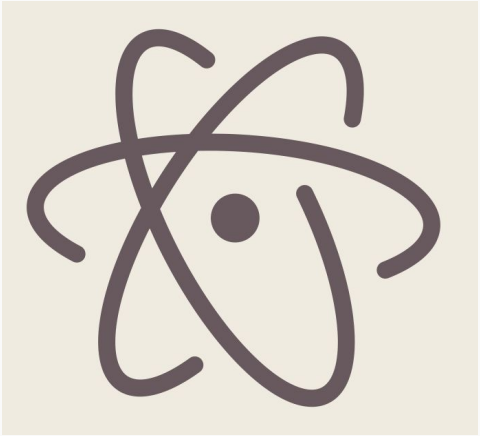
Logistics

- Slack? Node? Git repo cloned?
- Does everyone have a text editor installed?



Upcoming Events!

- Community Dinner Thursday at 6:30!
  - Get to know other HackCville members
  - Chip Ransler (our Executive Director) is speaking
  - Survivor Hour across the street afterwards...just sayin' ;)

Last Time...

- Variable Declarations + Scope
  - Let vs var vs const
  - Block Scoping
- Control Flow
  - If-else syntax
  - Ternary operator (x ? y : z)
  - Switch-case
- Declaring and using Functions

More JavaScript

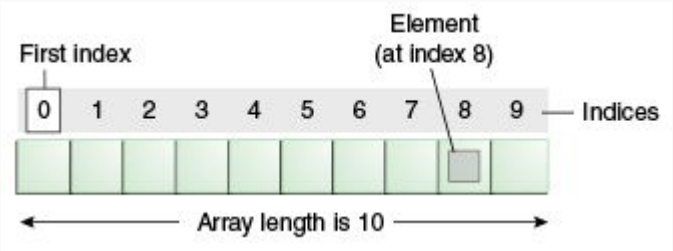
We’re Almost at the Good Stuff...

Agenda

- Arrays
- Objects + JSONs
- For and while loops
  - .map(): for loop for modifying arrays
- Arrow Functions + Functions As Parameters

Arrays

- Essentially an *ordered* collection of things
  - Strings, integers, objects, whatever
- Dynamically changes size
  - Like lists in Python, ArrayLists in Java
- Start from 0



Arrays

Create an Array

```
var fruits = ['Apple', 'Banana'];

console.log(fruits.length);
// 2
```

Access (index into) an Array item

```
var first = fruits[0];
// Apple

var last = fruits[fruits.length - 1];
// Banana
```

```
var animals = ['pigs', 'goats', 'sheep'];

console.log(animals.push('cows'));
// expected output: 4

console.log(animals);
// expected output: Array ["pigs", "goats", "sheep", "cows"]
```

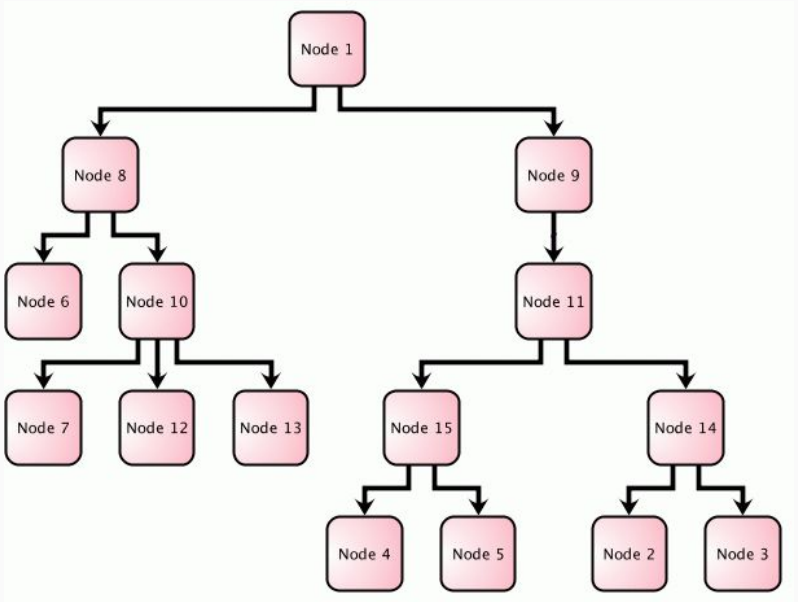
Objects

- Idea: a bunch of keywords, each of which is associated with a specific value
  - Similar to a dict in Python
- Key is a string/word, value could be ANYTHING

key	value
firstName	Bugs
lastName	Bunny
location	Earth

Objects

- Note: The value in a key-value pairing could be ANOTHER object
  - Allows us to think of Objects as trees



Object Syntax

```
var object1 = {a: 'foo', b: 42, c: {}};

console.log(object1.a);
// expected output: "foo"

var a = 'foo';
var b = 42;
var c = {};
var object2 = {a: a, b: b, c: c};

console.log(object2.b);
// expected output: 42
```

