**JAVASCRIPT**

**Programming language**

3 kinds of programming language

1. Machine language

\*Machine language consists of set of binary instructions that are either 0’s and 1’s.

\*Because computers can only read machine instructions in binary digits i.e, 0’s and 1’s.

1. Assembly level language

\*We need to translate to convert assembly level language to machine language since machine can only understand machine level instructions.

\*Assemblers are translators which are used to translate the code.

1. High level language

\*To covert high-level language into low level language a compiler is necessary as a translators

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

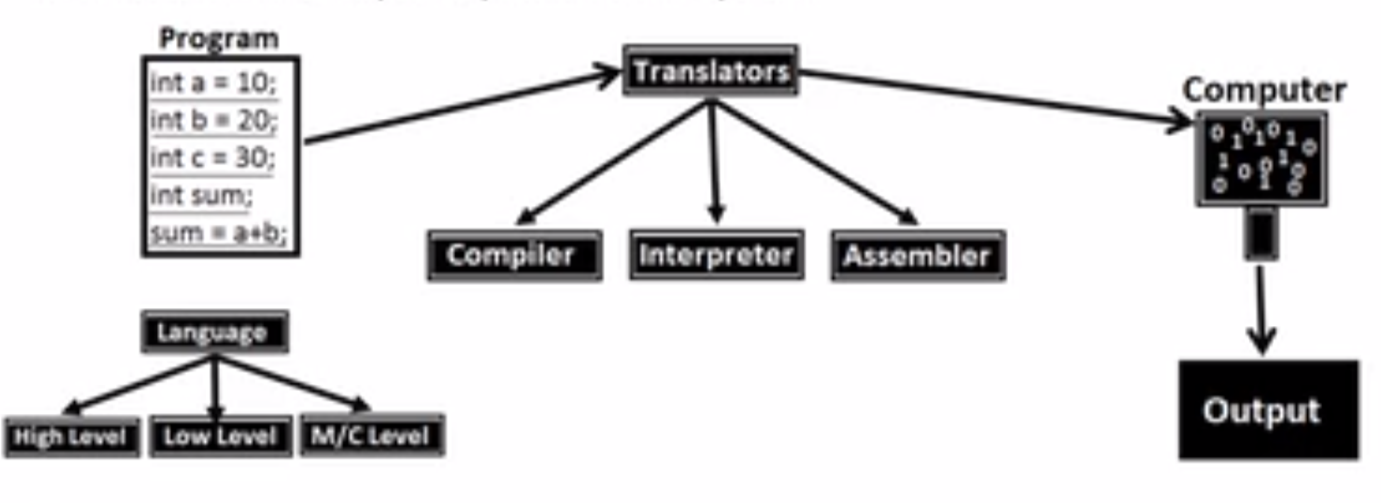
**Translators**

🡪What is a translator?

3 different types of translators

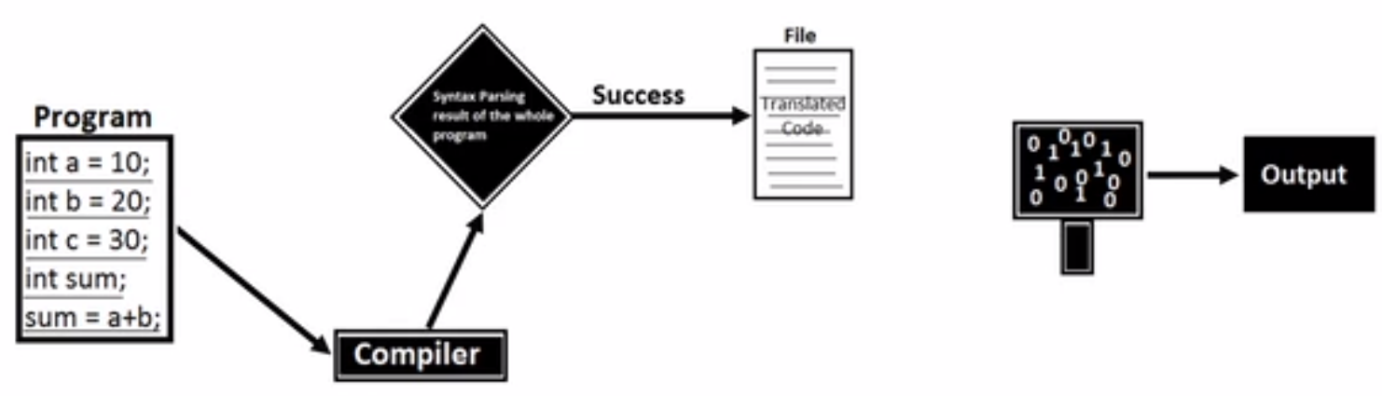
1. Compilers
2. Assemblers
3. Interpreters

