

# FEWD

Week 5 · Class 9

#### Computer Science

#### **Quick Review**

- What does the \$ mean in jQuery?
- Which tag do we use to add Javascript?
- Where do I put the Javascript on the page?
- Can someone give me an example of an event in Javascript?
- What special method is used to wrap all of our code so that it doesn't run until the page is fully rendered?

#### What We'll Cover

- Variables, Data Types & Operators
- Logic and Conditionals

#### Variables

#### Objectives: Variables

- Understand what variables are
- Define and name variables
- Assign values to variables

#### What is a Variable?

Variables in programming are like containers used for storing pieces of data. Variables have names so that we can *access* them in order to add data to and retrieve data from them.

## **Creating Variables**

- Variables are *declared* with the var keyword.
- Variables names can contain: letters, numbers, the underscore (\_) and the dollar sign (\$), but cannot begin with a number.
- By convention, variables are named with lower camelCase.

```
var homeTeamScore;
var firstName;
```

#### Assigning a Value

The action of storing a piece of data in a variable is referred to as *assigning a value* or simply *assignment*. Assignment is done with an equals sign (=) in Javascript.

```
var lastName; /* Declaration */
lastName = 'Meade'; /* Assignment */
var age = 21; /* Declaration and assignment together */
```

## Re-assigning Variables

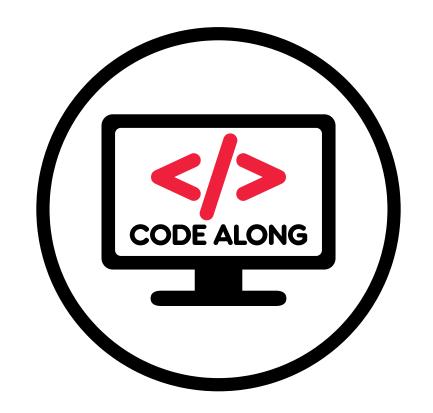
The values stored in variables declared with var can be reassigned.

```
var age = 21; /* Declaration and assignment */
age = 'not 21'; /* Reassigned */
console.log('My age is ' + age); /* outputs: My age is not 21. */
```

#### **Javascript Data Types**

# What can go in Variables?

```
height = 65.25; /* Number */
balance = -20.66; /* Number */
tired = true; /* Boolean (true or false) */
           /* undefined (declared but not assigned)*/
var book;
tickets = null; /* null (empty but not undefined) */
```

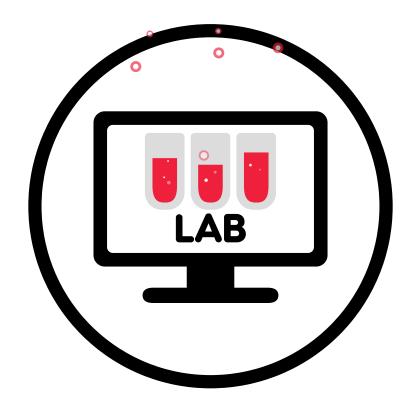


**Creating Variables** 

## jQuery .text() Method

The text method lets us replace the *text* inside of a tag.

```
var name = 'John Doe';
$('#output').text(name);
```



Score Keeper

# Operators

#### **Objectives: Operators**

- Understand how data type affects how operators behave
- Be able to recognize and use various assignment, arithmetic, string and comparison operators

## What are Operators?

Operators are special symbols that tell Javascript to perform specific operations.

- = "Yo, Javascript, assign this value to this variable."
- \* "Hey, Javascript, multiply these things!"
- > "Ummmm, Javascript, compare these and tell me if this one is larger than the other."

#### **Assignment Operators**

You've already seen the most often used assignment operator (=), but there are others.

```
Progress: %
```

```
progress /= 100;
/* progress = progress / 100 */
```

```
progress *= .01;
/* progress = progress * .01 */
```

#### **Arithmetic Operators**

With numbers, the +, -, \*, and / operators act as expected.

```
var width = 20;
var height = 30;
var area = width * height; /* 600 */
```

► HEADS UP: Area is not a function. It is assigned the number 600. If width or height is reassigned, area doesn't change!

#### Remainder Operator

The % does **not** mean percent. It's called the **remainder** operator. It gives us the remainder (as an integer) after dividing the first number by the second.

```
2 % 2;  /* 0 */
3 % 2;  /* 1 */
4 % 2;  /* 0 */
5 % 2;  /* 1 */

19 % 4;  /* 3 */
```

How could this be useful?

► HEADS UP: Beware of negative numbers and cases where the first value is smaller than the second.

