Intro to Programming

Class 23 - 15 Nov 2016
Ms. Santos

Partners

- Daya and Tyler
- Alex and Rohan
- Anna and Kathryn
- Jonah and Ayse
- Diego and Amaan
- Quinn and Tucker
- Rohan and Dev

Goals:

Goal 1: Name and understand the 4 different kinds of functions.

Goal 2: You will know how to implement the 4 different kinds of functions.

Vocabulary

return function

void function

arguments

Code

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

BIG IDEAS

Review: What is abstraction?

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

Functions are the primary way to modularize code.

DO

With your partner, write a program in a SEPARATE javascript file that simulates a die roll and alerts the result. (Remember to have a plane HTML file that calls the dieRoll.js and create a folder called dieRoll)

** HINT: We have done this before! **

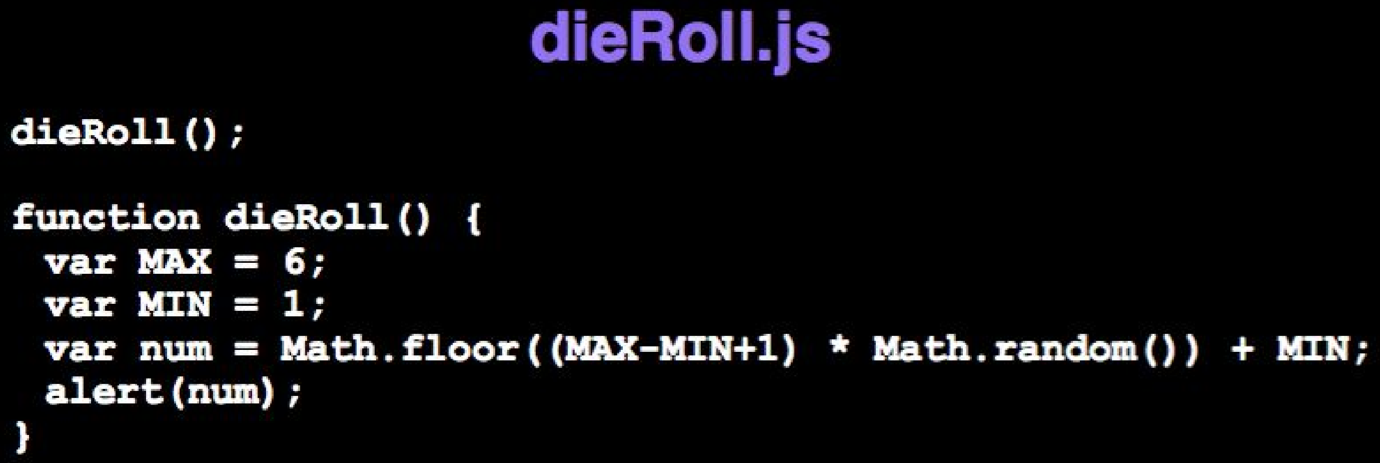
dieRoll.js

```
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>
```

