visible to the code inside the function**. If one function calls another function, the newly called function does not see the variables inside the first function.

- Example:

```
var v1 = "$2.56b";

function coFounder1() {
  alert("E. Williams' share: " + v1);
}

function coFounder2() {
  var v1 = "$1.05b";
  alert("J. Dorsey's share: " + v1);
  coFounder1();
}

coFounder2();
```

Eloquent Javascript, © Marijn Haverbeke 2007-2013

Loops

- A loop causes the program to repeat certain statements multiple times.
- The “for” loop syntax:

```
for ([initial-expression]; [condition]; [increment-expression]) {
  statements
}
```

- The “while” loop syntax:

```
while (condition) {
  statements
}
```

- **Exercise**: Write a “for” loop that prints out even numbers only from 2 to 1,000,000. Each number should be on a new line.

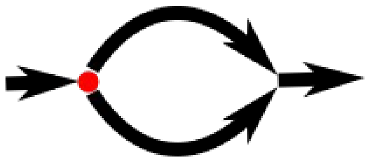
- Example:

  <p>What is the combined fortune of Evan Williams and Jack Dorsey after Twitter IPO?</p>

```
<select name="twitterCofoundersFortune">
<script>
  for (var i=1; i<10; i++) {
    document.writeln("<option value=" + i + ">" + i + " billion dollars.</option>");
  }
</script>
</select>
```

Conditions

- Executing statements in straight-line order isn't the only option we have. An alternative is conditional execution, where we choose between two different routes based on a Boolean value, like this:



- Conditional execution is written with the **if** keyword in JavaScript. In the simple case, we just want some code to be executed if, and only if, a certain condition holds.

- Example:

```
var theNumber = Number(prompt("Pick a number", ""));

if (!isNaN(theNumber)) {

  alert("Your number is the square root of " + theNumber * theNumber);

} else {

  alert("It's not a number, darling.");

}
```

Form Validation Example

```
  <script>
  function **checkAnswer**() {
     var answer = document.getElementById("**twitterCff**").value;

        if ((answer == 4) || (answer == 5)) {

           alert('correct!');

        } else {

          alert('try again');

        }

  }
  </script>

  What is the combined fortune of Evan Williams and Jack Dorsey after Twitter IPO?<br />
  <select name="twitterCofoundersFortune" id="twitterCff" onchange="checkAnswer()">
<script>

   for (var i=1; i<10; i++) {

      document.writeln("<option value=" + i + ">" + i + " billion dollars.</option>");

   }

</script>

</select>
```

Homework

- Using the Twitter example from the class, create a questionnaire with 2 questions and JavaScript form validation function(s) to validate the user's answer.
- If you end up writing 2 functions, that will work, but it is not an optimal solution, try to use only one function to validate both questions.
- Upload your code to the server and test it in several browsers (Firefox, Safari, Chrome, Internet Explorer).