GENERAL ASSEMBLY 6.2

Front-End Web Development

Albert Penticoss
Web Designer, University of Melbourne

Start your engines!

- Connect to wifi
- Open your text editor
- Open GitHub desktop and GitHub website
- Open schoology

Final Project Milestone 2

- Draft HTML/CSS
- Pseudo code for JavaScript
- Due week 7 - Wednesday 2nd March

Reminder - Graduation requirements

- To graduate from the course, students must:
  - 1. complete at least 80% of assigned homework
    - 8 assignments so you can only miss one!
  - 2. attend 80% of classes - so you can miss up to a maximum of four
  - 3. satisfactorily complete the final project (as determined by the course instructor and stated on the class syllabus).

What are we going to cover today?

- Mid course feedback - 15min
- Lesson 9 Review - 30min
- Finish jQuery DOM selector exercise - 15min
- Break - Dinner's on GA! - 15min
- JS Functions - 60min
- Revisiting jQuery - more jQuery stuff you can use - 15min
- Assignment 4 - 30min

Exercise – jQuery Basics Review

KEY OBJECTIVES

Practice jQuery selection

AGENDA

15 mins

1. Download lesson exercise 1 starter code from schoology
2. Follow instructions in the exercise.js file

DELIVERABLE

dom selectors

RESOURCES

Starter code
Code editor

LESSON 9 REVIEW

FEWD - LESSON 9

VARIABLES AND CONDITIONALS

FEWD - LESSON 9

GITHUB PAGES SETUP

GitHub Pages

- GitHub pages are a way to host a website on GitHub
- We are going to use this for our Final Project
- We are also going to use this today for our exercises! HOORAY

http://ipadisthin.github.io/fewd/

CSS POSTION PROPERTY

Position property

- Information in the position slides is from: http://www.w3schools.com/css/css_positioning.asp

Position property

- The position property specifies the type of positioning method used for an element.
- There are four different position values:
  - static
  - relative
  - fixed
  - absolute
- Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

Position: static;

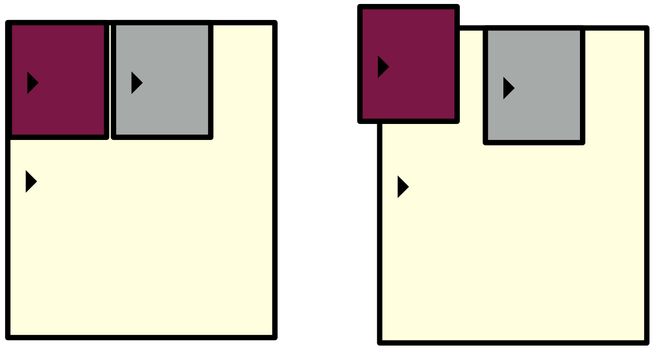
position: static;

- HTML elements are positioned static by default.
- Static positioned elements are NOT affected by the top, bottom, left, and right properties.
- An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page.

Position: relative;

position: relative;

- An element with position: relative; is positioned relative to its normal position.
- Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.



position: relative;

top: -10px;

right: -10px;

Position: fixed;

position: fixed;

- An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.
- A fixed element does not leave a gap in the page where it would normally have been located.

Position: fixed;

position: absolute;

- An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).
- However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.
- Note: A "positioned" element is one whose position is anything except static.

z-index

- If two elements overlap the one that comes second in the HTML will usually be on top.
- We can use z-index to change the stacking
- z-index is a numerical value and can be negative
- eg:

```
z-index: 100;
z-index: -1;
```

FEWD - LESSON 9

VARIABLES

Variables

What are variables?

- We can tell our program to remember values for us to use later on.
- The action of saving a value to memory is called assignment
- The entity we use to store the value is called a variable

Variables

- The action of getting the value from a variable is called accessing the variable
- We will use all the above techniques to store values into variables, and generate new values using existing variables

Variables Declaration

- Declaration: var age;
- Assignment: age = 21;
- Both at the same time: var age = 21;
- Note in JavaScript using var to declare the variable is optional, but if you don’t declare the variable with var when inside a function, that variable will have global scope! Sometimes you don’t want that. Declare your variables to avoid confusion.