🡪 What is a program?



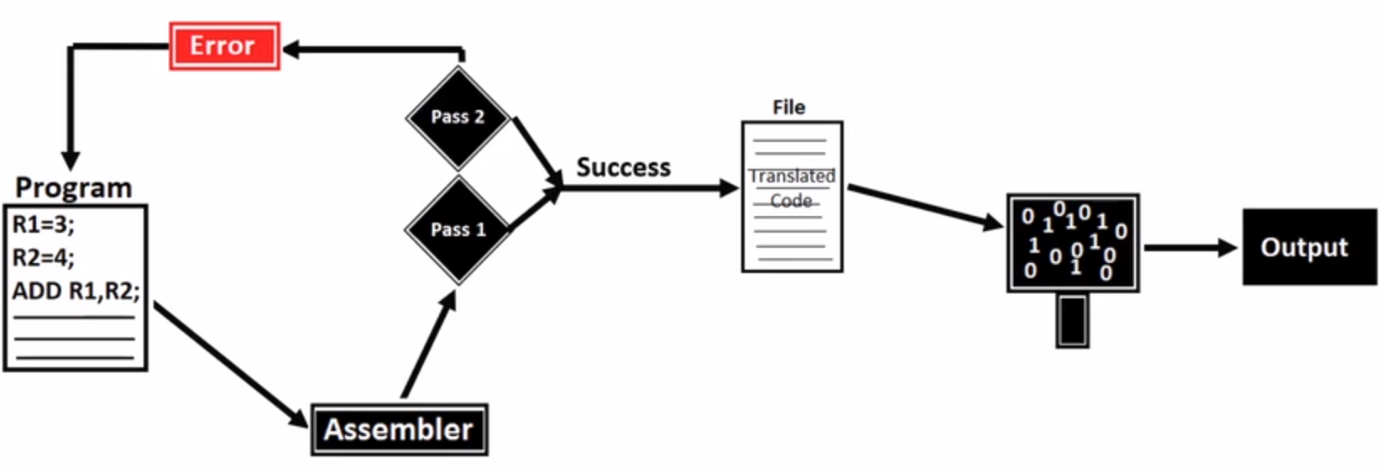
--------------------------------------------------------------------------------------------------------------------------------------

How Compiler works?



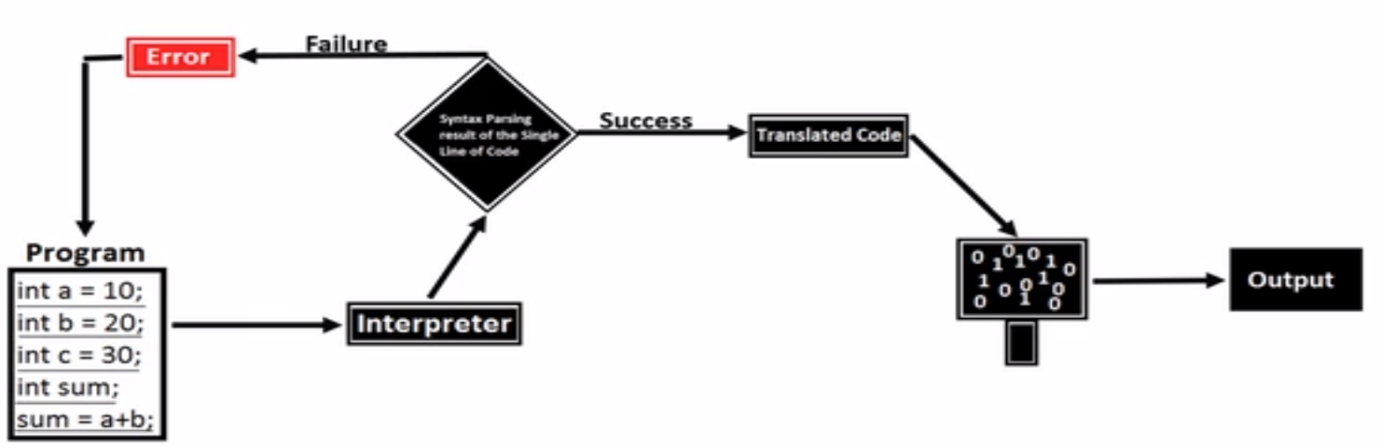
--------------------------------------------------------------------------------------------------------------------------------------

