Introduction to Programming

Class 22, 21 March 2017

- Sit where you want today. Make sure you have a partner.

DO

Write a program in a SEPARATE javascript file that simulates a die roll and alerts the result.

dieRoll.js

```
dieRoll();

function dieRoll() {
  var MAX = 6;
  var MIN = 1;
  var num = Math.floor((MAX-MIN+1) * Math.random()) + MIN;
  alert(num);
}
```

dieRoll.html

```
<!doctype html>
<html>
<head>
    <title>Die Roll</title>
</head>

<body>
    <script src="dieRoll.js"></script>
</body>
</html>
```

Introduction to Programming
Class 22, 21 March 2017

Goals

Goal 1: Name and understand the four different kinds of functions.
Goal 2: You will know how to implement the four different kinds of functions.

Vocabulary

abstraction
modularization
function
refactor
local variables
return function
void function
arguments

Code

```
return
function name(a, b) {}
```

Partners

- Sit where you want today.

- **If you could choose your age forever, what age would you choose and why?**

BIG IDEAS!

1. Algorithms, self-contained step-by-step set of operations, are used to develop and express solutions to computational problems.
2. Effective computer programs are often the result of a collaborative effort.

3. Computers use a process called **abstraction** which reduces information and detail to facilitat focus on relevant concepts.

3. Computers use a process called **abstraction** which reduces information and detail to facilitat focus on relevant concepts.

Apps
Files
Typing
prompt()

If the user knows what something does, but doesn't know how it does it, that's abstraction.

BIG SKILLS!

1. Memory (Variables)
2. Selection (Conditionals, if-statements)
3. Iteration (Repetition, for and while loops)

4. Modularization (.js, functions)

4. Modularization (.js, functions)

Separating programs into self-contained pieces. Each piece doing a different job.

BIG IDEAS!

What is abstraction?
Give an example.

Abstraction is the process by which computers reduce information and detail to facilitate focus on relevant concepts.

Functions are the primary way to modularize code.

Void Function

dieRoll.js

```
dieRoll();

function dieRoll() {
  var MAX = 6;
  var MIN = 1;
  var num = Math.floor((MAX-MIN+1) * Math.random()) + MIN;
  alert(num);
}
```

The function dieRoll() is called a void function, meaning you can't SAVE or USE the result of the function. You just call it, and something happens.

What if you didn't always want to alert the random number?

dieRoll.js

```
var player1 = dieRoll();
if (player1 > 3) {
   alert("You win.");
}

alert(dieRoll());

function dieRoll() {
  var MAX = 6;
  var MIN = 1;
  var num = Math.floor((MAX-MIN+1) * Math.random()) + MIN;
  return num;
}
```

The function dieRoll() is now a return function.

DO

With your partner, write a program that simulates a coin flip and returns either "heads" or "tails".

coinFlip.js

```
var message = coinFlip();
alert("You flipped a " + message);

function coinFlip() {
   var num = Math.random();
   if(num < 0.5) {
      return "heads";
   } else {
      return "tails";
   }
}
```

MULTIPLE RETURN STATEMENTS; ONLY ONE OUTPUT

What if you didn't always want to roll a six-sided die?

dieRoll.js

```
var player1 = dieRoll(2, 15);
if (player1 > 16) {
   alert("You win.");
}
```