Object Example

```
let person = {
  name: "Michael Crawford",
  age: 21,
  uvaStudent: true,
  family: {
    mother: "Kate",
    father: "David"
  }
};

console.log(person.age); // 21
console.log(person.family.mother); // "Kate"
console.log(person.family); // ???
```

JavaScript Object Notation (JSONs)

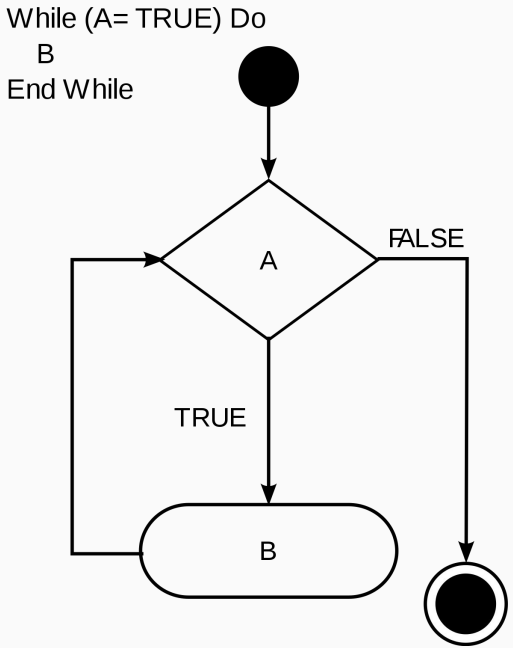
- Essentially a text file formatted like a single JavaScript Object
- Standard way to send data across the internet

Reddit Example

https://www.reddit.com/r/rarepuppers.json
http://jsonviewer.stack.hu/

While Loops

- Idea: we often want to repeat the same code multiple times in a row
- Use “while” loops when we don’t know how many times we’ll repeat



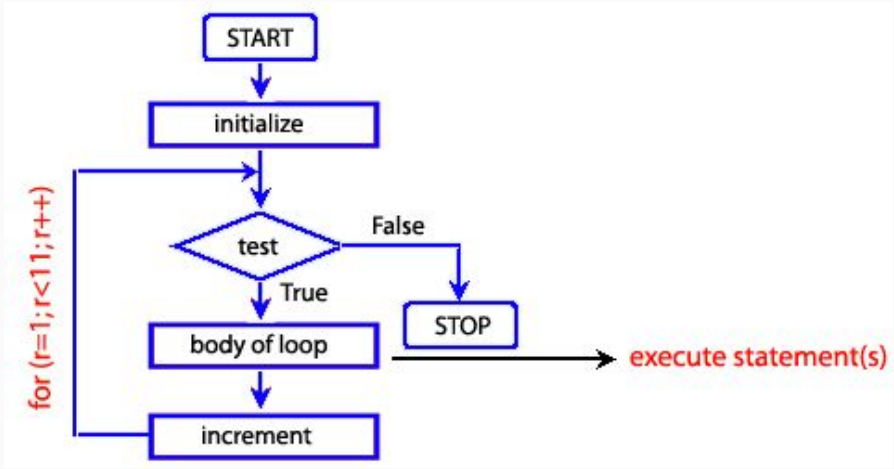
While Loop Syntax

```
while (condition) {
    code block to be executed
}
```

```
while (i < 10) {
    text += "The number is " + i;
    i++;
}
```

For Loops

- Used when we need to do something a *known* number of times
  - I.e. if there are n elements in an array, want to do something to each of the n items



For Loops Syntax

THE FOR LOOP:

```
for ([initialExpression];
     [condition];
     [incrementExpression]) { /* ACTION */ }
```

AN EXAMPLE:

```
for (var i = 0; i < 10; i++) {
    alert( i );
} // Loops 10 times
```

Functions As Parameters

- In JavaScript, function parameters don't have to just be strings, ints, etc – can also be functions!
  - Can even be a function we define "on the fly" without formally declaring
- A useful example is the .map() function

```
materials.map(function(material) {
  return material.length;
}); // [8, 6, 7, 9]
```

Arrow Syntax

- Came with ES6 conventions
- Offers a shorter way to write a function “on the fly”
  - Slightly cleaner syntax
  - Removes some problems with the “this” keyword – HUGE in React!
- (parameters) => { body of function }

```
materials.map(function(material) {
  return material.length;
}); // [8, 6, 7, 9]

materials.map((material) => {
  return material.length;
}); // [8, 6, 7, 9]
```

.map() and Arrays

- Idea: maps through an array, modifying each element based on the function you pass in, and returns a new array with the modified elements
  - Similar to a for loop for arrays
  - VERY useful in React

```
let array = [1, 2, 4, 8, 16];

let newArray = array.map((x, index) => {
  console.log(index);
  return x + 3;
});

console.log(array); // [1, 2, 4, 8, 16]
console.log(newArray); // [4, 5, 7, 11, 19]
```

hackcville.com/exit1