Variable Re-Assignment

- var name = "Jo";
- name = Amir;
- Note: name is now Amir.

Variable Conventions

- Variables start with a lower case letter
- If they contain multiple words, subsequent words start with an upper case letter.

```
var numberOfStudents = 10;
```

- This is called camel case (it has humps!)

Variables & Data Types

- What can you store in a variables?

Variables & Data Types

- What can you store in a variables?
- The types of different values we support include:
  - String text
  - Number: int (whole numbers) or float numbers (numbers with decimal point)
  - Boolean: true or false

Strings

- Stores textual information
- String literal is surrounded by quotes

```
var myString "How is the weather today?”;
```

Double vs Single quoted strings

- You can use either single or double quotes to define a string.
- If you use single quotes you can use double quotes inside your string and vice versa.

```
'They "purchased" it'
"It's a beautiful day"
```

Escaping

- If you use a special character in your string you need to escape it with the \ character

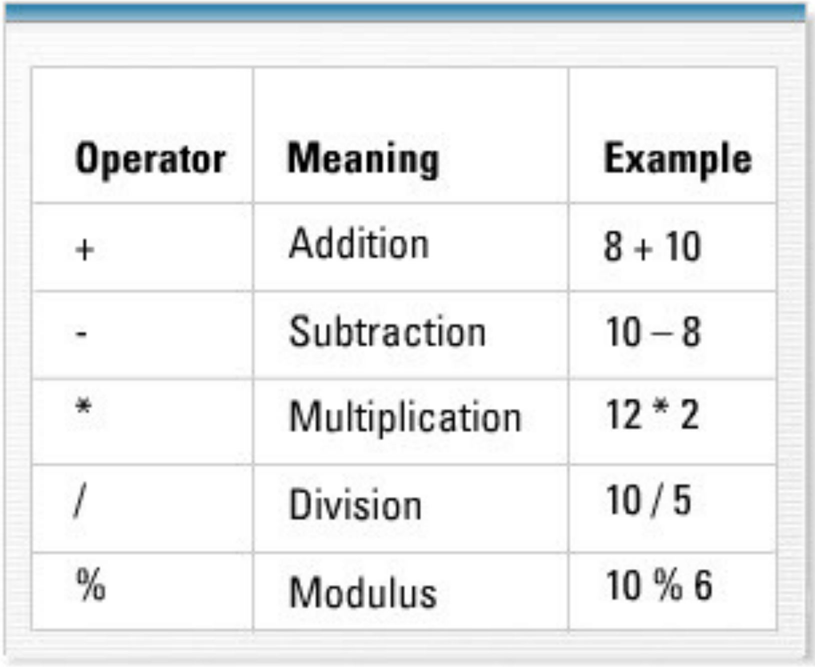
```
"They \"purchased\" it"

'It\'s a beautiful day'
```

Numbers

- Represent numerical data
  - int: 42
  - float: 3.14159265
- Signed
  - int: +6
  - float: -8.2

Arithmetic In JavaScript

- You can perform arithmetic on number data types



Operator	Meaning	Example
+	Addition	8 + 10
-	Subtraction	10 – 8
*	Multiplication	12 * 2
/	Division	10 / 5
%	Modulus	10 % 6

Conversion: String To Number

- You can convert a string into a number:

```
parseInt(“4”);
parseFloat("3.14159");
```

Conversion: Number To String

- You can convert a number into a string:

```
var number = 4;
number.toString(); or number + "";
```

number now equals “4”

GUEST SPEAKER

Exercise – Score Keeper

KEY OBJECTIVES

Use variables

AGENDA

25 mins

1. Download lesson exercise starter code from schoology
2. Code along with instructor

DELIVERABLE

Score card

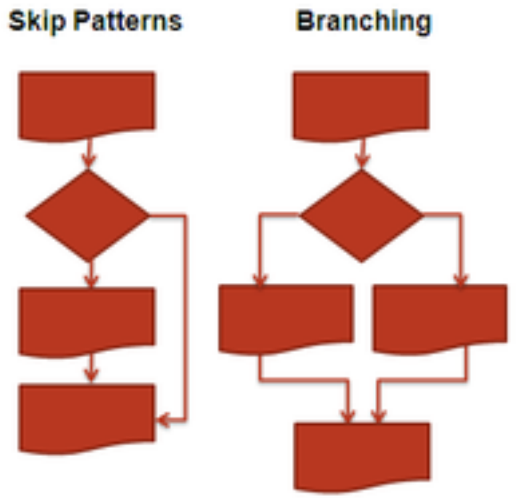
RESOURCES

Starter code
Code editor

FEWD - LESSON 9

CONDITIONALS

Conditionals



Making Decisions

- It's either TRUE or FALSE (like booleans)
- If you are greater than 18, you are an adult

