JavaScript Basic:Intro

* Language for web development
* onclick=”document.getElementById(‘id’).innerHTML=’Hello javaScript’”
* onclick=”document.getElementById(‘id’).src=’python.jpeg’”
* Object based
* Client site javascript:when you compute any task from your device using js then its client site java script
* Server site java script: when google gives search result when you search any thing using google then its server site js.

| simple.html | simple.js |
| --- | --- |
| <html lang="en">  <head>  <meta charset="UTF-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Document</title>  <script src='simple.js'></script>  </head>  <body>  <p id='id'>yo</p>  <button onclick='func()'>click me!</button>  </body>  </html> | function func(){  document.getElementById('id').innerHTML='sarthak gupta'  } |

1. Reserved Keys

* Export,let,super,class,extends,await,enum

1. Output

* document.write(5+6)
* console.log(5+6)
* windows.alert(5+6)
* document.getElementById("demo").innerHTML = 5 + 6;

1. Statements

* var st=”string”;
* var n=12;

1. Syntax

* Js has two types of values for syntaxin - fixed values and variable values
* Fixed values are literals that defines that numbers would or wouldn’t have decimal points and string should be under single or double quotes
* While variable values are variables used to store values and var is used to declare name example- var x=9;
* Identifiers are the name of the variable in the above case is x

1. Comments

* // fdfkv kdfsd
* /\*Sarthak gupta

And others\*/

1. Variables

* Global scope is those variable which can be work inside all over the code while function scope is that variable which work inside the function not outside.
* var \_lastName = "Johnson";
* var $myMoney = 5;
* A letter (A-Z or a-z)
* A dollar sign ($)
* Or an underscore (\_)

1. Let

* Block Scope

| var x = 10;  // Here x is 10  { //block scope  var x = 2;  // Here x is 2  }  // Here x is 2 |
| --- |
|  |

| var x = 10;  // Here x is 10  {  let x = 2;  // Here x is 2  }  // Here x is 10 |
| --- |
|  |

* We can not use let and var simultaneously if a let is declares inside the block scope then var can not be declared
* We can not use let twice in the same block

1. const

| const PI = 3.141592653589793;  PI = 3.14; // This will give an error  PI = PI + 10; |
| --- |
|  |

1. Operators

* Arithmetic operator- + , - , \* , / , % , \*\*, //
* Assignment Operator- =, (-,\*,/,%,//,\*\*) +=,(&,|,^,>>)<<=
* Comparison Operator- ==,===,!==,!=,>,<>=,<=.?
* Logical Operator- &&,||,!
* Bitwise operator- &, |, ^, ~, <<, >>,>>>

1. Arithmetic

| var x = (100 + 50) \* a;  var x = 5;  var y = 2;  var z=x/y; //quetient  var z=x%y; //remainder |
| --- |
|  |

1. Assignment

| var x = 10;  x /= 5;  var x = 10;  x \*= 5;  var x = 10;  x %= 5; |
| --- |
|  |

* Str="Sarthak"

Str+=" Gupta."

* str=str.length
* Ha

1. Data type

| var length = 16; // Number  var lastName = "Johnson"; // String  var x = {firstName:"John", lastName:"Doe"}; // Object  var x = 16 + 4 + "Volvo"; //20Volvo  var cars = ["Saab", "Volvo", "BMW"];//arrays  var person = {  firstName : "John",  lastName : "Doe",  age : 50,  eyeColor : "blue"  }; //objects |
| --- |
|  |

* Typeof() operator is used to give the data type of variable
* Null is nothing
* Myvar++ or my var--

1. Bitwise

* Bitwise operators always give result around the binary operations example

| ~5 //because ~0101 is 1010 and result is 8+2=10 |
| --- |
| 10 |

1. Error

| <script>  try {  adddlert(“message”);  }  catch(err) {  document.getElementById(‘id’).innerHTML=err.message;  }  </script> |
| --- |
|  |

|  |
| --- |
|  |

JavaScript Important:

1. Functions

* function myFunc(){

document.getElementById(‘id’).innerHTML=”Sathak”;

}

* function myFunction(p1, p2) {

return p1 \* p2;

}



1. Objects

| <script>  // Create an object:  var person = {  firstName: "John",  lastName: "Doe",  age: 50,  eyeColor: "blue"  };  // Display some data from the object:  document.getElementById("demo").innerHTML =  person.firstName + " is " + person.age + " years old.";  </script> |
| --- |
|  |

| <script>  // Create an object:  var person = {  firstName: "John",  lastName : "Doe",  id : 5566,  fullName :function(){  return this.firstName + " " + this.lastName;  }  };  // Display data from the object:  document.getElementById("demo").innerHTML = person.fullName();  </script> |
| --- |
|  |

