

Logistics

- Is everyone on Slack? In the #source channel?
- Does everyone have Node installed (instructions in Slack)?
- Does everyone have the GitHub Repo cloned?



Upcoming Events!

- Tech on Tap Tuesday 5–7 PM
 - Random Row Brewing Co. donating portion of sales to HackCville
 - Knock back a few with other members of the Charlottesville tech community
- 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' ;)

JavaScript Basics

(or just programming basics)



Who's done Codecademy?

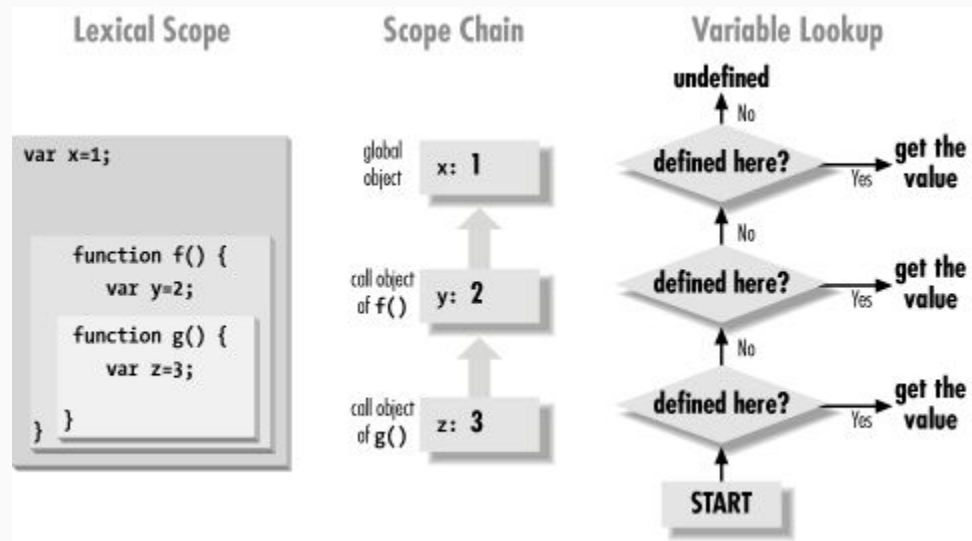
Agenda

- Variable Declarations + Scope
- Control Flow
 - If-else syntax
 - Ternary operator
 - Switch-case
- Declaring and using Functions


Variable Declaration

```
const doNotChangeMePlease = "constant";  
let iAmFreeSpiritAndWillingToChange = "free will";  
var doNotUseMe = "i suck";
```

Scope




var

var apple = 



a thing in a box
named "apple"

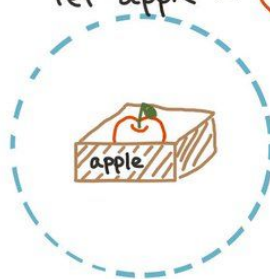
apple = 



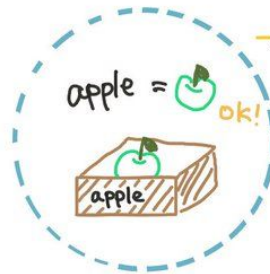
you can swap
item later

let

let apple = 



a thing in a box
named "apple" w/
protection shield



~~apple =  NG~~

you can swap item
only if you ask
inside of the shield

const

const apple = 



a thing in
LOCKED
cage
named "apple"



~~apple =  NG~~

you can't
swap item
later.



apple.multiply(3)
OK!

... but you can ask
the item to change itself
(if the item has method
to do that)

Current Conventions



Reginald Braithwaite

@raganwald



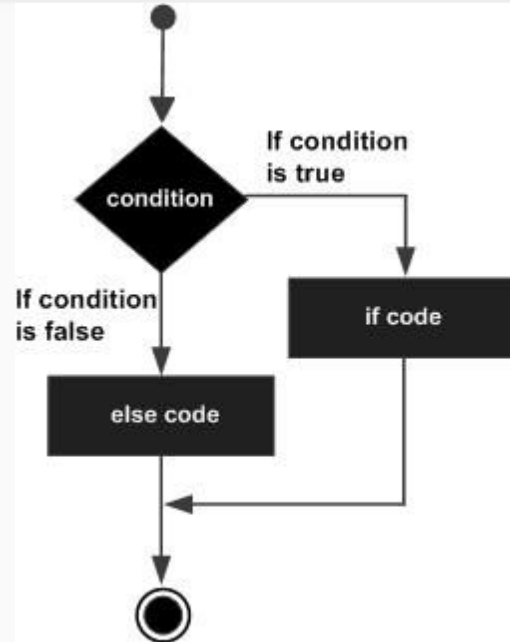
Follow

ES6 Conventions:

1. use `const` by default.
2. use `let` if you have to rebind a variable.
3. use `var` to signal untouched legacy code.