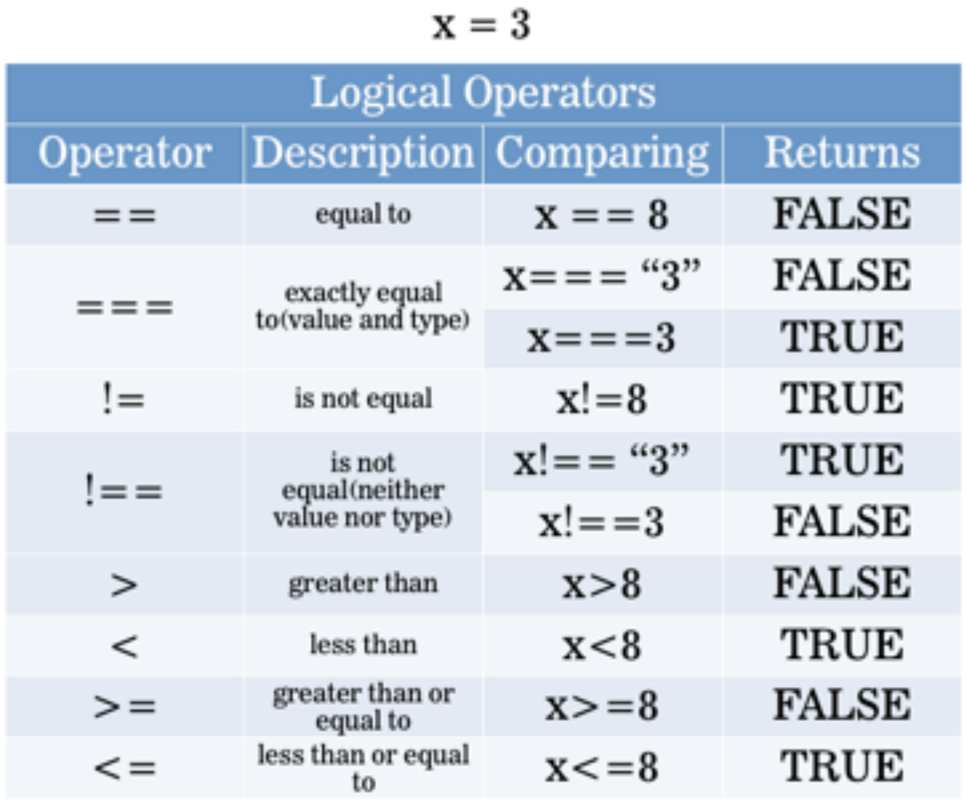
```
if (age > 18){
    document.write("You are an adult");
}
```

Comparisons - Equality

- Are two things equal?
- == equal to value
- === equal to value and type

```
10 == 10 //true

10 == 5 //false

"hi" == "hi" //true


10 == “10” //true

10 === “10” //false
```

Logical Operators



x = 3

Logical Operators			
Operator	Description	Comparing	Returns
==	equal to	x == 8	FALSE
===	exactly equal to(value and type)	x=== "3"	FALSE
		x===3	TRUE
!=	is not equal	x!=8	TRUE
!==	is not equal(neither value nor type)	x!== "3"	TRUE
		x!==3	FALSE
>	greater than	x>8	FALSE
<	less than	x<8	TRUE
>=	greater than or equal to	x>=8	FALSE
<=	less than or equal to	x<=8	TRUE

Conditional Syntax

```
if(condition is true) {
    //Do something
}
```

Conditional Syntax

```
if(condition is true) {
    //Do something
}else{
    //Do something else
}
```