1. Classes

* class c -> constructor() -> functions()

| class Car {  constructor(name, year) {  this.name = name;  this.year = year;  }  age() {  let date = new Date();  return date.getFullYear() - this.year;  }  }  let myCar = new Car("Ford", 2014);  document.getElementById("demo").innerHTML =  "My car is " + myCar.age() + " years old."; |
| --- |
|  |

* class -> constructor -> function()->class model extends car -> constructor ->show ->let myCar-new Model()

1. Events

* <button onclick="this.innerHTML = Date()">The time is?</button>

1. Conditionals

* if to specify a block of code to be executed, if a specified condition is true
* Use else to specify a block of code to be executed, if the same condition is false
* Use else if to specify a new condition to test, if the first condition is false
* Use switch to specify many alternative blocks of code to be executed

| <script>  function myFunction() {  var greeting;  var time = new Date().getHours();  if (time < 10) {  greeting = "Good morning";  } else if (time < 20) {  greeting = "Good day";  } else {  greeting = "Good evening";  }  document.getElementById("demo").innerHTML = greeting;  }  </script> |
| --- |
|  |

| <script>  var day;  switch (new Date().getDay()) {  case 0:  day = "Sunday";  break;  case 1:  day = "Monday";  break;  case 2:  day = "Tuesday";  break;  case 3:  day = "Wednesday";  break;  case 4:  day = "Thursday";  break;  case 5:  day = "Friday";  break;  case 6:  day = "Saturday";  }  document.getElementById("demo").innerHTML = "Today is " + day;  </script> |
| --- |
|  |

1. Looping

| <script>  var cars = ["BMW", "Volvo", "Saab", "Ford", "Fiat", "Audi"];  var text = "";  var i;  for (i = 0; i < cars.length; i++) {  text += cars[i] + "<br>";  }  document.getElementById("demo").innerHTML = text;  </script> |
| --- |
|  |

* for - loops through a block of code a number of times
* for/in - loops through the properties of an object
* for/of - loops through the values of an iterable object
* while - loops through a block of code while a specified condition is true
* do/while - also loops through a block of code while a specified condition is true

| // for in  <script>  var txt = "";  var numbers = [45, 4, 6, 16, 25];  var x;  for (x in numbers) {  txt += numbers[x] + "<br>";  }  document.getElementById("demo").innerHTML = txt;  </script> |
| --- |
|  |

| <script>  var txt = "";  var numbers = [45, 4, 9, 16, 25];  numbers.forEach(myFunction);  document.getElementById("demo").innerHTML = txt;  function myFunction(value, index, array) {  txt = txt + value + "<br>";  }  </script> |
| --- |
|  |

| <script>  let cars = ["BMW", "Volvo", "Mini"];  let text = "";  for (let x of cars) {  text += x + "<br>";  }  document.getElementById("demo").innerHTML = text;  </script> |
| --- |
|  |

1. Break and continue

| //break  <script>  var text = "";  var i;  for (i = 0; i < 10; i++) {  if (i === 3) { break; }  text += "The number is " + i + "<br>";  }  document.getElementById("demo").innerHTML = text;  </script>  //continue  <script>  var text = "";  var i;  for (i = 0; i < 10; i++) {  if (i === 3) { continue; }  text += "The number is " + i + "<br>";  }  document.getElementById("demo").innerHTML = text;  </script> |
| --- |
|  |

1. This keyword

| <script>  // Create an object:  var person = {  firstName: "John",  lastName : "Doe",  id : 5566,  fullName : function() {  return this.firstName + " " + this.lastName;  }  };  // Display data from the object:  document.getElementById("demo").innerHTML = person.fullName();  </script> |
| --- |
|  |

1. Strict mode
2. Arrow function

| <script>  var hello;  hello = () => {  return "Hello World!";  }  document.getElementById("demo").innerHTML = hello();  </script> |
| --- |
|  |

JavaScript Methods and datatype:

1. Strings

| <script>  var answer1 = "It's alright";  var answer2 = "He is called 'Johnny'";  var answer3 = 'He is called "Johnny"';  var strlen=answer1.length;  //length of string  var x = "We are the so-called \"Vikings\" from the north.";  //escape sequence  var x = "John"; // x is a string  var y = new String("John"); // y is an object  document.getElementById("demo").innerHTML =  typeof x + "<br>" + typeof y;  var x = "John"; // x is a string  var y = new String("John"); // y is an object  document.getElementById("demo").innerHTML = (x==y);//(x == y) is true because x and y have equal values  var x = "John"; // x is a string  var y = new String("John"); // y is an object  document.getElementById("demo").innerHTML = (x===y);//(x === y) is false because x and y have different types (string and object)  document.getElementById("demo").innerHTML =  answer1 + "<br>" + answer2 + "<br>" + answer3;  </script> |
| --- |
|  |

