HACK YALE

< Web Development />

www.hackyale.com

TODAY

Plan

- Homework review
- Javascript
  - The interesting history
  - Beginning programming techniques

HACK UNIVERSITY

HOMEWORK REVIEW

HACK UNIVERSITY

PROGRAMMING

a quick intro

HACK
UNIVERSITY

WHAT IS A PROGRAMMING LANGUAGE?

Multiple definitions we might consider:

- A language designed specially to communicate instructions to a machine
- A notation for writing programs, which are specifications of a computation or algorithm
- A vocabulary and set of grammatical rules for instructing a computer to perform specific tasks

HACK UNIVERSITY

WHAT IS A PROGRAMMING LANGUAGE?

Defined by two core components: syntax and semantics

- **Syntax**: describes the possible combinations of symbols (letters, numbers, shapes, etc.) that can be used to generate code
- **Semantics**: the meaning of a programming language; i.e. how the syntax is converted into something useful

KEY TERMINOLOGY

- Code
  - The program to execute, written in a high-level language (e.g. Javascript)
- Parser
  - A program that “reads” the code and turns it into machine language (i.e. something the computer can process)
- Console
  - An environment that sends input to and receives output from the parser and execution of the program

HACK UNIVERSITY

JAVASCRIPT

A brief history

HACK UNIVERSITY

“Are JavaScript and Java the same thing?

HACK UNIVERSITY

“Are JavaScript and Java the same thing?

NO

THE NAME

In 1995, Java was hot stuff, and the Netscape guys wanted their language to be also.

- So they named it JavaScript
- Supposedly this was in exchange for bundling Sun's Java runtime with their market-leading browser: Netscape Navigator.
- Java Applets (e.g. slimezone.com) later fell out of favor, but JavaScript kept kicking.

HACK UNIVERSITY

“Isn’t Javascript that crappy little toy language that was invented in like twenty minutes by some guy at Netscape?”

HACK
UNIVERSITY

“But isn’t Javascript that crappy little toy language that was invented in like twenty minutes by some guy at Netscape?”

Sort of...

but not really anymore.

HACK UNIVERSITY

BRIEF HISTORY

- Javascript was hurriedly developed by Brendan Eich at Netscape in 1995
  - “Mocha” > “LiveScript” > “JavaScript”
  - Submitted to Ecma International for standardization in 1996 > “‘ECMAScript”
  - Iterated, updated, and enhanced since then
  - For more on Javascript’s colorful history, give this a read: http://ask.metafilter.com/195482/Lets-assume-that-I-am-the-stupidest-person-that-ever-lived-Explain-to-me-what-JavaScript-is-what-it-does-and-how-a-moron-would-go-about-learning-it#2813956

HACK UNIVERSITY

JAVASCRIPT AIN'T PERFECT...

Given the rushed conditions of its development, JavaScript has some downsides

- Too permissive. Doesn't complain
- Type coercion (i.e. == vs. ===)
- Awkward typing
  - `if ( typeof x == 'undefined' ) {...}`

HACK UNIVERSITY

BUT IT AIN'T BAD NEITHER!

JavaScript has its advantages

- Functions as first class objects
- Closures
- Event loop
- Prototype chain
- Tail recursion (in theory)
- Fast (thanks to Google's V8 engine)

HACK
UNIVERSITY

BUT IT AIN'T BAD NEITHER!

Appreciation for JavaScript has grown in recent years thanks to several key figures

- Douglas Crockford – YUI, Javascript: The Good Parts
- John Resig – jQuery
- Jeremy Ashkenas – Backbone.js
- Ryan Dahl / Joyent – Node.js

HACK UNIVERSITY

SYNTAX & SEMANTICS

HACK UNIVERSITY

VARIABLES

What is a variable? A named object to which we can assign a value (of any data type desired)

- e.g.

```
var name = “Zack”;
```

- This says: Assign the string “Zack” to the variable “name”

HACK UNIVERSITY

VARIABLE NAMING

When naming variables in Javascript, you must:

- Always begin with a lowercase letter
- Never use spaces in the name
- Use underscores or camelCase to have a multi-word name
  - Underscores: `my_multi_word_variable`
  - camelCase: `myMultiWordVariable`

HACK UNIVERSITY

ASSIGNMENT VS. EQUALITY

To **assign** a value to a variable, put the variable name on the left, the value on the right, and a single = in between

```
var name = “Josh”;
```

HACK UNIVERSITY

ASSIGNMENT VS. EQUALITY

To test for equality between values or variables, use a triple `===`

```
name === “Josh”;
```

- The parser would evaluate this expression to the boolean values `true` or `false` depending on the value stored in `name`

ASSIGNMENT VS. EQUALITY

Never use the double ==

- Like ===, == is evaluative
- It is to be avoided, because it attempts to coerce the types of its operands (more on this later)

KEY SYNTAX POINTS

- Should end every line of declarative code with a semicolon
  - e.g. `var name = “Zack”;`
- Encapsulate “code blocks” with curly braces
- Pass arguments or conditions inside parentheses
- White space doesn’t matter

HACK UNIVERSITY

WHITE SPACE BEST PRACTICES

White space doesn't matter, but...

- Start each declarative statement on a new line
- Use logical indentation to help code readability
- Include spaces inside parentheses, curly braces, or square brackets e.g. `if( condition ){ var x = 5; }`
- Use spaces to separate operators (e.g. +, *, =) from their operands (e.g. numbers, strings, variables)

HACK UNIVERSITY

DATA TYPES

Six of ‘em

HACK UNIVERSITY

DATA TYPES

Javascript has six fundamental data types

- (1) Number
- (2) String
- (3) Boolean
- (4) Function
- (5) Object
- (6) Undefined

HACK
UNIVERSITY

NUMBER TYPE

Usual arithmetic operations apply

Careful with integer arithmetic

- `3 / 2; // => 1.5`, **NOT** 1 as in other languages

Use parentheses as usual to enforce order of operations

- `(3 + 4) * 2`

HACK UNIVERSITY

NUMBER TYPE

Test for number type via:

- `typeof x === 'number';`

`parseInt` converts a non-integer or string to an integer

- `parseInt( 3.14 ); // => 3`
- `parseInt( "2" ); // => 2`

NUMBER TYPE

Useful assignment options, given a variable (e.g. `i`)

- `i++; // -> “Add 1 to i and store back to i”`
  - Equivalent to `i = i + 1;`
- `i += 1; // -> “Add 1 to i...”`
- `i--; // -> “Subtract 1 from i...”`
- `i -= 1; // -> “Subtract 1 from i...”`

HACK UNIVERSITY

STRING TYPE

Strings are text

- Use ‘’ (single quotes) or “” (double quotes) interchangeably to define
  - `“dog” === ‘dog’; // => true`
- Test for string type via:
  - `typeof x === ‘string’;`

HACK UNIVERSITY

STRING TYPE

Useful string properties

- `“dog”.length; // => 3`
- `“dog”[1]; // => “o”`
- `“dog” + “cat”; // => “dogcat”`

HACK UNIVERSITY

BOOLEAN TYPE

Booleans are truth values

- Can only have one of two values: `true` or `false`
- Test for boolean type via:
  - `typeof x === 'boolean';`

HACK UNIVERSITY

TRUTHINESS

no, we're not talking about
stephen colbert.

HACK
UNIVERSITY

TRUTHINESS AND FALSINESS

We will often rely on things
(strings, numbers, functions, etc.)
being true or false to help control
the logic of our programs

HACK
UNIVERSITY

TRUTHINESS AND FALSINESS

Easiest to think about what is “false”

- 0 (the number zero)
- ” (an empty string)
- null (a special value in JS, other languages use nil)
- undefined (a variable that has never been declared)
- NaN (not a number)

Anything that’s not false is true!