Function Type 1: The Void Function

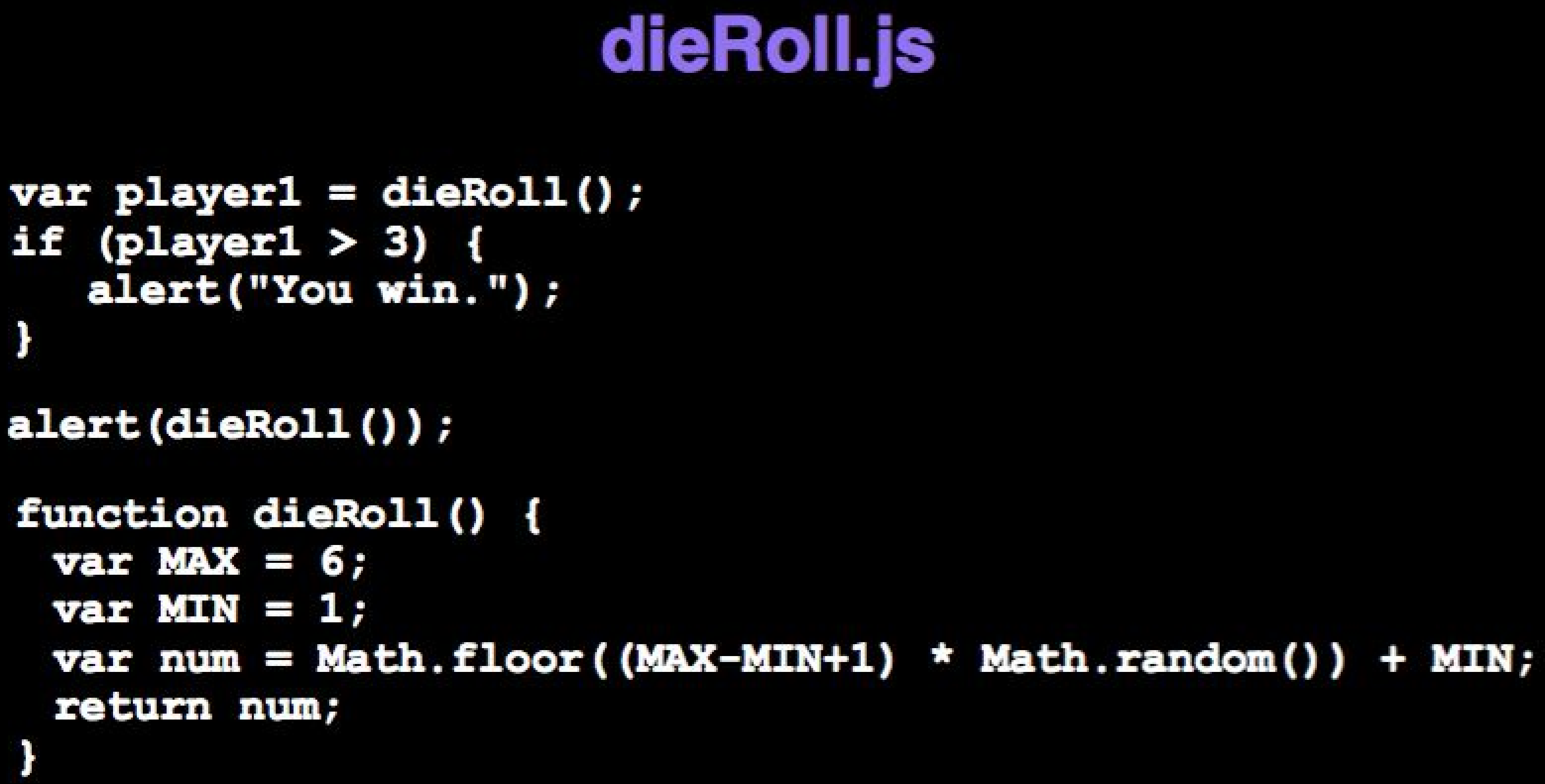


dieRoll.js

```
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 and 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 “heads” or “tails”.

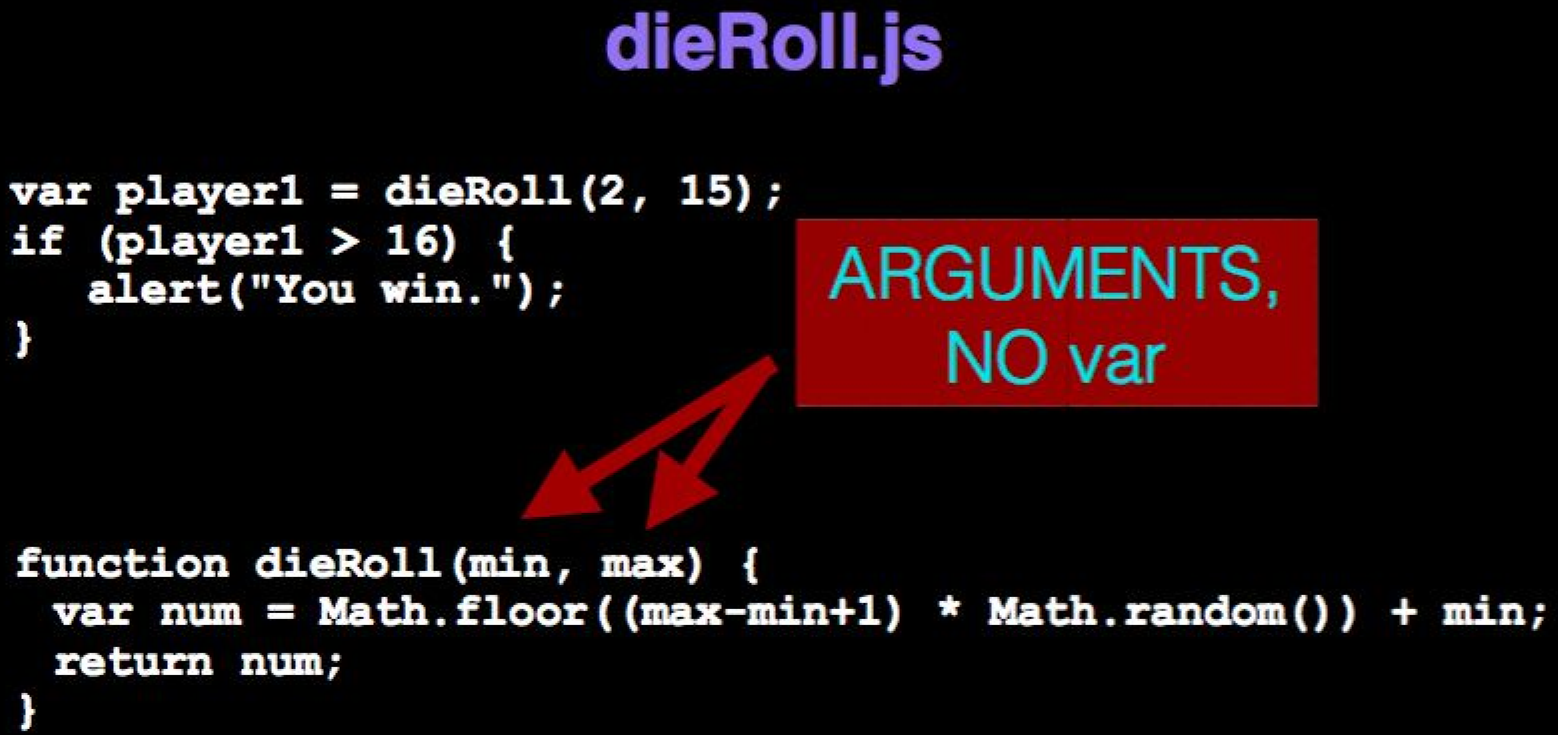
Create a folder called CoinFlip, create an HTML file called coinFlip.html that calls coinFlip.js

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

BACK TO DIEROLL: What if you didn't always want to roll a 6-sided die?



dieRoll.js

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


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

ARGUMENTS, NO var

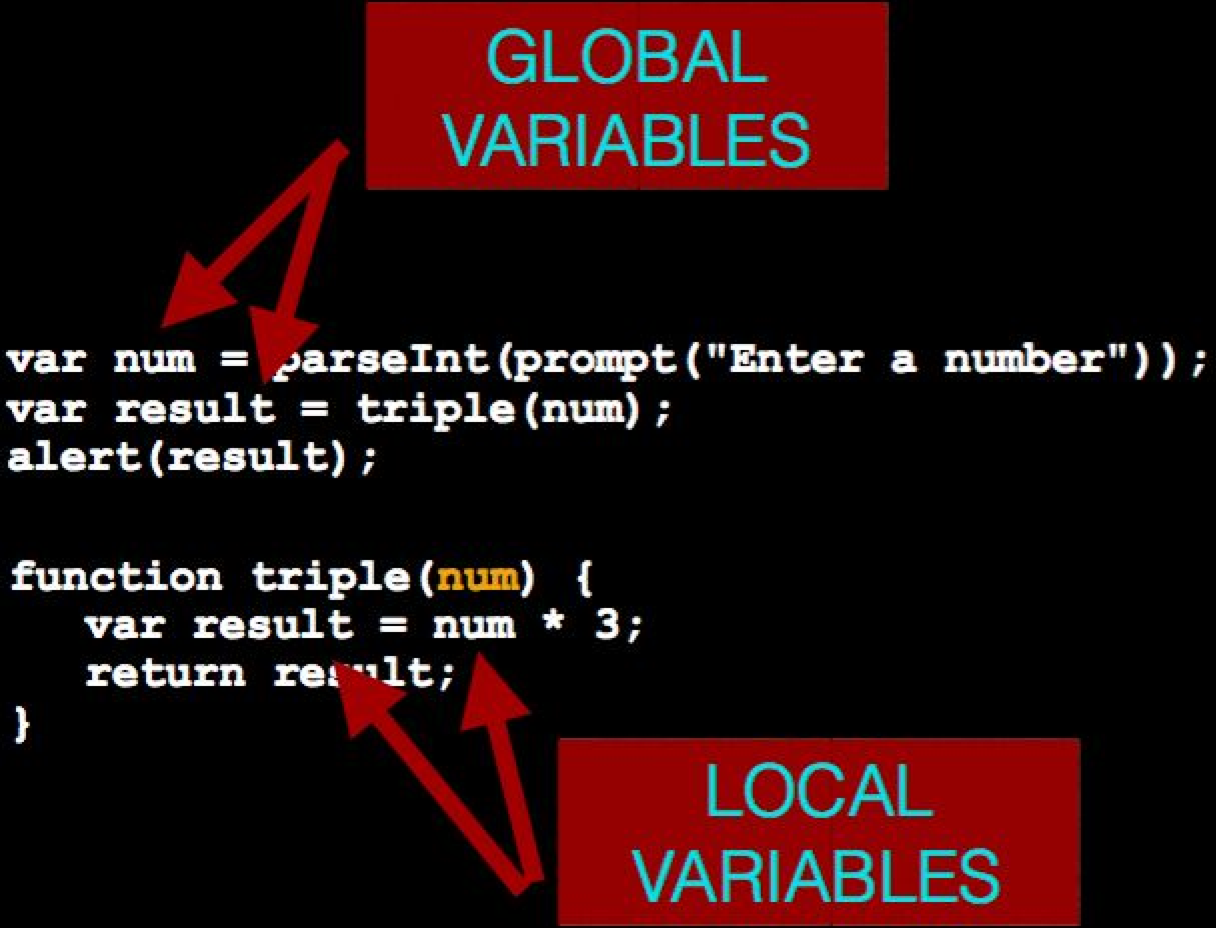
The function dieRoll() is now a return function that accepts arguments.

DO

With your partner, write a program that uses a function called triple. Triple should take on argument and return three times it’s value.

Create a folder called Triple, create an HTML file called triple.html that calls triple.js

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, then the local variables will win!

Four types of functions:

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

DO

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

Create a folder called Max3, create an HTML file called max3.html that calls max3.js

max3.js

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

//max3 will return the largest of 3 arguments
function max3(a, b, c) {
    var max = a; //assuming it is the largest
    if (b > max) {
        max = b;
    }
    if (c > max) {
        max = c;
    }
    return max;
}
```

DO

CHALLENGE TIME 2:
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.

Create a folder called sign, create an HTML file called sign.html that calls sign.js

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;
}
```

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;
}
```

Goals:

Goal 1: Name and understand the 4 different kinds of functions.

Goal 2: You will know how to implement the 4 different kinds of functions.

Vocabulary

return function
void function
arguments

Code

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