1. Number Methods

| <script>  var x = 123;  document.getElementById("demo").innerHTML =  x.toString() + "<br>" +  (123).toString() + "<br>" +  (100 + 23).toString();  var x = 9.656;  x.toExponential(2); // returns 9.66e+0  var x = 9.656;  x.toFixed(0); // returns 10  x.toFixed(2); // returns 9.66  x.toFixed(4); // returns 9.6560  x.toFixed(6);  var x = 9.656;  x.toPrecision(); // returns 9.656  x.toPrecision(2); // returns 9.7  x.toPrecision(4); // returns 9.656  x.toPrecision(6); // returns 9.65600  Number(true); // returns 1  Number(false); // returns 0  parseFloat("10"); // returns 10  parseFloat("10.33"); // returns 10.33  var x = Number.MAX\_VALUE;  </script> |
| --- |
|  |

1. Array

| <script>  var cars = [  "Saab",  "Volvo",  "BMW"  ];  var cars = new Array("Saab", "Volvo", "BMW");  var cars = ["Saab", "Volvo", "BMW"];  document.getElementById("demo").innerHTML = cars[0];//accessing array element  cars[0] = "Opel";//changing array element  fruits = ["Banana", "Orange", "Apple", "Mango"];  var last = fruits[fruits.length - 1];//accessing last array element  document.getElementById("demo").innerHTML = cars;  </script> |
| --- |
|  |

1. Array methods

* fruits.toString(); fruits.join(" \* ");
* fruits.pop(); // Removes the last element ("Mango") from fruits
* fruits.push("Kiwi"); // Adds a new element ("Kiwi") to fruits
* fruits.push("Kiwi"); // Adds a new element ("Kiwi") to fruits
* fruits.sort();
* fruits.reverse();
* function myFunction() {

1. Array sort

* points.sort(function(a, b){return a - b});
* fruits.sort();

|  |
| --- |
|  |

1. Array Iteration

| <script>  var numbers1 = [45, 4, 9, 16, 25];  var numbers2 = numbers1.map(myFunction);  document.getElementById("demo").innerHTML = numbers2;  function myFunction(value, index, array) {  return value \* 2;  var numbers = [45, 4, 9, 16, 25];  var over18 = numbers.filter(myFunction);  function myFunction(value, index, array) {  return value > 18;  var numbers1 = [45, 4, 9, 16, 25];  var sum = numbers1.reduce(myFunction);  function myFunction(total, value, index, array) {  return total + value;  var fruits = ["Apple", "Orange", "Apple", "Mango"];  var a = fruits.indexOf("Apple");  var a = fruits.lastIndexOf("Apple");  }  }  var numbers = [4, 9, 16, 25, 29];  var first = numbers.find(myFunction);  function myFunction(value, index, array) {  return value > 18;  }  </script> |
| --- |
|  |

1. Dates

* var d = new Date();
* new Date()
* new Date(*year, month, day, hours, minutes, seconds, milliseconds*)
* new Date(*milliseconds*)
* new Date(*date string*)

1. Date formats

* var d = new Date("January 25 2015");
* var d = new Date("2015/03/25");
* var d = new Date("2015-3-25");
* var d = new Date("2015-03-25T12:00:00-06:30");

1. Date Get methods

* var d = new Date();

document.getElementById("demo").innerHTML = d.getFullYear();

1. Date set methods

* var d = new Date();

d.setFullYear(2020);

document.getElementById("demo").innerHTML = d;

1. Math

* Math.E // returns Euler's number
* Math.PI // returns PI
* Math.SQRT2 // returns the square root of 2
* Math.SQRT1\_2 // returns the square root of 1/2
* Math.LN2 // returns the natural logarithm of 2
* Math.LN10 // returns the natural logarithm of 10
* Math.LOG2E // returns base 2 logarithm of E
* Math.LOG10E // returns base 10 logarithm of E

1. Random

* Math.random();

1. Booleans

* Boolean(10 > 9) // returns true
* var x = 0;

Boolean(x); // returns false

* var x = false;

var y = new Boolean(false);