## String Operator

When working with strings, the + concatenates.

```
var firstName = 'Jennifer';
var lastName = 'Meade';
var fullName = firstName + ' ' + lastName; /* "Jennifer Meade" */
```

#### Numbers + Strings

 If one value is a string and the other a number, the + operator concatenates them:

FYI: The term for this is coercion. Javascript is coercing the number value data type into a string.

#### **Arithmetic on Strings**

 If a string value could be a number, Javascript will coerce it into a number when performing other arithmetic operations:

```
var a = 2;   /* number */
var b = '5';  /* string */
var c = a * b; /* 10 */
```

FYI: If Javascript cannot coerce the string to a number, it returns a special value of Nan which stands for Not a Number.

#### **Unary Operators**

Unary operators have only one operand. The increment (++) and decrement (--) operators are unary operators you'll see a lot.

## **Converting Data Types**

You can convert a string that looks like a number to a number and numbers to strings.



Variables & Operators Takeaways

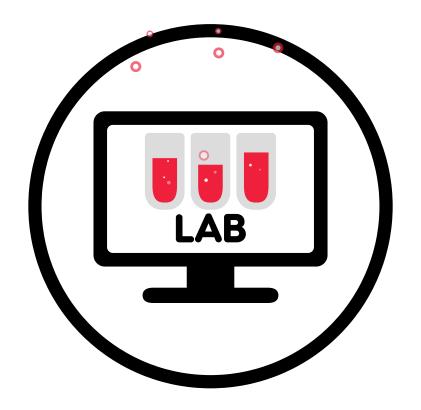
## **Takeaways**

- 1. Declare variables with var
- 2. Assign variables with
- 3. Strings must be surrounded in straight quotes
- 4. Arithmetic operators act normally with numbers
- 5. The + concatenates values that include a string
- 6. Javascript will do what it can to obey you, but coercion can lead to unexpected results

## jQuery .val() Method

The val method gets the *value* from an input field.

```
var name;
$('button').click(function(){
  name = $('#name').val();
  $('#output').text(name);
});
```



**Temperature Converter** 

#### Logic & Conditionals

# Objectives: Logic & Conditionals

- Understand how to test for equality
- Understand what logical operators are and how they work
- Assign values to variables

## **Comparison Operators**

We need to understand comparison operators to write conditional statements. Some you already know.

## **Equality**

We know that  $\blacksquare$  is the assignment operator. To check for equality, we need to use  $\blacksquare$  or  $\blacksquare$ .

► HEADS UP: In order for two things to be strictly equal (===), they must be *exactly* the same, including the data type.

# Negation (NOT)

The exclamation symbol, known as bang, means **NOT**.

## **Falsy and Truthy**

Certain values *always* return false. These are called *falsy*. If the statement does not evaluate to false and is not falsy, it is considered truthy!

## Logical OR and AND

Logical AND is written as and logical OR is written as in Javascript.

```
(6 > 5) || (6 == 7)  /* true */
(6 > 5) && (6 == 7)  /* false */
```

#### **Short-circuit Evaluation**

- false && (anything) is always false
- true || (anything) is always true

#### If Statement Syntax

```
if(condition is true) {
  /* Do cool stuff */
}
```

## If Then Syntax

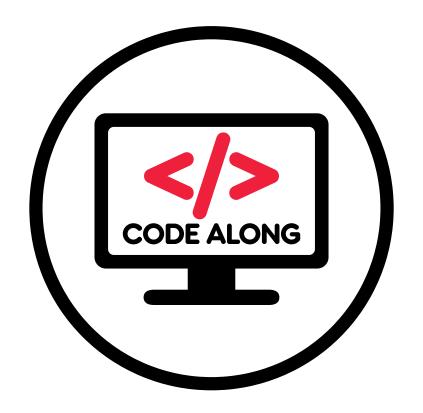
```
if(condition is true) {
   /* Do cool stuff when true */
} else {
   /* Do other stuff when false */
}
```

## If Else If Syntax

```
if(condition is true) {
    /* Do cool stuff when true
        DOESN'T CHECK ELSE IF */
} else if (condition is true) {
    /* Do cool stuff if first if condition was false
        but second if condition is true */
} else {
    /* Stuff to do if both statements are false */
}
```

## **Using Multiple Conditions**

```
if ((wifiname === 'GA-Guest') && (password === 'yellowpencil')) {
   //Give 'em access to the wifi
}
```



Working with Conditionals

#### NO CLASS 4/16

Complete any outstanding assignments:

- Profile
- Relaxr Landing and Blog pages
- Matchmaker Starter
- Proposal & Wireframes

#### Also:

- Draft HTML are due week 7, so get on it!
- Schedule office hours check in with me.

#### Go Build Awesome Things!