IA FOR THE WEB

JavaScript Part 1
Fall 2014

Week 3+

Content

HTML

Week 6+

Style

CSS

Week 9+

Interactivity

JavaScript,
PHP

MARKUP VS. PROGRAMMING

- Markup languages define and describe things
- Programming languages define and describe things, as well as **instruct** the computer to do something

PROGRAMS

- Provide step-by-step instructions on how to accomplish something

LET'S WRITE A PROGRAM

…in English

WHAT DID WE NEED?

- A way to express data and make computations
- A way to give an instruction
- A concept of different types of data, such as text and numbers
- Containers for data
- A way to repeat a set of instructions
- A way to act upon different conditions
- A way to contain multiple pieces of data
- A way to group instructions together

PROGRAMMING LANGUAGES TEND TO HAVE

- Expressions
- Variables
- Statements
- Data types
- Control flow statements (i.e., loops)
- Conditional statements
- Functions
- Arrays
- Objects

JAVASCRIPT

- Allows you to manipulate HTML and CSS
- Understood by all major browsers
- Formally called ECMAScript
- No relation to Java

JAVASCRIPT AS AN ESSENTIAL LANGUAGE

"Web servers, rich web client libraries, HTML5, databases, even JavaScript-based languages: I see JavaScript everywhere. If you have avoided JavaScript, this is the year to learn it. There's no excuse — and if you don't, you risk being left behind." (Loukides, 2011)

PLAN

- This week we talk about JS as a language
- Next week we continue that discussion and explore using it in the browser environment

EXAMPLES

- A basic alert
- Responding to a user selection
- Providing feedback based on user activity

EXPRESSIONS

- A way to express data and make computations
- In JS, anything that produces a value is an expression. Examples:
  - 4 + 4
  - true
  - price + 5.22
  - 6 > 7
  - 9 == 9
  - age++

EXPRESSION OPERATORS

- Arithmetic
- Assignment
- Comparison
- Logical
- String

ARITHMETIC

Operator	Name	Example
+	Addition	9 + 3
-	Subtraction	`4 - height`
*	Multiplication	`height * width`
/	Division	`6 / 3`
%	Modulus (Remainder)	`rowNumber % 2`

ASSIGNMENT

Operator	Example	Result
`=`	`page = 4`	`page` is set to 4
`+=`	`page += 2`	2 is added to `page`
`-=`	`page -= 3`	3 is subtracted from `page`

INCREMENTING

Operator	Name	Example
`++`	Increment	`counter++;`
`--`	Decrement	`triesLeft--;`

COMPARISON

Operator	Name	Example
`==`	Equal	`4 == "4"`
`!=`	Not equal	`4 != 3`

`===`	Identical	`4 === 4`
		`4 === "4"`
		`4 === 4.0`

`!==`	Not identical	`3 !== 3`
`<`	Less than	`5 < 6`
`>`	Greater than	`5 > 6`
`<=`	Less than or equal to	`4 <= 4`
`>=`	Greater than or equal to	`4 >= 5`

LOGICAL

Operator	Name	Example
`&&`	And	`true && true`
`||`	Or	`true || false` `false || true`
	Exclusive or	`Doesn't exist!`
`!`	Not	`!false`

LET'S WORK THIS OUT WITHOUT THE COMPUTER

```
!((true == false) || 6 + 5 != 12)
```

STATEMENTS

- Make up a JavaScript program
- Tell the JavaScript processor what to do
- Typically end with a semicolon
- Control structures, such as `if` statements and `for` loops do not have semicolons

EXAMPLES OF STATEMENTS

- `var x = 4;`
- `x++;`
- `if(x == 4)`
- `document.getElementById('mainNav');`
- `alert('Hi!');`
- Many others

COMMON STATEMENT BEGINNINGS

- Keywords (also called reserved words)
- Variable names
- Object names
- Function names
- …among other things

COMMENTS

```
// Single-line comment

var sum = 4; // Can go here, too


/* Multiple-line

comment */
```

VARIABLES

- Containers for information, such as numbers, text, etc.
- Declared using **`var`** keyword

```
var counter;
counter = 1;

var counter = 1;
```

VARIABLES

- Start with a letter, $, or _
- Are case sensitive. Counter and counter are different variables.
- Cannot be <u>keywords</u>
- Should be meaningful and descriptive

DATA TYPES

- string
- number
- boolean
- undefined
- null

STRINGS

- Stores a group of one or more charactersEnclosed in " " or ' '
- No difference between single or double quotes in JS

```
var name = 'John Doe';

alert(name + ' is ' + name.length + '
characters long and the letter "h" is at
position ' + name.indexOf('h') + '.');
```

NUMBER

- JavaScript only has one number type
- This type can support decimals

NUMBER EXAMPLES

```
var someNumber = 9;

var someOtherNumber = 6.7;
```

BOOLEAN

- Boolean variables contain one of two values: `true` or `false`.

- `var isOpen = true;`

WHAT COUNTS AS FALSE?

- Anything that's undefined
- Anything that's null
- Anything that evaluates to `NaN`
- The numbers 0 and -0
- The empty string
- The literal `false`

WHAT COUNTS AS TRUE?

- Pretty much anything else
- The literal `true`
- Any non-empty string, including `"false"`
- Anything that evaluates to `NaN`
- Any number other than 0 and -0
- The empty string
- The literal `false`

UNDEFINED AND NULL

- `var x; //Currently undefined`
- `var x = 1;`
- `x = null; //null`

CONDITIONAL STATEMENTS

- A way to make decisions based upon different conditions
- Main structure is the `if` statement

IF

```
if(/* expression */){
   /* statements */
}

if(true && 4 < 5){
  alert(“yep!”);
}
```

IF-ELSE

```
if(/* expression */){
   /* statements */
}else{
   /* statements */
}

if(4 > 5){
  alert(“yep!”);
}else{
  alert(“nope!”);
}
```

IF-ELSE IF-ELSE

```
if(/* expression */){
   /*Do something. */
}else if(/* expression 2 */){
   /*Do something else. */
}else{
   /*Do this. */
}

if(page > 5){
  alert(“greater”);
}else if(page == 4){
  alert(“equal”);
}else{
  alert(“lesser”);
}
```

CONTROL FLOW STRUCTURES (LOOPS)

- A way to repeat a set of instructions
- `for` and `while`

WHILE

- A while loop needs one thing: an expression that must be true to continue
- It's up to the programmer to ensure that this condition will eventually be false or the program will be stuck in an infinite loop

WHILE

Expression
that must be true
to continue

```
while(i >= 0){
  alert(i);
  i--;
}
```

FOR

- A for loop needs three things:
  - A counter
  - An expression that must be true for the loop to continue
  - An expression that increments or decrements the counter

FOR

Counter

Expression
that must be true
to continue

Expression that
increments or
decrements the
counter

```
for(var i = 10; i >= 0; i--){
  alert(i);
}
```

SCRATCHPAD

- A quick way to run JavaScript
- In Firefox, go to Tools->Web Developer->Scratchpad
- Type this in Scratchpad and hit ⌘-R (Mac) or CTRL-R (Windows or Linux):

```
alert('Hello, programmer!');
```

EXERCISE

1. Write a while loop that increments a counter from 1 to 10. If the counter is 5, the loop should print out to the user that the counter is 5.

2. Write the loop you wrote under #1 as a for loop.

3. You are working on a team with other programmers. One of your fellow programmers wants to name a variable "var1". His code has examples like the following:

```
var1 = 1;
```

Briefly explain why this is not a good idea.