Javascript:

OO Programming language:

Not a compiled language: interpreted: but we do not need any kind Interpreted:

90% websites build JS:

Features:

1. Inbuild support
2. JS follows the C programming langue syntax: Function:
3. JS weakly typed language: Primitives and non premitives
   1. int a=100; a=”Omkar”//other languages
   2. var , let: 2015 ES6 : ECMA: TS: ES-6 2015:
   3. var a=100; a=”Nikhil” , a=true;
4. JS is OOPL
5. It case sensitive Language: let a=100; A
6. JS is supportable by various OS:
7. Can a good control over your website:

History:

1993: Mosaic first: 1994: Netscape: Marc Andresson: Glue: 1995 Brendan Eich: 1995 : Sun Micro system: : May 1995: MOCHA; renamed: Livescript: : Dec 1995: Java is much boom: trending: Javascript:

OOP: OOSL

Sun: Oracle

Where we can use the JS:

1. Client side Validation:
   1. Validation: we are validating the Data : we are checking: nilesh:XZ
   2. Authorization: Authorized: Customer: Admin: add product: update: delete : purchase
   3. Authentication : user is correct: UN PW:
2. **Dynamic Drop down**: Registration: Country: State: City: India: Maharashtra: City : Pune:
3. **Date and time**:
4. **Dialogue, alert, pop, notification, prompt**:

**Coding using JS:**

**Just like CSS: where to write the JS: 3 Places:**

1. **Inside the Body tag**
2. **Inside the head tag**
3. **.js file : external file:**

**Reusability**

**Readability**

**Disadvantages:**

1. **JS: following the JS url:**
2. **2 file connected: failure:**
3. **JS and HTML: make separate HTTP:**

Comments in JS:

//Hello All

/\*sfj slfj lkdsfjlksdjflksdj flksdjflksdj flksdjflsdjfs

Fsdf

Sd

Fsd

Fsd

Fsd

Fs

Df\*/

**Variable**:: : var a=100; var b=200

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1111000001 | 000111110 | 1111000 | 00011110001 | 110000 | 111110 |  |  |  |  |
| 700 | 200 | 300 | 400 | 500. | . | . | . | . | 1000 |
| a | b | c |  |  |  |  |  |  |  |

Variable: variable: able:

const c=300

ES-6: 2015

Var – let and const:

var or let

int a=100;//other programming:

var a=100;

a=”SimpliLearn”;

2 Types of variables:

1. Local variable: let: var
   1. Declared inside any block or function: scope is local :
2. Global Variable: window object : var
   1. Not inside any block nor inside any function

Differences between var and let

let do not allow declaration variable: it possible with var:

hoisting is possible with var keyword but not with let:

Data types in JS:

1. Primitive DT: in-build:
   1. String: var name=”Nilesh”;
   2. Number: var no=10.30
   3. Boolean: true or false: var flag=false
   4. Undefined: implicit : var a; a undefined:
   5. Null: var a=null
2. Non Primitive DT:
   1. User create: with help of Primitive: DT
3. Object
4. Array
5. RegExp

Operators in JS:

1. Unary : only one operand: ++a, --a, a++, a--
2. Binary : 2 operand : +, -, \*,/,% : a+b
3. Ternary : operator which works on more than 2 operands: (a>b)?a :b
4. Arithmetic operator: +, -, \*, /, %, ++, --
5. Comparison /Relational: ==, !=, >, <,>=, <=; ===
6. Bit wise: binary bit: 0 1: 0000 1111: byte: &, |, ^, ~,<<, >>, >>> 1 | 0= 1
7. Logical: &&, ||, !: (age>18) || (gender==’female’): you are aligible for getting married
8. Assignment: =; a**=**10; +=: a=a+10: a+=10: -=, \*=, /=, %=
9. Special Op: ?:, . , . delete, in , new, instanceOf…

If—else:

1. If statement
2. If else statement
3. If else if else

Indian Girls: boys:

If(boy=Gevernmet job{

yes

}

Goodlooking : yes: Wel suttle: yes: neuclear she: yes: ---------yes

If else:

If : 10 don’t go if else:

Switch: :

switch(expression){//1

case 1:

statements;

break;

case 2:

statements;

break;

case 3:

statements;

break;

.

.

.

.

.

.

.