If-Else Statements

- Idea: we need our program to make decisions based on certain conditions
- Variables can be assigned “true” or “false” (called **booleans**)



“Truthy” and “Falsy”

Truthy

```
if (true)
if ({})
if ([])
if (42)
if ("foo")
if (new Date())
if (-42)
if (3.14)
if (-3.14)
if (Infinity)
if (-Infinity)
```

Falsy

```
if (false)
if (null)
if (undefined)
if (0)
if (NaN)
if (')
if (")
```

If-Else Syntax

```
if (expression) {
```

```
    // do this
```

```
} else {
```

```
    // or do this
```

```
}
```

```
function testSize(num) {  
    // Only change code below this line  
    if (num < 5) {  
        return "Tiny";  
    }  
    else if (num < 10) {  
        return "Small";  
    }  
    else if (num < 15) {  
        return "Medium"  
    }  
    else if (num < 20) {}  
  
    return "Change Me";  
    // Only change code above this line  
}
```

Checking Multiple Conditions

- Can use “&&” (read as “AND”) to check if BOTH conditions are met
- Can use “||” (read as “OR”) to check if AT LEAST ONE condition is met

```
var temperature = 105;
var rain = false;

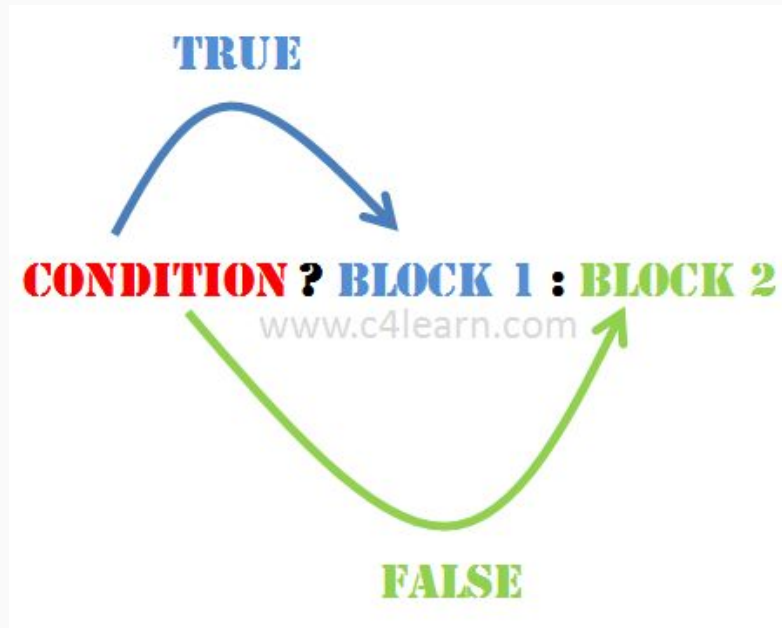
if ((temperature > 100) && (rain == false)) {
    document.writeln("It's pretty hot!");
} else {
    document.writeln("It's gonna be cold!");
}

document.writeln("XYZ");
```

Comparisons: “==” versus “===”

- “===” a bit stronger, does type-checking:
- `5 == “5”` // returns true
- `5 === “5”` // returns false (one’s a string, one’s an integer)

Ternary Operator



Ternary Operator

```
let isSourceFun = true;
```

```
let x = isSourceFun ? 80 : 72
```

```
console.log(x) // prints 80
```

```
let isSourceFun = false;
```

```
let x = isSourceFun ? 80 : 72
```

```
console.log(x) // prints 72
```


Switch-Case

- Think of as a condensed chain of if-statements

```
var expr = 'Papayas';
switch (expr) {
  case 'Oranges':
    console.log('Oranges are $0.59 a pound.');
```

break;

```
  case 'Mangoes':
  case 'Papayas':
    console.log('Mangoes and papayas are $2.79 a pound.');
```

// expected output: "Mangoes and papayas are \$2.79 a pound."

```
    break;
  default:
    console.log('Sorry, we are out of ' + expr + '.');
```

}

Functions

- Procedure or algorithm - takes an input and delivers the same output
- Three main components:
 - Function name
 - Parameters (the input(s))
 - Computations/the “return” statement

```
function square(number) {  
    return number * number;  
}
```

Functions

```
function myFunc(theObject) {  
    theObject.make = 'Toyota';  
}  
  
var mycar = {make: 'Honda', model: 'Accord', year: 1998};  
var x, y;  
  
x = mycar.make; // x gets the value "Honda"  
  
myFunc(mycar);  
y = mycar.make; // y gets the value "Toyota"  
                // (the make property was changed by the function)
```