How Assemblers works?



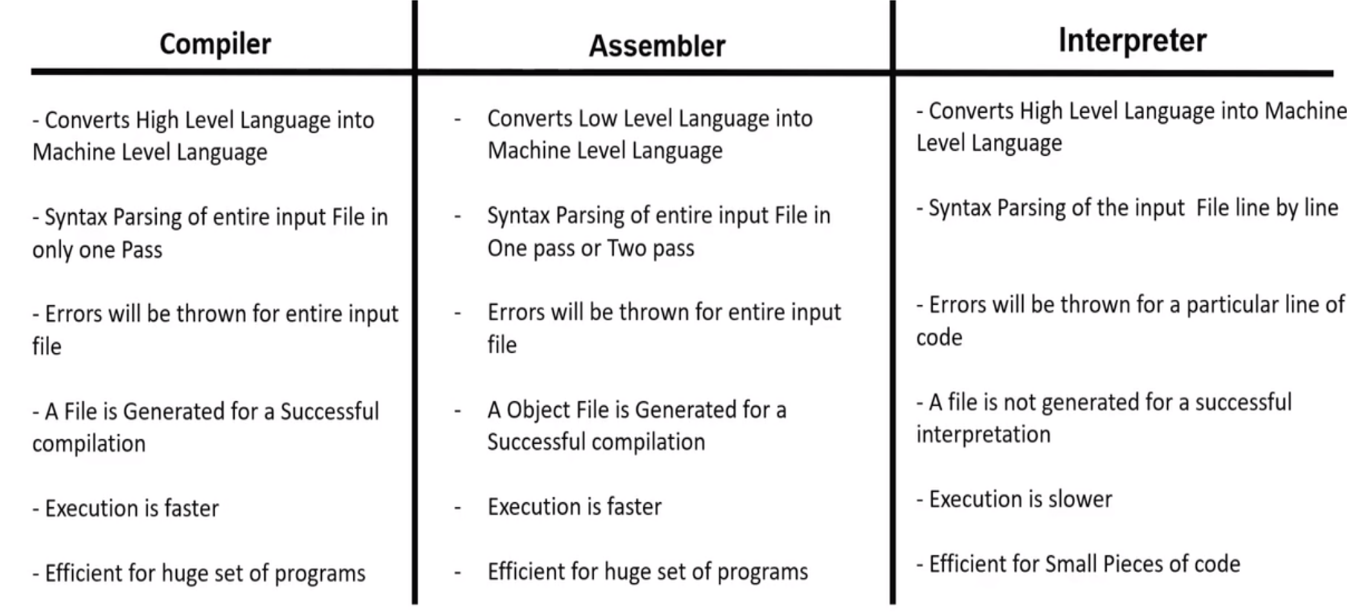
--------------------------------------------------------------------------------------------------------------------------------------

How Interpreter works?



--------------------------------------------------------------------------------------------------------------------------------------

Difference between Compiler, Assembler, Interpreter



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**History of JavaScript**

* 1990’s

Internet for advertisements.

