# Learn to Program

In Javascript

# Numbers, Variables & Strings

# What is a Program?

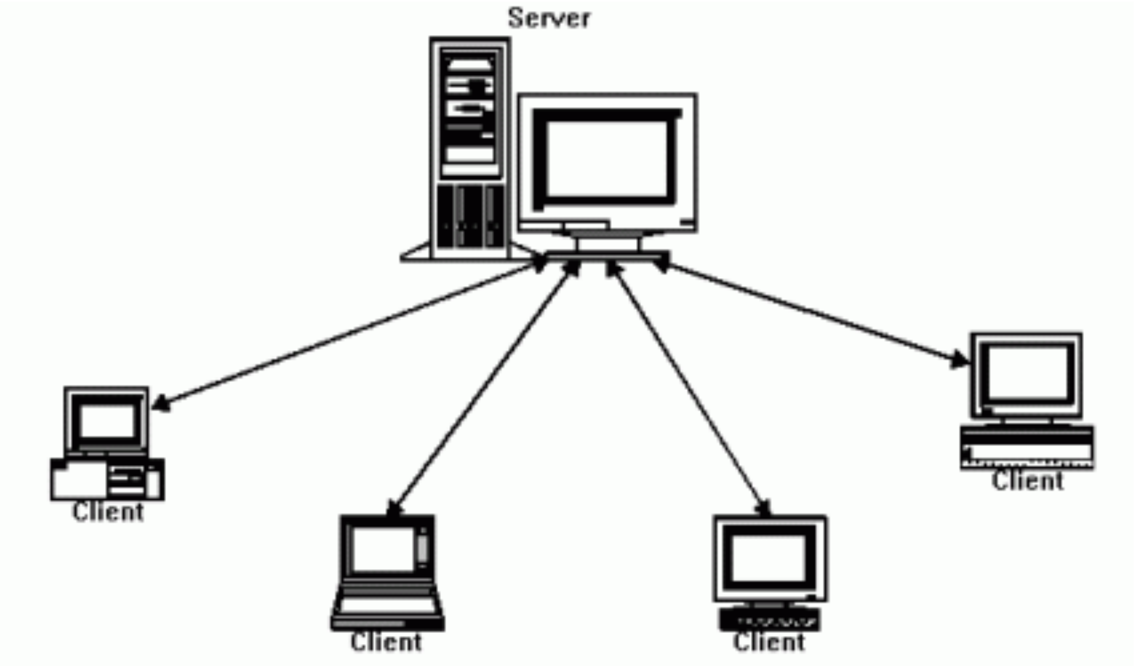
# What is a Program?

- Instructions to your computer.
- Programs are written in a code that computers understand.

# What does ‘client side’ mean?

# Javascript: Client Side

- Web based Javascript is a **client side** programming language.
- This means that the javascript code you write is interpreted by the client computer rather than the server.
- The browser is normally responsible for interpreting javascript



# Let’s write some code!

# Code for this Class:

- Clone from github:
  - git clone git://github.com/liahhansen/Learn-to-Program-Javascript.git

# Firebug

- Debugging tool
- Console
- HTTP requests

# alert() & console.log()

Javascript gives us several ways to execute code and see what is happening. Try them out in HTML page & firebug console:

```
alert(“IT’S AN EMERGENCY”);

console.log(10);
```

# Variables

- A variable is a container for a word, number or more complex data.

```
var x = "hi";
console.log(x);

var x = "oh, sorry, I thought you were someone
  else"
console.log(x);

awkwardConversationVariable = x
console.log(awkwardConversationVariable)
```

# Data Types

- Programmers create replicas of the real world inside of computers. We need a way to describe the world's people, places and things.
- The first step is to allow for different behaviors of data.
  - You want a person represented in your program to have a name
    - Use a string
  - You want to count all the people in your program.
    - Use an number
- Some Common data types: String, Number, Array, Boolean

# Strings: Quotes

Strings are a group of characters contained within quotes. The quotes can be either single or double:

```
console.log("silly string")
console.log('silly string')
```

# Single and Double Quotes

```
console.log('this won't work')
```

because the quote in "won't" is not escaped

TWO SOLUTIONS:

1. double quotes around a string with a single quote in it
2. escape the single quote with a slash

```
console.log("this won't not work")
console.log('this won\'t not work')
```

# Concatenation

You can add strings together with the plus operator:

```
console.log("silly" + " " + "string")
```

# Accessing String Characters

We can determine the character at a particular place in a string by including the index of the character we want in square brackets next to the string or charAt method:

```
alert("stringalicious"[0])
alert("stringalicious”.charAt(2))
```

We can also access more than one character using substring:

```
alert("stringalicious".substring(4))
```

# String Length

Sometimes it is very helpful to know how long your string is.  For instance, if you want to use substring to return the last three letters of a string.

```
console.log("How long am I".length);
```

# Changing Capitalization

```
console.log("I talk loudly".toUpperCase());


console.log("I TALK QUIETLY".toLowerCase());
```

