JavaScript

Devpoint Labs - Dave Jungst / Jake Sorce

JavaScript

- JavaScript and Java are NOT the same thing
- JavaScript is an object-oriented programming language
- JavaScript is commonly used to create interactive effects within web browsers
- JavaScript is built in to all major web browsers

Operators

Standard JS operators are +, -, *, /, %

JS respects PEMDAS (parentheses, exponents, multiplication, division, addition, subtraction)

(5 + 3) * 2

Modulus fits in with multiplication & division

Comparators

Comparators return booleans

- 4 > 2
  - true
- 9 < 5
  - false
- 3 == 4
  - false
- 3 === 2
  - false
- 5 != 4
  - true
- 4 >= 4
  - true
- 3 <= 5
  - true

When JavaScript uses a comparator it automatically does type conversion with the exception of ===. The === operator is the same as == except that types must match

Strings

```
var str = “Hello”
var str = “Hello #{planet}”
var str = “Hello “ + planet
var str = “The meaning of life is ” + 42
var str = “The value is “ + 3/4
var str = “Name:\tDave”
var str = “I \”am not a crook\”- Nixon”
var str = “You can do it, put your \\ into it”
var str = “Roses are red\nViolets are blue”
```

```
“Hello”
“Hello #{planet}”
“Hello Earth”
“The meaning of life is 42”
“The value is 0.75“
“Name:   Dave”
“I ”am not a crook”- Nixon”
“You can do it, put your \ into it”
“Roses are red

Violets are blue”
```

Variables

var name = “Dave”

var tells js to set some space a side name is the name of the variable and in this case it has been set to the string “Dave”

console.log(name)

-> “Dave”

Convention for variable names in JavaScript is camelCase

var firstName = “Dave”

var lastName = “Jungst”

Variable naming rules

- no spaces
  - var first name
- can’t start with digits
  - var 3rdBase
- underscores are permitted but are not convention
  - this_is_a_bit_ruby_like
- JavaScript uses camel case meaning first word lowercase and all the rest capitalized
  - goodVariableName

Variable Reassignment

```
var myNumber = 3

myNumber = 4
```

Notice there is no need to add var at the beginning

```
myNumber = myNumber + 1

myNumber++

myNumber += 1
```

String properties

var longString = “This is a long string”

var shortString = “shorter”

longString.length

-> 21

longString.length > shortString.length

-> true

shortString.charAt(2)

-> “o”

longString.charAt(4) -> “ “

JavaScript doesn’t always behave like you would think.
shortString.charAt(6) … infinity

-> “”

Number methods

In ruby we have to_i and to_f in javascript we need to parse numbers from strings

Number(argument)

parseInt(argument)

parseFloat(argument)

5.toString()

5.213.toFixed(1)

5.213.toPrecision(1)

Math

JavaScript has a built in math library with very useful methods such as:

- Math.random()
- Math.min(5, 17, -4, 0)
- Math.max(1, 22, 11)
- Math.round(8.3)
- Math.celi(4.8)
- Math.floor(4.8)
- Math constants (Math.Pi, Math.E, …)

Loops

Note: In javascript blocks are surrounded by { } and we need to use ; to terminate lines.

```
for(var i = 0; i < 10; i++) {

  console.log(i);

}
```

There are 3 parts inside of our for loop parenthesis

var i = 0 initializes the variable and sets it to 0
i < 10 tells the loop to continue while i is less than 10
i++ increments i each time through the loop

Conditionals

```
if(x == true) {
  //do something
}
```

```
if(x == true) {
  //do something
} else {
  //do something else
}
```

```
if(x > 5) {
  //do something
} else if(x < 3) {
  //do something
} else {
  //do something
}
```

The parenthesis are necissary which is different from ruby. Also instead of if / end there are { } surrounding each block.

Notice the convention of where the opening and closing { } are located.

```
var theTruth = Math.pi < 4 ? true : false;

 switch ( new Date().getDay() ) {

  case 0:

    day = “Sunday”;

    break;

  case 6:

    day = “Saturday”;

   break;

  default:

     console.log(“The weekend is over”);

}
```

Functions

Methods in JS are called functions.

```
function addTwoNumbers(num1, num2) {
    return num1 + num2;
}

var myNumber = addTwoNumbers(1, 3);
```

Functions

```
function addTwoNumbers(num1, num2) {

        return num1 + num2;

}

var myNumber = addTwoNumbers(1, 3);
```

functions are declared with the keyword function followed by the function name in camelCase and then any arguments that will be passed in.

```
function name(arguments) {

  logic

}
```

Note: In JS you need to explicitly use the keyword return

Note: Functions are invoked with ()

Note: Functions are wrapped in { }

Arrays

var colors = [“white”, “blue”, “red”]

colors[0]

colors.indexOf(“white”)

colors.indexOf(“black”)

colors.join(“~”)

colors.include(“purple”)

colors.push(“green”)

colors.pop

Arrays (cont)

```
colors.length

colors.slice(1)

colors

colors.reverse()


for(i = 0; i < colors.length; i++) {

  console.log(colors[i]);

}
```

JavaScript Hash (Object)

JavaScript has hash like data structures called Objects

var obj = Object.new()

var person = { name: ‘Dave’, height: ‘6’1’ }

Accessing object properties:

person.name

person[‘name’]

JavaScript Hash (Object)

Modifying properties

person.name = ‘Jake’

person.hobby = ‘Snowboarding’

person.occupation = ‘Programmer’

Dates

JS Date objects give you the year, month, day, hours, minutes, seconds, milliseconds, and timezone offset

To get the current date: new Date()

To create a specific date either provide a date string or pass in params in the order listed above. new Date(“2015-9-15”)

d = new Date()

d.getDate()

d.getMonth()

d.getFullYear()

Listeners

JavaScript events can be triggered through the use of listeners such as a button being clicked. Common listeners:

- onchange
- onclick
- onmouseover
- onmouseout
- onkeydown
- onkeyup
- onload

<button onclick=”window.alert(‘hello world’)”>Try me</button>

Practicing JavaScript

Chrome Dev Tools

JSFiddle

JS BIN

Adding JS to HTML files

```
<html>
  <head>
    <script src='my_script_file.js'></script>
  </head>
</html>
```

Finding and modifying elements

HTML

```
<div id='some_id'>

</div>
```

JS

```
document.getElementById('some_id')
```

HTML

```
<input id='my_input'>
```

JS

```
var input = document.getElementById
('my_input');

var value = input.value
```

Finding and modifying (cont)

HTML

<label = ‘my_label’></label>

<input id=’my_input’>

JS

var input = document.getElementById(‘my_input’).value;

document.getElementById(‘my_label’).innerText = input

Event

```
<div onclick=’myFunction()’></div>
```

JS

```
function myFunction() {
  var div = event.currentTarget;
}
```

Alerts, Confirms, Console

```
alert('Hello World')
```

```
console.log("Hello World")
```

```
var choice = confirm("Do you really want to do this?")
```

JavaScript Resources

- W3 Schools
  - w3schools.com
- JavaScript
  - javascript.com
- stack overflow
  - stackoverflow.com
- Mozilla developer network
  - https://developer.mozilla.org/en-US/