**Introduction to Functions**

**Principles of JavaScript**

Take words that have meaning, and the JavaScript engine interprets those words to tell our computer what to do.

When we run code line by line, we are storing stuff in memory.

When JavaScript find the label of function, store that label to use it later with the variable declared.

As soon we start running our code, we create a *global execution context*

* Thread of execution (parsing and executing the code line after line).
* Live memory of variables with data (known as a Global Variable Environment)

The thread in JavaScript

* Single threaded (one thing at a time)
* Synchronous execution (for now) – Line by line

JavaScript always set a value even when we don’t declare it (undefined)

Local execution context, space where we run our code of the function

When you run a function, you create a new execution context comprising:

* The thread of the execution (We go through the code in the function line by line)
* A local memory (Variable environment) where anything defined in the function is stored.

**We keep track of the functions being called in JavaScript with a Call stack**

**High Order Functions**

Follows the principle when we have not decided exactly what our functionality is until we run our function.

Takes in a function or passes out a function

**Functions in JavaScript = first class objects**

They can co-exist with and can be treated like any other JavaScript object

1. Assigned to variables and properties of other objects

2. Passed as arguments into functions

3. Retuned as values from functions