HACK UNIVERSITY

FUNCTION TYPE

Functions are a good part of Javascript

- Javascript lifted the famously good functional model of LISP
- Functions are a block of code to which we assign a name
- Declare functions as follows
  - ```
    function func_name(arg1, ... , argN) {
        /* body */
    }
    ```

HACK UNIVERSITY
    ```

OBJECT TYPE

Objects

- Object literals are defined using curly braces: {}
- Objects are collections of named variables of any data type
- Ex:
  - `{ name: “Will”, age: 27, isAwesome: true }`
- Test for object type via:
  - `typeof x === ‘object’`

HACK UNIVERSITY

UNDEFINED TYPE

Undefined

- Not too much to say here
- Until a variable is defined, its value is `undefined`
- `undefined === undefined; // => true`

HACK UNIVERSITY

REVIEW

Javascript has six fundamental data types

- (1) Number
- (2) String
- (3) Boolean
- (4) Function
- (5) Object
- (6) Undefined

HACK UNIVERSITY

CONTROL FLOW

your app's "logic"

HACK UNIVERSITY

CONTROL FLOW

Javascript control flow is similar to many other programming languages

- `if / else if / else`
- `ternary operator`
- `switch`
- `for loops`
- `while loops`

HACK UNIVERSITY

CONDITIONAL OPERATORS

- ==
  - Equality, allows for type conversion
- ===
  - Strict equality, no type conversion
- !=
  - Not equal to
- > or <
  - “Greater than” or “less than”
- >= or <=
  - “Greater than or equal to” or “less than or equal to”
- &&
  - Requires that both conditions are true
- ||
  - One or both conditions are true

HACK UNIVERSITY

A FEW EXAMPLES

```
var jaredsAge = 22;
var zacksAge = 20;
if ( jaredsAge == "22" ) { /* true because of type conversion */ }
if ( jaredsAge === "22" ) { /* false */ }
if ( jaredsAge >= zacksAge ) { /* true */ }
if ( jaredsAge === 22 && zacksAge === 20 ) { /* true */ }
if ( jaredsAge === 20 || zacksAge === 20 ) { /* true */ }
```

IF / ELSE

Controls flow based on the truth or non-truth of a condition

- Syntax: `if( condition ) { /* execute this code */ }`
- `else` specifies what to do if the `if` condition is false

```
var name = "Will";
if ( name === "Will" ) {
 console.log( "Wow, you're awesome." );
} else {
 console.log( "Seek a better name..." );
}
```

HACK UNIVERSITY

ELSE IF

If you want to specify more than 1 condition, each with different actions, you can use `else if`

```
var name = "Zack";
if ( name === "Zack" ) {
    console.log( "Wow, you're awesome." );
} else if( name === "Jared" ) {
    console.log( "Yeah, you're awesome too." );
} else {
    console.log( "Seek a better name…" );
}
```

HACK UNIVERSITY

FOR

Iterates a block of code **for** a "determinate" period

```
var name = "Jared";
for ( var i = 0; i < name.length; i++ ){
    console.log( "Letter #" + i + " is " + name[i] );
}

/* OUTPUT:
    Letter #0 is J
    Letter #1 is a
    Letter #2 is r
    Letter #3 is e
    Letter #4 is d
*/
```

HACK UNIVERSITY

WHILE

Iterates a block of code **while** a condition is true

```
var name = "Jared";
var i = 0;
while ( i < name.length ){
    console.log( "Letter #" + i + " is " + name[i] );
    i++;
}

/* OUTPUT:
    Letter #0 is J
    Letter #1 is a
    Letter #2 is r
    Letter #3 is e
    Letter #4 is d
*/
```

HACK UNIVERSITY

HOMEWORK

HACK
UNIVERSITY

ADMIN

- Homework
  - Build this:
    - http://hackyale-intro.herokuapp.com/demo
  - Extra credit:
    - Implement a drop-down menu on your website from last week.

HACK
UNIVERSITY

THANKS!

HACK
UNIVERSITY

HACK YALE

< Web Development />

www.hackyale.com