# Quick Exercise:

- Log the following to the console:
  - Length of the word "ambidextrous"
  - The 4th letter in the word "monkey"
  - The 6th through 10th letters in the word "demonstration"
  - The word "shout" with capital letters
  - The word "whisper" with lower case letters

# Test First Teaching

& Test Driven Development

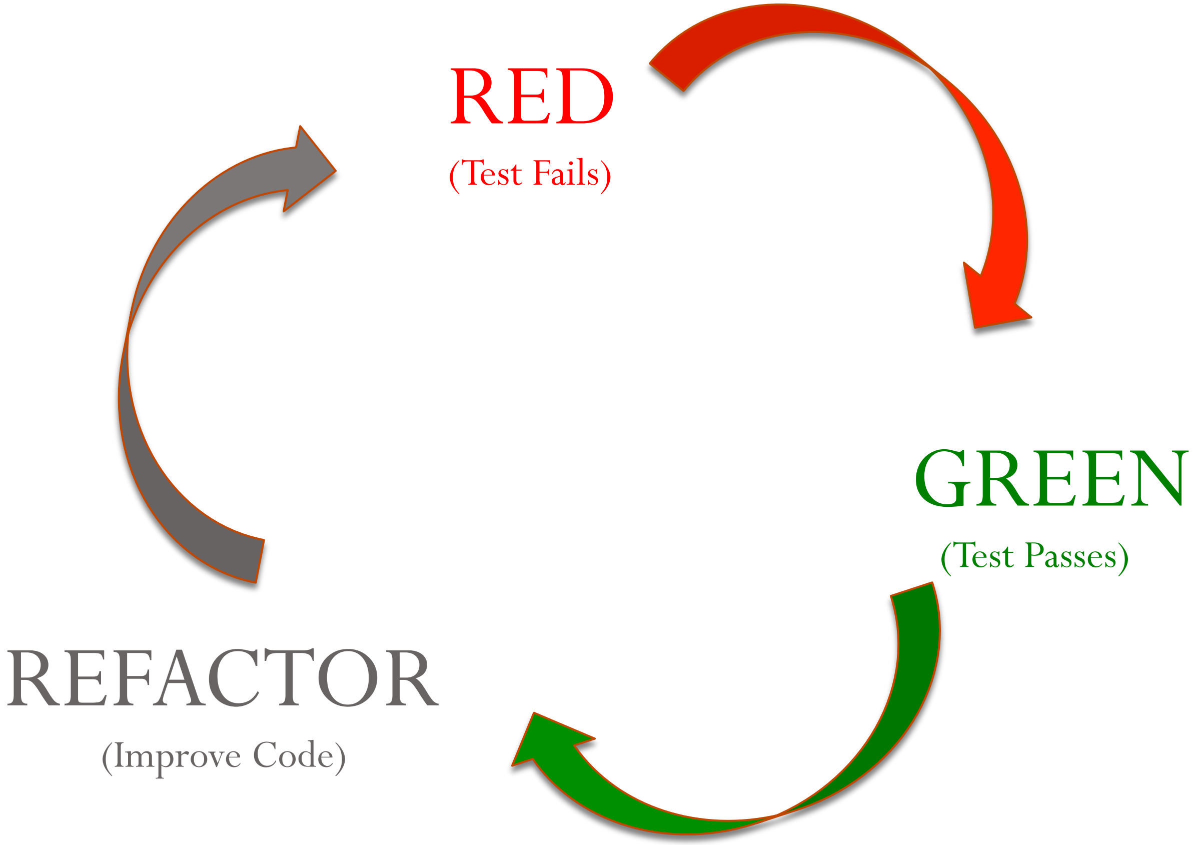
# Humans Aren’t Perfect

- Mistakes
- Unforseen Collisions
- Bugs, Bugs, Bugs!

# Testing: A Solution to Human Error

- Tests live together in a folder inside of your codebase
- Every time you add a feature, run the tests
- You can see if your new code broke any old code

# Test Driven Development: TDD



RED

(Test Fails)

GREEN

(Test Passes)

REFACTOR

(Improve Code)

# Why TDD?

- Design
  - Tests specify what behavior is desired of the code.  To write a test, the programmer needs to think through how the code will implement a behavior.  This leads to a well thought out code design.

- Project management
  - A tight testing/code-writing cycle often leads to writing less wasted code.
  - Separates design from implementation – understanding your code before you start writing it.

- Creation of tests
  - Programmers who don't use TDD often see tests as a chore to do as an afterthought, sometimes leading to tests not getting created.