HTML author 🡪develop static web pages for advertisements and post it on internet so people can access.

* 1993

As Browser was necessary to view the webpages, “Netscape” company created their own browser called “Netscape Navigator”.

To create completion Microsoft created its own browser called “Internet Explorer” and made it free whereas the Netscape Navigator was paid browser.

If people wanted to use Netscape Navigator, they have to buy whereas people can use Internet Explorer as a free browser. So people opted for Internet Explorer.

* Netscape company hired “Brenden Eich” to develop a new programming language to make webpages interactive. In 10 days a new programming language was ready and named it as “MOCHA”. It was renamed as “LIVESCRIPT” during the launch. But for the marketing purpose it was again renamed it as “JAVASCRIPT”
* In 1995 JavaScript was released in “Netscape Navigator2”. So in order to use JavaScript people need to buy Netscape Navigator.

But later Microsoft reverse engineered and built the similar language and named it as “Jscript”.

* Netscape went to **ecma**  to create standards. Ecma used JavaScript and created a standard and named it as ecmascript (es)

1997 – es 1 was launched

1999 – Netscape got sold and took the source code and made it as open source. The reincarnation of the Netscape Navigator is “Firefox”

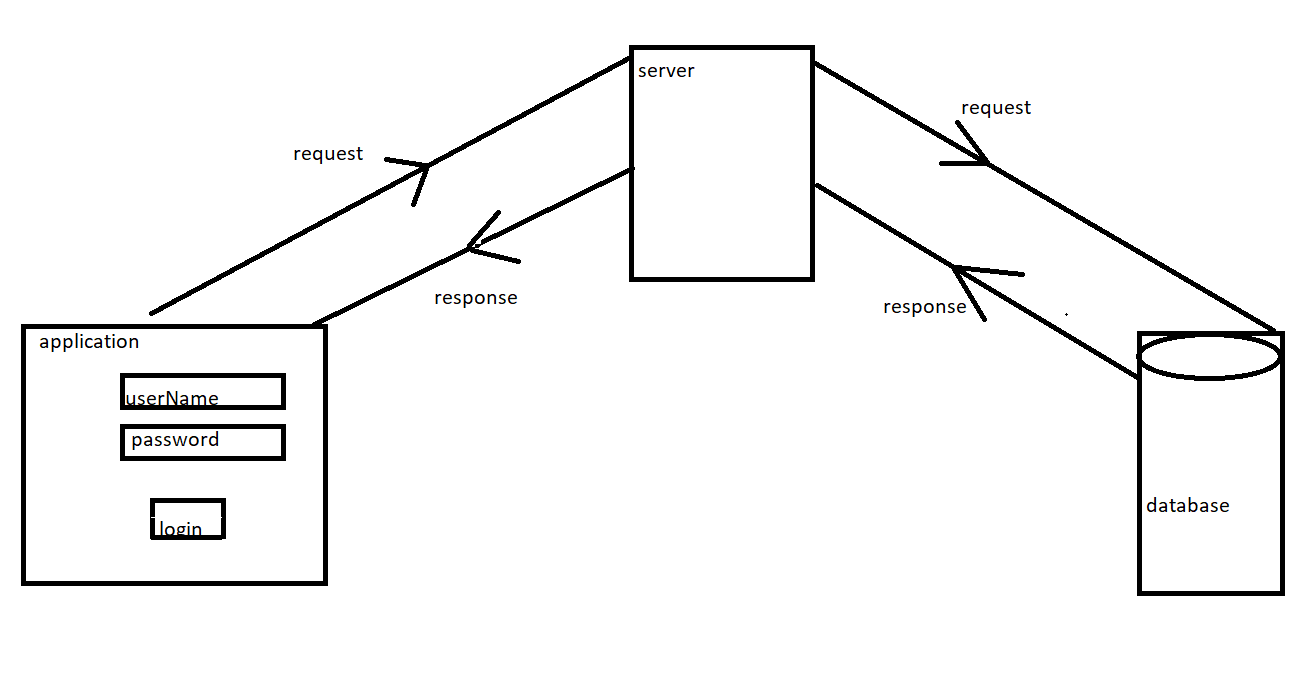
2009 - es5 was launched (can build desktop app, websites app, server app)🡪called “modern javascript”

