**Javascript Notes**

If the JavaScript loads first and it is supposed to affect the HTML that hasn't loaded yet, there could be problems. Placing JavaScript near the bottom of an HTML page is one way to accommodate this dependency.

JavaScript is case sensitive

 To signify that the value is a string, enclose it in single or double quote marks.

The words true and false are special keywords that don't need quote marks.

 Everything in JavaScript is an object and can be stored in a variable

Operators:  
**Addition, Subtraction, Multiplication, Division, Assignment, Strict equality(===):** It returns a true/false (Boolean) result.

**Not, Does-not-equal->(**!, !==)

The **nullish coalescing (??)** operator is a logical operator that returns its right-hand side operand when its left-hand side operand is [null](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/null) or [undefined](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/undefined), and otherwise returns its left-hand side operand.

we create a simple function which takes two numbers as arguments and multiplies them:

function multiply(num1, num2) {

let result = num1 \* num2;

return result;

}

The return statement tells the browser to return the result variable out of the function so it is available to use. This is necessary because variables defined inside functions are only available inside those functions. This is called **variable scoping.**

**Event handler attaching to html element:**

document.querySelector("html").addEventListener("click", function () {

alert(“Learning javascript");

});

Here we select the [<html>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/html) element. We then call its addEventListener() function, passing in the name of the event to listen to ('click') and a function to run when the event happens.

The function we just passed to addEventListener() here is called an *anonymous function*, because it doesn't have a name.

**Arrow function:**

There's an alternative way of writing anonymous functions, which we call an *arrow function*.

An arrow function uses () => instead of function ():

document.querySelector("html").addEventListener("click", () => {

alert("Learning javascript ");

});

Null is a special value in JavaScript that refers to the absence of a value.

JavaScript interpreters actually use a technique called **just-in-time compiling** to improve performance; the JavaScript source code gets compiled into a faster, binary format while the script is being used, so that it can be run as quickly as possible. However, JavaScript is still considered an interpreted language, since the compilation is handled at run time, rather than ahead of time.

**Creating object using constructor**

A screen shot of a computer code

Description automatically generated

Adding property and methods to constructor

A computer screen with text

Description automatically generated

A black screen with white text

Description automatically generated

Function in Js

so myfunction refers to a function while myfunction() refers to an object

Local variables are created when a function starts, and deleted when the function is completed.

object are variables which can contain many values

The name:values pairs in JavaScript objects are called properties

A method is a function stored as a property.

The **optional chaining ‘?.’**is an error-proof way to access nested object properties.

The **optional chaining (?.)** operator accesses an object's property or calls a function. If the object accessed or function called using this operator is undefined or null, the expression short circuits and evaluates to undefined instead of throwing an error.

A screenshot of a computer

Description automatically generated

**Javascript datatypes:**

//JavaScript has 8 Datatypes

1. String

2. Number

3. Bigint

4. Boolean

5. Undefined

6. Null

7. Symbol

8. Object

Objects in Javascript

The object data type can contain:

1. An object

2. An array

3. A date

In JavaScript, almost "everything" is an object.

Booleans can be objects (if defined with the new keyword)

Numbers can be objects (if defined with the new keyword)

Strings can be objects (if defined with the new keyword)

Dates are always objects

Maths are always objects

Regular expressions are always objects

Arrays are always objects

Functions are always objects

Objects are always objects

All JavaScript values, except primitives, are objects.

A primitive value is a value that has no properties or methods.

string

number

boolean

null

undefined

symbol

bigint

const x = person;  // Will not create a copy of person.

The object x is not a copy of person. It is person. Both x and person are the same object.

Any changes to x will also change person, because x and person are the same object.

The delete operator is designed to be used on object properties. It has no effect on variables or functions.

Variables in Javascript

Names must begin with a letter.

Names can also begin with $ and \_

A variable declared without a value will have the value undefined.

You cannot re-declare a variable declared with let or const.but can redeclare a variable with var.

Redeclaring a variable inside a block will also redeclare the variable outside the block: for var.

Jquery:javascrript library

In jQuery $("p"); means "select all p elements".

let:block scope

var:global scope

let and const must be declared before use.

let and const does not bind to this.

With let, redeclaring a variable in the same block is NOT allowed

Variables defined with let are also hoisted to the top of the block, but not initialized.

using a let or const variable before it is declared will result in a ReferenceError

const can not be reassigned

const can be used when declaring a new array,new function,new object,new RegExp

It does not define a constant value. It defines a constant reference to a value.

we can still change elemnts of constant array,properties of constant object

Redeclaring an existing var or let variable to const, in the same scope, is not allowed

Javascript numbers are always one type:

double (64-bit floating point).

Any variable can be emptied, by setting the value to undefined.

The type will also be undefined.

An empty value has nothing to do with undefined.

An empty string has both a legal value and a type.