# Test First Teaching: TFT

- Teacher writes the test
- Student runs the test & watches it fail
- Student writes the code to make it pass

# Jasmine

- Jasmine is a Javascript test framework
- It is in the repo of this class on github

# Exercise

- Do the exercises in /materials/strings/

# Numbers & Arithmetic Operators

```
Addition: 4 + 2
>>> 6

Subtraction: 4 – 2
>>> 2

Multiplication: 4 * 2
>>> 8

Division: 4 / 2
>>> 2
```

# Modulus Operator:

The Modulus operator finds the remainder.

```
>>> 4 % 2

0

>>> 4 % 1

0

>>> 4 % 3

1
```

# Incrementing

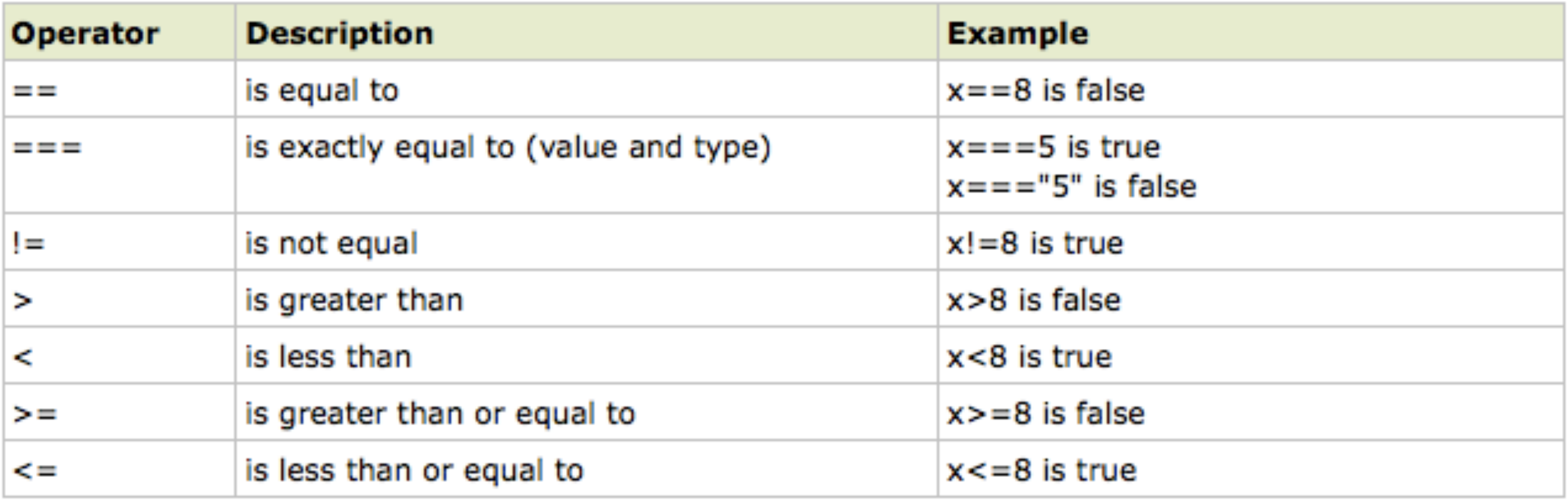
When the incrementer comes after the variable the original value of the variable is returned. When the incrementer comes before the variable, the incremented value is returned:

```
>>> i = 1
>>> console.log(i++)
1

>>> i = 1
>>> console.log(++i)
2
```

# Comparing

- If `x = 5`



| Operator | Description | Example |
|---|---|---|
| == | is equal to | x==8 is false |
| === | is exactly equal to (value and type) | x===5 is true<br>x==="5" is false |
| != | is not equal | x!=8 is true |
| > | is greater than | x>8 is false |
| < | is less than | x<8 is true |
| >= | is greater than or equal to | x>=8 is false |
| <= | is less than or equal to | x<=8 is true |

(Table is from: http://www.w3schools.com/js/js_comparisons.asp)

# Comparison Examples

```
console.log("1>1000 is ", 1>1000)
console.log("3<=16 is ", 3 <= 16)
console.log("1===1 is ", 1===1)
console.log("4==='4' is ", 4==='4')
```

# Converting a Number to String

```
var c = 12;
c.toString();
console.log(c);
```

# Number Exercise

- Write a program that calculates how old you are in seconds
- Do the exercises in /materials/numbers/

# prompt() Method

- Similar to alert(), but it can interact with the user and receive data from them.

```
var x = prompt("tell me your name")
console.log(x)
```

# Exercises

- Full name greeting:
  - Write a program that asks for a person's first name, then middle and then last. Finally, it should greet the person using their full name.
- Bigger better favorite number:
  - Write a program that asks for a person's favorite number. Have your program add 1 to it and then suggest the result as a bigger and better favorite number.
- Math
  - Write a program that accepts three numbers from the user and alerts back the the user: the sum of the numbers, the total of the numbers multiplied together, the average of the numbers, and, for extra credit, figure out how to find the largest number and smallest number.

# Homework

- Complete exercises not finished in class
- Make a program that tells the first line of a knock knock joke, records the joke reciever's answer ("who's there"), and reply with a witty punchline.  Some inspiration:
  - http://www.google.com/search?q=knock+knock+jokes&ie=utf-8&oe=utf-8&aq=t&rls=org.mozilla:en-US:official&client=firefox-a
- Reading:
  - http://www.w3schools.com/js/js_comments.asp
  - http://www.w3schools.com/js/js_variables.asp
  - http://www.w3schools.com/js/js_obj_string.asp
  - http://www.w3schools.com/js/js_comparisons.asp
  - http://www.w3schools.com/js/js_operators.asp