ARGUMENTS,
NO var

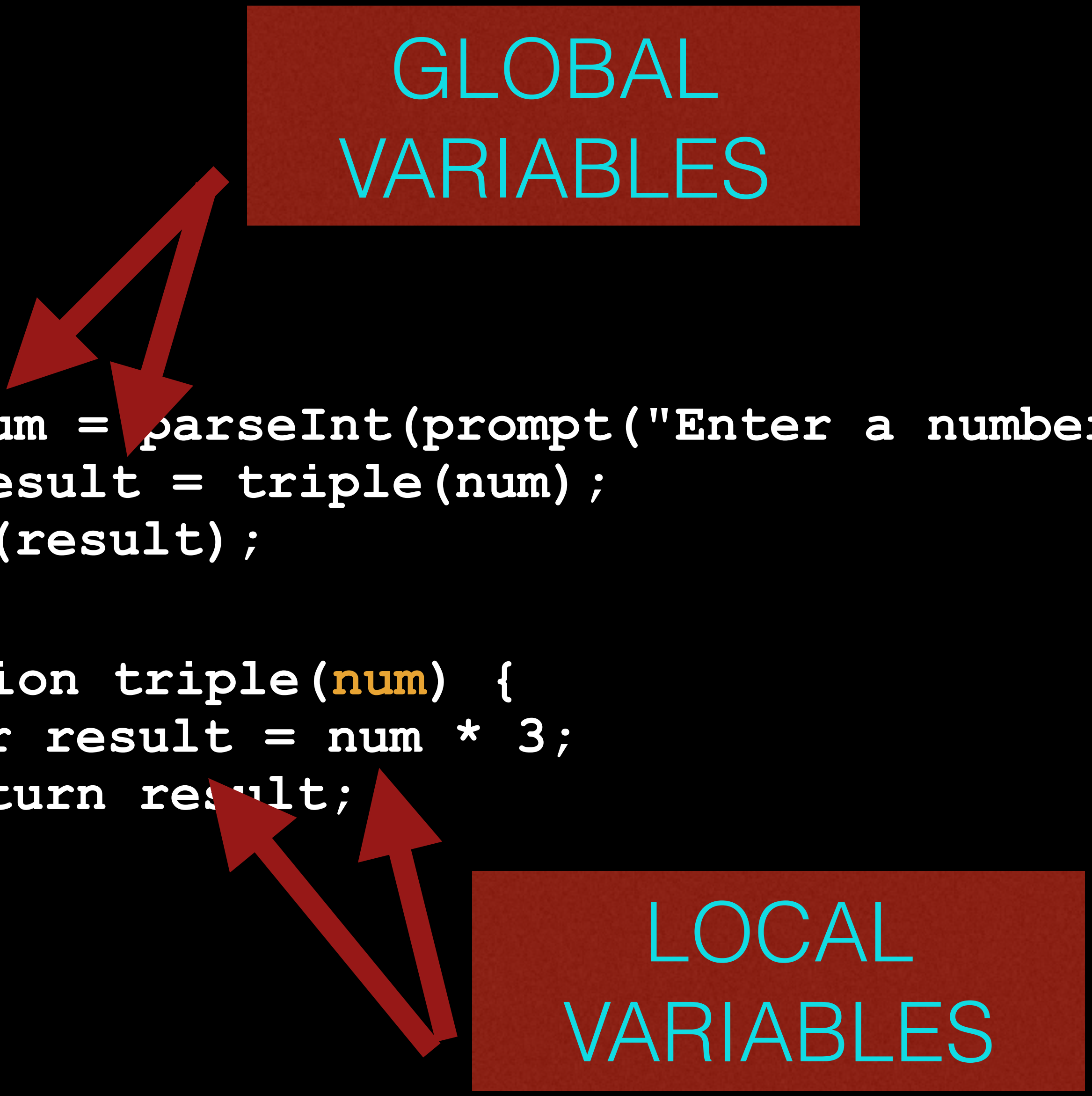
```
function dieRoll(min, max) {
  var num = Math.floor((max-min+1) * Math.random()) + min;
  return num;
}
```

The function dieRoll() is a return function that accepts arguments.

DO

With your partner, write a program that uses a function called triple. Triple should take one argument and return three times its value.

triple.js



GLOBAL VARIABLES

```
var num = parseInt(prompt("Enter a number"));
var result = triple(num);
alert(result);

function triple(num) {
   var result = num * 3;
   return result;
}
```

LOCAL VARIABLES

If local variables have the same names as global variables, the local variables will win!

Four types of functions

	Void	Return
No arguments	close()	Math.random()
Arguments	alert(*message*)	prompt(*message*)

DO

Compose a function max3() that takes three numerical arguments and returns the largest one.

Use the Math.max(a, b) function which will return the largest of either a or b.

max.js

COMMENT YOUR FUNCTIONS

```
// should display 8
alert(max3(15, 2, 8));

// max3 will return the largest of its three arguments
function max3(a, b, c) {
   var max1 = Math.max(a, b);
   var max2 = Math.max(max1, c);
   return max2;
}
```

DO

Compose a function sign() that takes a single argument *n* and returns -1 if *n* is less than 0, 0 if *n* is equal to 0, and 1 if *n* is greater than *0.*

sign.js

```
// should display 1
alert(sign(5))

// sign() will return -1, 0, or 1 depending on if n is
// negative, 0, or positive
function sign(n) {
   var result;
   if (n<0) {
      result = -1;
   } else if (n>0) {
      result = 1;
   } else {
     result = 0;
   }
   return result;
}
```

sign.js

```
// should display 1
alert(sign(5))


// sign() will return -1, 0, or 1 depending on if n is
// negative, 0, or positive
function sign(n) {
   if (n<0) {
      return -1;
   } else if (n>0) {
      return 1;
   } else {
     return 0;
   }
}
```

Challenge: can you make sign(n) shorter?

sign.js

```
// should display 1
alert(sign(5))
```

```
// sign() will return -1, 0, or 1 depending on if n is
// negative, 0, or positive
function sign(n) {
   if (n==0) {
      return 0;
   } else {
      return n / Math.abs(n);
   }
}
```

DISCUSS

What does the following code display?

```
var s = "Hello";
s = duplicate(s);
var t = "Bye";
t = duplicate(duplicate(duplicate(t)));
alert(s + t);

function duplicate(s) {
   return s + s;
}
```

DISCUSS

How many times will the loop execute?

```
var i = 0;
while (i<1000) {
   cube(i)
   i++
}

function cube(i) {
   i = i * i * i;
}
```

Four Types of Functions

	Void	Return
No arguments		
Arguments		

Homework

Complete Task 4-1 kilometerConverter.

Introduction to Programming
Class 22, 21 March 2017

Goals

Goal 1: Name and understand the four different kinds of functions.
Goal 2: You will know how to implement the four different kinds of functions.

Vocabulary

abstraction
modularization
function
refactor
local variables
return function
void function
arguments

Code

```
return
function name(a, b) {}
```