Conditional Syntax

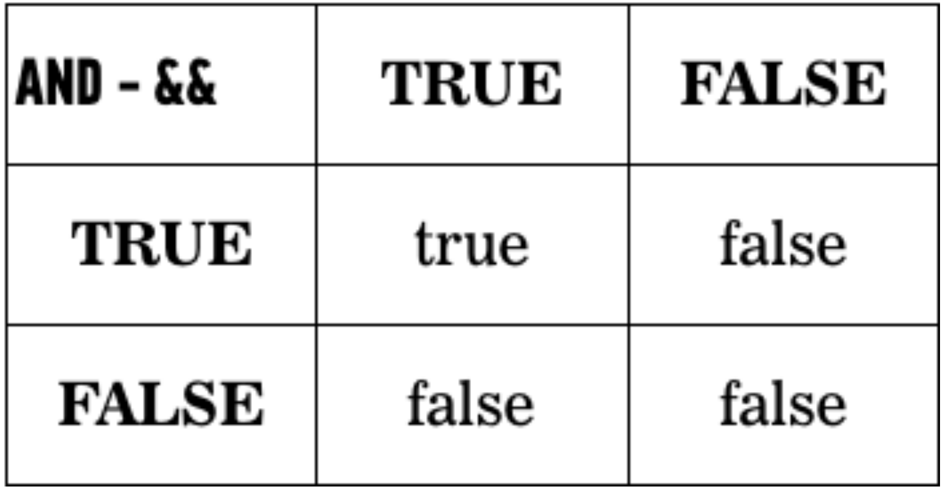
```
var topic = "JS";
if (topic == "JS") {
    console.log("You're learning JavaScript");
} else if(topic == "JavaScript") {
    console.log("You're still learning JavaScript");
} else {
console.log("You're learning something else");
```

Multiple Conditions

```
if (name == "GA" && password == "YellowPencil"){
   //Allow access to internet
}
```

The Truth Table - AND

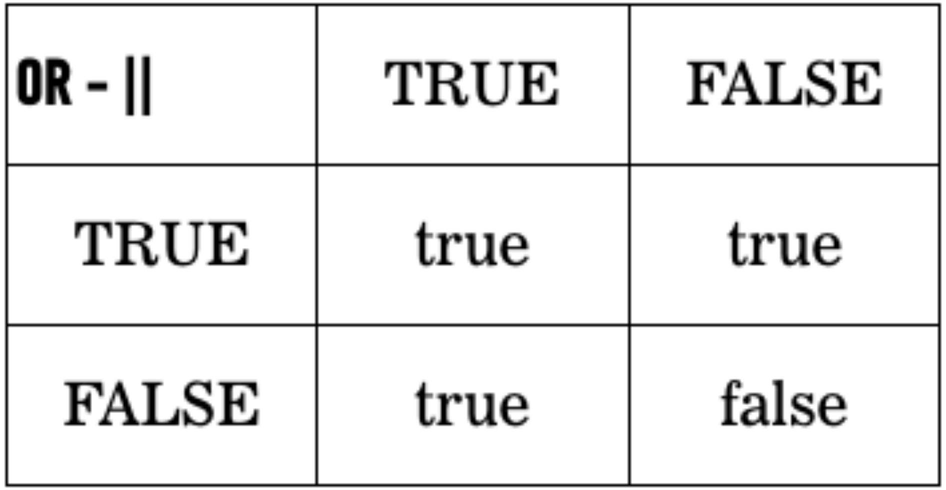
```
if (name == "GA" && password
== "YellowPencil"){
   //Allow access to internet
}
```



AND - &&	TRUE	FALSE
TRUE	true	false
FALSE	false	false

The Truth Table - OR

```
if (day == "Monday" || day == "Wednesday"){
    //We have class today
    }
```



OR - ||	TRUE	FALSE
TRUE	true	true
FALSE	true	false

Exercise - Compare That

KEY OBJECTIVES

Use comparison operators

AGENDA

20 mins

1. Download lesson exercise starter code from schoology
2. Code along with instructor

DELIVERABLE

Value compare

RESOURCES

Starter code
Code editor

Exercise - Blackout

KEY OBJECTIVES	**AGENDA**
Use variables and comparison operators	*15 mins* — 1. Download lesson exercise starter code from schoology 2. Code along with instructor
DELIVERABLE	**RESOURCES**
Lights on/off function	Starter code Code editor