2015 – es6 was launched (even more important and new features were added)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**JavaScript**

* JavaScript is a scripting language (interpretation and execution occurs at the run time)
* is also called interpreted programming language (as it uses interpreter for code translations)
* JavaScript is case sensitive
* using JavaScript one can create interactive webpages/dynamic web pages
* mainly developed for client side validation



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**NOTE**

Why JavaScript file is called as “Full stack” data file?

Can be used in client side, server side, and even data base

* For client side (browser) we use core js (vanilla js)
* In webserver side we can use NodeJS
* In database we can use mongo DB and Couch DB

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Libraries built by using JavaScript**

libraries are used to simplify a complex task i.e. js alone can perform a task in bulk code, but the same task can be performed with minimal/optimal code by using libraries.

JQuery, Load#, Bootstrap, \_js Etc.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Frameworks built using JavaScript**

JS + Enhancements – Node JS (used for developing web applications)

JS + Enhancements – React JS (used for developing web applications)

JS + Enhancements – React Native (used for developing client server applications)

JS + Enhancements – angular JS (used for developing single page web applications. Eg: gmail, google maps)

JS + Enhancements – electron JS (used for developing standalone applications)

JS + Enhancements – Tensor flow (AI, machine learning applications)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Tokens**

Smallest unit of the code is called tokens

1. Keywords
2. Identifiers
3. Literals
4. Operators
5. Separators

* Keywords
* Keywords are the predefined words with predefined meaning in it (developers has defined their meaning already)

e.g.  break, case, catch, continue, debugger, default, delete, do , else , finally , for , function , if , in , instance of , new , return , switch , this , throw , try , type of , var , void , while , and with

Rules to write the keywords:

🡪JavaScript is a case sensitive language

🡪We should use keywords in lowercase letters

🡪We cannot use keywords as identifiers

* Identifiers
* JavaScript Identifiers are names given to variables, functions, etc

Rules to write the Identifiers:

🡪 You should not use any of the JavaScript reserved keywords as a variable name. For example, break or Boolean variable names are not valid.

🡪 JavaScript variable names should not start with a numeral (0-9). They must begin with a letter or an underscore or dollar character. For example, 5demo is an invalid variable name but \_5demo is a valid one.

🡪 JavaScript variable names are case-sensitive. For example, Name and name are two different variables.

* Literals
* Values what we store inside the memory allocated.

e.g. numeric literals –7, 5, 8

string literals – “7”, “5”, “8”, “hello”

Boolean literals – true, false

null

undefined

* Operators:

Used to perform some mathematical operations e.g. (+, \_, \*, \*\*, /, %) etc.

* Separators:

Used to separate the statements in the program e.g. (, (), {}, [],) etc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**JS Engine**

* Engine is basically an Interpreter
* Each and every browser has JS engine

Eg: Chrome browser 🡪 v8

Mozilla Firefox 🡪 SpiderMonkey

IE 🡪 Chakra

Safari 🡪 JavaScriptCore

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Declaring a variable in JavaScript**

Variables are the containers that you can store the values in it

Syntax :

var(keyword) name(identifier) = value

Followed by

var a // declaration

a = 10 // Initialization

console.log(a) // Utilization

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***Datatypes**

There are 2 types of datatypes

1. Primitive datatypes
2. Non-primitive datatypes (object references)

1.Primitive datatypes/immutable:

* + - * undefined
      * [Boolean](https://developer.mozilla.org/en-US/docs/Glossary/Boolean)
      * [Number](https://developer.mozilla.org/en-US/docs/Glossary/Number)
      * [String](https://developer.mozilla.org/en-US/docs/Glossary/String)
      * null
    1. Non-primitive datatypes (object references)/mutable:
* Objects
* Functions
* Arrays
* Date
* Math

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Flow control statements**

1. Conditional Statements
2. Looping Statements