Call and Apply Methods

A function can use an unspecific object called this and use that objects properties in its body

const me = { fname: ‘Dean’, lname: ‘Carver’};

function sayMyName() {

console.log(`I’m ${this.fname} ${this.lname}. Hello!`);

}

Memoization

Allows results of a function call to be stored in an array cache

IIFE

A function that is invoked upon definition

(function() {

…

} ()

This is especially good when using strict mode, incase code from someone else doesn’t fit within strict mode (function() { ‘use strict’; /\* the rest of ALL of the code goes here})();

Temporary variables

Similar syntax to IIFE. Allows a variable to be accessible outside of its scope and does not exist outside of its scope

(()=>{

Const temp = 3;

})();

Console.log(temp) // temp not defined

Redefine functions

Functions can be redefined, even within themselves

function party(){

console.log('Wow this is amazing!');

party = function(){

console.log('Been there, got the T-Shirt');

}

}

The first time this function is executed, it will run Wow this is amazing, and anytime after Been there got the T-shirt

Recursive Functions