

**Javascript By Pankaj kumar agarwal**

**Contact : 7348820668**

**Whatsapp : 7348820668**

**Email : [pankaj.7613@gmail.com](mailto:pankaj.7613@gmail.com)**

**Blog : <http://pankajagarwal.in/>**

**Git Repository : <https://github.com/seecog>**

**Facebook : <https://www.facebook.com/Pankaj.Spybird>**

## **Small Talk about javascript**

What is javascript.

Define its history.

Define all versions of javascript

## **Variables in javascript**

Simple arithmetic operations :

## **Data types in javascript**

Examples on each data type

## **Condition in javascript**

If-else Condition

Else if condition

Conditional operator

## **Loop in javascript**

While loop

For Loop

Do-while loop

## **Array in javascript :**

What is an array ?

Creating an array?

Create array through different ways

Accessing the element of array Array are objects. Make a proof of it

Array properties and methods

a)length

b) sort() ,

Looping array elements

a)forEach()

b)for()

Adding new element to array : push

Removing element from array : pop

**Note** : Adding new element at higher index creates undefined holes in array Associative array in javascript

**Note** : Javascript does not supports the named index

Difference between array and object .

How to recognise an array?

### **Array methods :**

a)toString()

b)join()

c)shift()

d)unshift()

e)delete particular index item

f)splice() Use splice To add element Use splice To remove element

g)concat()

h)slice()

### **Array sort**

a)sort()

b)reverse()

c)Numerical sorting in javascript

d)Numerical sorting in descending order

e)Highest array value through sorting

f)Sort object array

### **Array iteration :**

a)Array.forEach()

b)Array.map()

c)Array.filter(value,index,array)

d)Array.reduce(total, value, index, array)

e)Array.every() f)Array.some()

g)Array.indexOf()

h)Array.lastIndexOf()

i)Array.find()

j)Array.findIndex()

## **Function in javascript**

Function having arguments

Function having arguments with return type

Anonymous function. Give an example ?

Function hoisting Self calling function :

Immediately invoked function Function closures Passing function as argument

Function return function

## **Local Variables**

a) Create a variable inside a function and try to access it outside the function

b) Create a global variable outside the function. Access it

## **Difference between call and apply**

a) Define call method

b) Define apply method

c) Define bind method

d) Difference between call , apply and bind

## **Function closures**

Define closure with examples

## **Objects in javascript**

What is an object

Math is always object

Regular expression is always object

Array is always object

Date is always object

Functions are always object

Object is always object

Create object literal

Create an object through the new operator

Javascript for..in loop

Adding new property to object

Delete property in javascript object

Object Accessors *Define the get and set methods*

Object constructors :

Adding property to an object

Adding method to an object

Adding property to a constructor

Adding method to a constructor

## **Object Prototypes**

All javascript object inherit properties and methods from prototype.

Date object inherit from Date.prototype

Array object inherit from Array.prototype

Person object inherit from Person.prototype

Object.prototype is at top of inheritance chain.

Thus Date , Array , Person object inherit from Object.prototype Define with proper example

## **Events in javascript**

a)Write a js code to change the innerHTML through button click

b)Write a js code to display the current date when clicked on button

c)Write a js code to change the image on image hover [onmouseover,onmouseout]

d)Write a js code to change the image on image hover [onkeyup,onkeydown]

e)Write a js code to change the text color on hover

f)Write a js code to change the text to uppercase in textfield : onchange

## **Html DOM Event Listener**

Example for addEventListener Add multiple events on same element

## **Event Bubbling and event Capturing**

Use addEventListener(event,function,useCapture)

In bubbling innermost event is handled first : **false**

In capturing the outermost event is handled first : **true**

Use removeEventListener()

## **Javascript errors**

Throw and catch custom error

Define all the types of errors :

Explain each of them with examples

Eval Error :