// typeof x returns boolean

// typeof y returns object

1. Comparisons

| **Operator** | **Description** | **Comparing** | **Returns** |  |
| --- | --- | --- | --- | --- |
| == | equal to | x == 8 | false |  |
| x == 5 | true |  |
| x == "5" | true |  |
| === | equal value and equal type | x === 5 | true |  |
| x === "5" | false |  |
| != | not equal | x != 8 | true |  |
| !== | not equal value or not equal type | x !== 5 | false |  |
| x !== "5" | true |  |
| x !== 8 | true |  |
| > | greater than | x > 8 | false |  |
| < | less than | x < 8 | true |  |
| >= | greater than or equal to | x >= 8 | false |  |
| <= | less than or equal to | x <= 8 | true | [Try it »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_comparison11) |

CALENDER:

| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Document</title>  </head>  <body>  <div>  <form action="/action\_page.php">  <label for="cars">Choose year:</label>  <select name="cars" id="cars">  <option value="volvo">1990</option>  <option value="saab">1991</option>  <option value="opel">1992</option>  <option value="audi">1993</option>  <option value="volvo">1994</option>  <option value="saab">1995</option>  <option value="opel">1996</option>  <option value="audi">1997</option>  <option value="saab">1998</option>  <option value="opel">1999</option>  <option value="audi">2000</option>  <option value="volvo">2001</option>  <option value="saab">2002</option>  <option value="opel">2003</option>  <option value="audi">2004</option>  <option value="volvo">2005</option>  <option value="saab">2006</option>  <option value="opel">2007</option>  <option value="audi">2008</option>  <option value="saab">2009</option>  <option value="opel">2010</option>  <option value="audi">2011</option>  <option value="volvo">2012</option>  <option value="saab">2013</option>  <option value="opel">2014</option>  <option value="audi">2015</option>  <option value="volvo">2016</option>  <option value="saab">2017</option>  <option value="opel">2018</option>  <option value="audi">2019</option>  <option value="saab">2020</option>  <option value="opel">2021</option>  <option value="audi">2022</option>  </select>    </form>  </div>  <div>  <form action="/action\_page.php">  <label for="cars">Choose Month:</label>  <select name="cars" id="cars">  <option value="volvo">JAN</option>  <option value="saab">FEB</option>  <option value="opel">MAR</option>  <option value="audi">APR</option>  <option value="volvo">MAY</option>  <option value="saab">JUN</option>  <option value="opel">JUL</option>  <option value="audi">AUG</option>  <option value="saab">SEP</option>  <option value="opel">OCT</option>  <option value="audi">NOV</option>  <option value="volvo">DEC</option>    </select>    </form>  </div>  <div>  <form action="/action\_page.php">  <label for="cars">Choose Day:</label>  <select name="cars" id="cars">  <option value="volvo">1</option>  <option value="saab">2</option>  <option value="opel">3</option>  <option value="audi">4</option>  <option value="volvo">5</option>  <option value="saab">6</option>  <option value="opel">7</option>  <option value="audi">8</option>  <option value="saab">9</option>  <option value="opel">10</option>  <option value="audi">11</option>  <option value="volvo">12</option>  <option value="saab">13</option>  <option value="opel">14</option>  <option value="audi">15</option>  <option value="volvo">16</option>  <option value="saab">17</option>  <option value="opel">18</option>  <option value="audi">19</option>  <option value="saab">20</option>  <option value="opel">21</option>  <option value="audi">22</option>  <option value="volvo">23</option>  <option value="saab">24</option>  <option value="opel">25</option>  <option value="audi">26</option>  <option value="volvo">27</option>  <option value="saab">28</option>  <option value="opel">29</option>  <option value="audi">30</option>  <option value="saab">31</option>    </select>    </form>  </div>  <div>  <br><br>  <input type="submit" value="Submit">  </div>  </body>  </html> |
| --- |

JSON -

* Java Script Object Notation
* Sending any java script object to server by converting it to JSON text
* Java Script Object: var myObj = { name: "John",

age: 31,

city: "New York" };

* How to convert javascript object to json:: var myJSON = JSON.stringify(myObj);
* Vice versa: var myJSON = '{"name":"John",

"age":31,

"city":"New York"}';

var myObj = JSON.parse(myJSON);

* Data is in name/value pairs

Data is separated by commas

Curly braces hold objects

Square brackets hold arrays

* How to access?:myObj = { name: "John", age: 30, city: "New York" };

x = myObj.name; or x = myObj["name"];

* How to delete?:delete myObj.cars.car2;
* How to modify data?:myObj.name = "Gilbert";
* The file type for JSON files is ".json"