FEWD - LESSON 10

FUNCTIONS

FIRST LETS FINISH DOM SELECTION EXERCISE

Exercise – jQuery Basics Review

KEY OBJECTIVES

Practice jQuery selection

AGENDA

15 mins

1. If you didn’t do so last week, download lesson 9 exercise 1 starter code from schoology
2. Follow instructions in the exercise.js file

DELIVERABLE

dom selectors

RESOURCES

Starter code
Code editor

FEWD - LESSON 10

FUNCTIONS

What is a function?

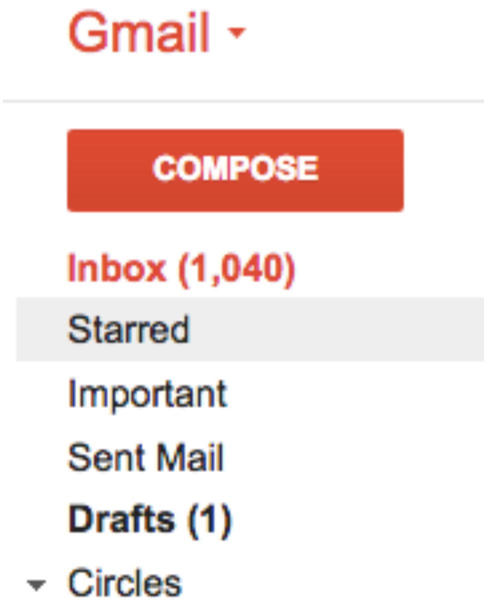
Discuss!

When do I use a function?

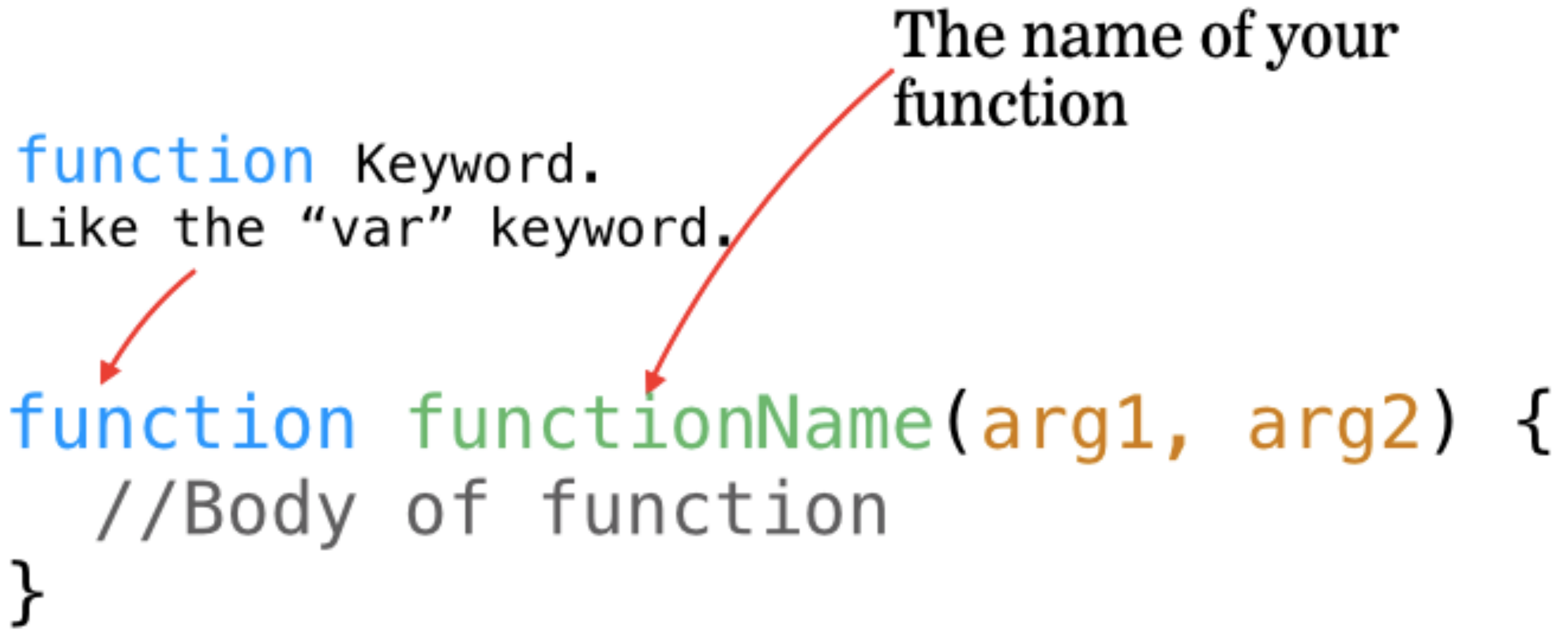
Use a function for repeatable blocks of code or tasks that you need to do more than once and possibly with different values.