Default:

Statements;

}

Looping statements in JS:

Documen

1. for
2. while
3. do while

for(initialization; termination cond; inc/decr ){

}

for(let i=0; i<100;i++)

{

document.write(“ ”+i);

}

**While:**

**let i=1;**

**while(i<=10){**

**document.write(i);**

**i++;**

**}**

do while

do{

document.write(i);

i++;

}while(i<=10);

while: while over for where we do not know the exact termination condition :

while and do while:

1. top tested loop:
2. If the condition is **false** at the beginning then statements will not be executed even once

**Do while:**

1. **Exit level testing:**
2. **Even if condition is false for the first time at least once statement is get executed:**

**Function in JS: method:**

**Self contained block of code:**

1. **Wheneven you are writing a function :**
   1. **Function declaration: not required in JS : in C Language:**
   2. **Function definition**
   3. **Function call**

**How we define function in JS:**

**function name(a,b){// function signature : (parameter list)**

**statements //body**

**}**

**name(10,20)**

**Function with Parametrs:**

**Return statement**

**JS function can be anonymous or synonymous:**

**Arrow function: fat arrow function => lambda expression**

**Object in JS:**

**Class Employee{**

**Int id;**

**String name;**

**Int salary;**

**}**

**Employee e1=new Employee();**

**e1.id=101;**

**e1.name=”Remi”;**

**e1.salary=85k;**

**3 different ways to creat an object:**

1. **Object literal:**

**var emp={id:101,name:”Remi”,salary:85000}**

1. **Creating an instance of the Object: new keyword**
2. **Using the constructor of the Object: new keyword**

**Function : arguments:**

**Array: JS: Similar kinds of elements: [101,102, 103….110], [“nilesh”,”Akash”,”Remi”,”Anjali”]**

**Weakly typed language: but even though its possible:[101,”Akash”,4000.10]**

**Single dimension and multi dimension: JS: TS: multi:**

1. **By literal :**

**var names=[“Nilesh”,”Akash”,”Remi”]**

1. **Instance of Array: new**
2. **By using an Array Constructor: new**

**Java:**

**concat(): concatenating two:**

**copywithin(target,start\_index,end\_index): a[]=[10,20,30,40]**

**indexOf()**

**pop()**

**push()**

**sort()**

**reverse()**

**String: registration: ; most String:**

1. **Literal :**

**var name=”Hello”**

1. **By using string objects:**

**var str=new String(“Hello All Welcome to SimpliLearn!!!”)**

**charAt()**

**concat()**

**indeOf()**

**lastIndexOf()**

**replace()**

**substring()toLowerCase()**

**toUpperCase()**

**toString()**

**split()**

**Date:**

1. **Date()**
2. **Date(miliseconds)**
3. **Date(string)**
4. **Date(year,Month,day,hours,minutes,seconds, miliseconds)**

**getDate()**

**getDay(): 0-6 What if I want print Sunday:**

**getFullYears()**

**getHours()**

**BOM: Browser:**

**window: this is not JS object: Browser: Default object of the browser:**

**window.alert();**

**confirm()**

**open():**

**close()**

**setTimeout()**

**document**

**history: back(), forward(), go()**

**screen**

**navigator: appName: appVersion; appCodeName**

**location**

**DOM: document parent object of all the objects that are coming in DOM: document:**

**Html document:**

**document:**

**form:**

**textbox**

**radio**

**checkbox**

**getElementById()**

**getElementsByName()**

**getElementsByTagName()**

**getElementsByClassName()**

**innerHTML:**

**JS: Concept; Class: inheritance: Multi-dimension: , Function, declarations variable: Strictly typed:**

**Angular: TS:**

**Typescript:**

**Has no build in support through the browser: Installation :**

1. **NodeJs: NPM:**
2. **TSC:**

**JS: DT: Weakly typed:**

**Not new programming language: but superset of the JS:**

**TS: strongly typed language: static typed language: var a:number=100;**

**TS: Compiled language: tsc: set of tools: TS : superset JS: it’s a ES 6:**

**Inbuild:**

**Abc.ts compile: abc.js:**

1. **Strongly typed and statically:**
2. **TS: OOSC: class, inheritance: generic**
3. **Faster: : easy learn: simplest:**
4. **Compiler: Vs: JS:**
5. **Inheritance: reuasbilty of the code:**
6. **All libraries: along it has some additional features as well**

**History: TS:**

**Declaration of the variable:**

**Dt: Ts**

**Function : TS**

**Class concept:**

**Inheritance:**

**Angular:**