EECS 368
Programming Language Paradigms

Dr. Andy Gill

Department of Electrical Engineering & Computer Science
University of Kansas

September 15, 2015

JavaScript Stack

JavaScript

```
alert("Hello, World!");
```

Library

```
// Examples: jQuery, chalk
function $(...) { ... }
```

Browser

- Can modify DOM (Document Object Model)
- Can request specific JavaScript be run when things happen, for example a mouse click.

Your first JavaScript program in Chalk

index.html

```
<html>
  <body>
    <pre id="chalk"></pre>
  </body>
  <script src="http://code.jquery.com/jquery-1.10.2.min.js"></scrip
  <script src="chalk.20150915.js"></script>
  <script src="hello.js"></script>
</html>
```

hello.js

```
// Chalk calls main when it is completed initalization
main = function(){
  chalk.println("Hello, World!");
};
```

Chalk JavaScript Commands

chalk-commands.js

```
// Chalk calls main when it is completed initalization
main = function(){
  chalk.print("Hello, "); // print without space
  chalk.println("World!");
  chalk.newline();
  chalk.println("Hello, World!");
  chalk.hr();
  chalk.println("That’s all folks");
};
```

Chalk JavaScript Control Flow

for-loop.js

```
main = function() {
    for(var i = 0;i < 10;i++) {
        chalk.println("i = " + i);
    }
}
```

while-loop.js

```
// print all square numbers less that 10000
main = function() {
    var x = 1;
    while(x * x < 10000) {
        chalk.print(x * x + " ");
        if (x % 8 == 0) {
            chalk.newline();
        }
        x++;
    }
}
```

JavaScript Types

- `string` — "Hello!"
- `number` — 64bit numbers
- `boolean` — true or false
- `object` — key-value pairs
- `function` — can be invoked
- `undefined`

Arrays

array.js

```
main = function() {
    var arr = new Array(); // or var arr = [];
    arr[1] = "Hello";
    arr[2] = 2;
    for(var k in arr) {
        chalk.println(k + " " + arr[k]);
    }
    chalk.hr();
    arr[9] = "World";
    for(var k in arr) {
         chalk.println(k + " " + arr[k]);
    }
}
```

Array Methods

- push – push a value onto the end of an array
- pop – pop a value from the end of an array
- unshift – push a value onto the front of an array
- shift – pop a value from the front of the array
- sort – sort an array
- toString – return a String

Array Examples

sieve.js

```
var ARRAY_SIZE = 50;
main = function() {
    var sieve = new Array(ARRAY_SIZE);
    for(var i = 0;i < ARRAY_SIZE;i++) {
        sieve[i] = true;
    }
    for(var n = 2;n < Math.sqrt(ARRAY_SIZE);n++) {
        for(i = 2; i < ARRAY_SIZE;i++) {
            if (sieve[i]) {
                chalk.print(i + " ");
            } else {
                chalk.print("- ");
            }
        }
        chalk.newline();
        chalk.hr();
        for(var i = n * 2;i < ARRAY_SIZE;i+=n) {
            sieve[i] = false;
        }
    }
}
```

Objects are Named Collections

object.js

```
main = function() {
    var obj = new Object(); // or var obj = {};
    obj["a"] = "Hello";
    obj.b = "World";
    for(var k in obj) {
        chalk.println(k + " " + obj[k]);
    }
}
```