Range Error

Reference error

Syntax Error

Type Error

DecodeURI Error

finally block Validate form inside try and catch block

## **Scopes in javascript**

Types of scopes in javascript

Give an example of automatic global in javascript

## **Hoisting in javascript**

What is hoisting in javascript. Give an example

## **Javascript forms :**

## **Javascript Ajax call**

Explain ajax XMLHttpRequest and properties Send request to server through :

Get method

Post method

Put method

Delete method

## **DOM manipulation in Javascript :**

What is DOM

Different DOM methods

What is DOM document object

`document.getElementById("")`

`document.getElementsByTagName("p")` `document.getElementsByClassName("")`

Changing HTML element

`.innerHTML` , `.attribute` , `.setAttribute()` , `.style` , `.property`

## **Adding & deleting DOM elements**

`document.createElement()`

`document.removeElement()`

`document.appendChild()`

`document.replaceChild()`

`document.write()`

Adding event handlers

## **DOM elements**

a)Reference element by id

b)Reference element by className

c)Reference element by TagName ()

d)querySelectorAll() :

## **DOM Html**

## **DOM Css**

**DOM animations** : Create a simple animation example to explain dom manipulation over events

## **DOM Navigation**

parentNode

childNodes

firstChild

lastChild

nextSibling

nodeName

previousSibling

## **DOM Nodes**

## **DOM Collections**

## **DOM Node Lists**

## **Cookies in Javascript**

Create cookie in javascript

Read cookie in javascript

Change cookie

Delete cookie

Small working operation in js cookie

## **Javascript Timing Events**

setTimeout

setInterval

clearTimeout

## **Javascript Window navigator**

Window navigator

Browser cookie

Application name +code

Browser engine

Browser version

Browser agent  
Browser platform + language

## **Browser History event**

### **Javascript location**

window.location.href  
window.location.hostname  
window.location.pathname  
window.location.protocol  
Window.location.assign

## **JSON operation in javascript**

JSON intro  
JSON syntax  
JSON vs XML  
JSON data types  
JSON parse + stringify methods  
JSON object + array

### **Javascript Date**

Date format  
Date get and set methods

## **Javascript Math & Random methods**

### **Budget manager**

Project Setup and Details  
Download starter project from Github  
Project Planning and Architecture  
Implementing the Module Pattern  
Setting up the First Event Listeners  
Reading Input Data  
Creating an Initialization Function  
Creating Income and Expense Function Constructors  
Adding a New Item to Our Budget Controller

Adding a New Item to the UI  
Clearing Our Input Fields  
Updating the Budget: Controller  
Updating the Budget: Budget Controller  
Updating the Budget: UI Controller  
Project Planning and Architecture: Step 2  
Event Delegation  
Setting up the Delete Event Listener Using Event Delegation  
Deleting an Item from Our Budget Controller  
Deleting an Item from the UI  
Project Planning and Architecture: Step 3  
Updating the Percentages: Controller  
Updating the Percentages: Budget Controller  
Updating the Percentages: UI Controller  
Formatting Our Budget Numbers: String Manipulation  
Displaying the Current Month and Year  
Finishing Touches: Improving the UX

## **ECMA 6 :**

### **Installing nodejs**

### **Variable in ECMA6**

Var keyword

Let and block scope

The const

### **Parameterized function in ecma6**

### **Rest parameters**

### **Lambda**

Lambda functions

Lambda expression

Lambda statement

Generator functions

### **Collection in ECMA6**

Maps

Map operations

Sets



Sets operations

For..of() loop

Weak maps

Weak sets

Iterator

Set & Iterator

Map & Iterator

## **Class**

Constructors and functions

Creating object

Static keyword

Instanceof operator

Class inheritance

single

Multiple

Multi level

Class inheritance with method overriding

Super keyword

## **Promise in ECMA6**

Understanding promise

Understanding callback

Understanding asyncCallback

## **Module**

Exporting the module

Import a module

Template Literal

**Javascript By Pankaj kumar agarwal**