Week 2

Monday, April 26, 2021 2:36 PM

Low-level programming languages are assembly language or machine code.

High-level programming languages are C, C++ and Java.

Scripting languages, they are also high-level, but they are translated into machine code at run time.

Well commented code is the hallmark of a ninja programmer

```
// Inline comments
/*
Multi line comments
*/
```

A block is a series of statements that are collected together inside curly braces:

```
{
//statement 1
//statement 2
}
```

An operator applies an operation to a value, which is known as the

Variables are used in programming languages to refer to a value stored in memory

You can even declare and assign multiple variables at the same time if you separate them with commas:

```
let x = 3, y = 4, z = 5;
```

When declaring variables if you declare a variable and you redeclare a variable with the same name without using let or const you will make it global variable.

```
Const variable = 5;
{
    Variable = 6;
}

alert(variable)
// 6 will be printed
```

The best convention to follow on naming in JavaScript is with Camel case.

if you assign a non-primitive value to a variable, then this is done by reference, so any changes that are subsequently made will affect all references to that object:

```
const c = { value: 1 };

let d = c; // c.value = 1, d.value = 1

d.value = 2; // c.value = 2, d.value = 2
```

Temple Literals

They also allow interpolation of JavaScript code. This means that a JavaScript expression can be inserted inside a string and the result will be displayed, as can be seen in the examples below:

```
const name = `Siri`;
`Hello ${ name }!`;
<< 'Hello Siri!'

const age = 39;
`I will be ${ age + 1 } next year`;
<< 'I will be 40 next year'
```

NaN means Not a Numbers!

Type coercion happens when the operands of an operator are of different types.

```
Delete array[3]
```

Deleting on arrays that way will delete the value inside but if not the space, so if you try to call it, it will return undefined.

Destructuring an array is the concept of taking values out of an array and presenting them as individual values.

Destructuring allows us to assign multiple values at the same time, using arrays:

```
const [x,y] = [1,2];
```

The spread operator is three dots, ... that are placed in front of an array, with the effect of spreading out the elements of that array. This can be used to spread the elements of two arrays and put them together in a new array, like so:

```
avengers = [ ...avengers, ...['Hulk','Hawkeye', 'Black
➥ Widow'] ];
<< ['Captain America', 'Iron Man', 'Thor', 'Hulk',
➥ 'Hawkeye', 'Black Widow']
```

// I love Sets!
A set is a data structure that represents a collection of unique values, so it cannot include any duplicate values

An empty set is created using the new operator and Set() constructor:

```
const list = new Set();
```

A memory leak occurs when a program retains references to values that can no longer be accessed in its

memory. This means that memory is being used to store values that are no longer required by the program, effectively wasting system resources.

An empty map object can be created using the new operator and Map() constructor:

```
const romanNumerals = new Map();
```

Ternary Operator
A shorthand way of writing an if ... else statement is to use the ternary operator, ?, which takes three operands in the following format:

condition ? (//code to run if condition is true) : (//code to run if condition is false)

The ternary operator can make your code more succinct, but can also make it harder to read, so think carefully before using it.

Something I often hear is that I need to make my code as readable as possible, both for me and for those who will read my code.

ES6 introduced an improved iterator function for arrays called a for-of loop that uses a slightly different syntax:

```
for(const value of avengers){ console.log(value); }
<< 'Black Widow'
<< 'Captain America'
<< 'Hawkeye'
<< 'Iron Man'
<< 'Quicksilver'
<< 'Scarlet Witch'
```

I use to hate these for loops… such in what mistake I was…

Page 126 explains how to loop over maps

REMEMBER THIS!
Don't Repeat Yourself, or DRY, is a principle of programming that specifies that every part of a program should only be written once. This avoids duplication and means there's no need to keep multiple pieces of code up to date and in sync.

If too many arguments are provided when a function is invoked, the function will work as normal and the extra arguments will be ignored

Rest Operator
A much better option is to use the rest operator. This was introduced in ES6 and can be used to deal with multiple arguments by creating an array of arguments that are available inside the body of the function. To use the rest operator, simply place three dots in front of the last parameter in a function declaration. This will then collect all the arguments entered into an array.

```
function rest(...args){
 for(arg of args){
        console.log(arg);
```

```
}
}

rest(2,4,6,8);
<< 2
4
6
8
```

Function Hoisting
Hoisting is the JavaScript interpreter's action of moving all variable and function declarations to the top of the current scope, regardless of where they are defined. Functions that are defined using a function declaration are automatically hoisted, meaning they can be invoked before they have been defined.

Callbacks
This means that functions can also be given as a parameter to another function. A function that is passed as an argument to another is known as a callback.

The various iterator functions can be used in combination to create some powerful transformations of data stored in arrays. This is achieved by a process called chaining methods together.

```
[1,2,3].map( x => x*x ).reduce((acc,x) => acc + x );
 << 14
```