For example the click of different buttons/links on the page may do the same or similar function, let's say we have several buttons for filtering a list, the filter code is the same the only thing that is different is the filter value.

e.g. Gmail >



Function Syntax



```
function functionName(arg1, arg2) {
  //Body of function
}
```

Function Calls

```
function helloWorld() {
  console.log("Hello Functions");
}

helloWorld(); //Prints “Hello Functions to the
console.
```

The brackets execute the function. Try calling the function without them to see what happens.

Function Arguments

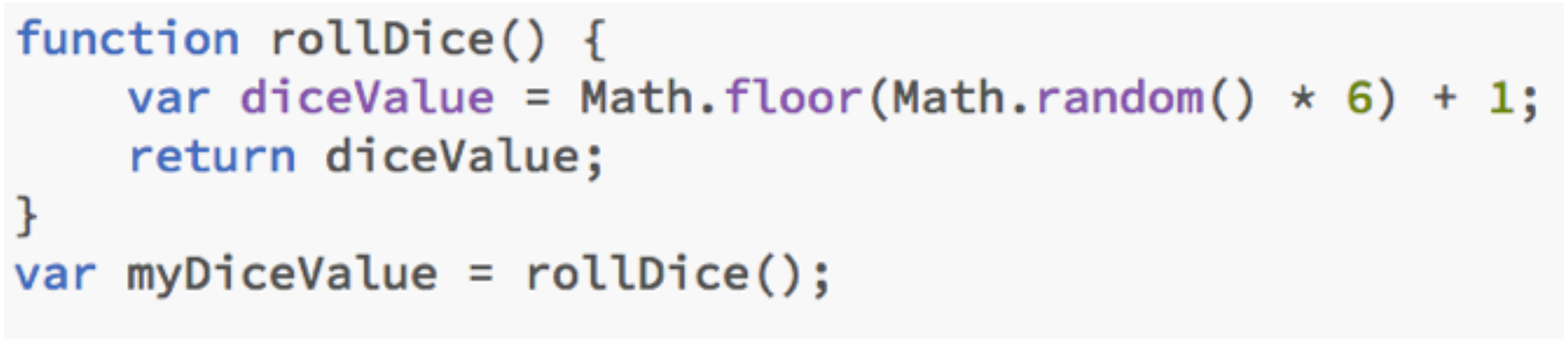
```
function addAndPrint(num1, num2) {
  var sum = num1 + num2;
  console.log(sum);
}
```

```
addAndPrint(1, 2); // Result is 3
```

```
addAndPrint(8, 2); // Result is 10
```

Return Functions

- Some functions return a value



```
function rollDice() {
    var diceValue = Math.floor(Math.random() * 6) + 1;
    return diceValue;
}
var myDiceValue = rollDice();
```

https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Statements/return

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random

Organising Functions

- Where do I put my functions

Exercise – Scorekeeper

KEY OBJECTIVES

Update Scorekeeper to use a function

AGENDA

15 mins

1. Open your JSFun website in brackets
2. Code along with instructor

DELIVERABLE

Refined js for scorekeeper

RESOURCES

Starter code
Code editor

FEWD - LESSON 10

MORE JQUERY

Let’s make

A mega menu using .hover( )

A slide in menu using .animate()

And if we have time a box that fades in and out using .fadeIn( ) .fadeOut ( )

ASSIGNMENT 4