

INTRODUCTION TO JS:

Notes: Moving from **static** pages to **dynamic** pages,

Dynamism: Changing contents/formatting of a page when events occur, creating new pages on the fly, constitute dynamic pages

USING JS:

For Validation

JS Events

DOM to Change HTML content

DOM to Change stylesheets

Inserting scripts in the HEAD and BODY sections

Writing output, Using Alerts and Prompts

JS logical operators

JS arrays and Objects

JS functions

Next Class: arrays, objects, strings, string methods, type conversion, validation, Regexp, writing cookies, best practices

JS Events

onclick, Onmouseover, onmouseout, onload

DOM

<button Onclick='document.getElementById("id1").innerHTML="the content can be changed"'>

(Notes: observe the single quote and double quote usage, capitalization)

Other document attributes,

document.getElementsByName(" ">,

document.getElementsByTagName(" ")

```
<button  
onclick='document.getElementById("id1").style.color="green"'>  
<button onclick='document.getElementById("id1").src="planes.jpg"'>
```

Writing output:

Window, document methods: ***window.alert()***, ***window.prompt()***,
document.write(), ***window.open()***

`document.write("<h1>anything</h1>")` //writing output by erasing
all the current content

`alert("anything")` //shows an alert box

(Notes: Scripts can be Embedded both in the Head and the Body
section with `<script>` `</script>` tags.)

- External Javascript can be included in the html file by
`<script src="extscript.js"></script>`
- All JS statements should end with a semicolon
- Variables can be declared with VAR / LET/ CONST keywords

JS Datatypes and operators

String, Number, Boolean, Array, Object

`==, ===, >=, <=, !=, ++, --`

`&&, ||, !`

JS arrays and Objects

`const branch = ["CSE", "IT", "ECE", "EEE"]; / const branch = [];`

alternately,

`const branch = new Array("CSE", "IT", "ECE", "EEE");`

array properties and methods:

`branch.length, branch.sort(), branch[0]`

`const mobile = {brand:"Samsung", model:2023}`

`mobile.brand / mobile["brand"]`

JS control statements

If/else, for, while

`if(x<5){}`

`else{}`

```
for (let i = 0; i < arr.length; i++) {  
    temp+= arr[i] + "<br>";  
}  
for (x in mobile){ } // to loop through keys of objects /properties  
of an array  
for(x of mobile) { } // loops through properties of any iterable  
object  
*****  
While (condition){ }
```

JS Functions

Syntax:

```
function myfun(a,b){  
    }  
}
```

Invoking: myfun(1,2)

Ex: onclick=myfun(1,2)

.....
Getting input from user using window.prompt() and converting string
to integer

```
var contact=window.prompt("enter your phone number",  
"0000000000");  
var contact-num=parseInt(contact); //string to integer
```

Common String methods:

```
str.length; // string length  
str.substring(5,20); //string position between the numbers is  
extracted  
str.trim(); //removes blank spaces on either sides of the string if  
any.
```

`str.replace("target", "string to be replaced");` //replace target with the string in the second argument.
`str.charAt(1);` //returns the character at the position 1
`str.indexOf("substring");` // returns the position of the substring
`str.search("substring");` //returns the position of the substring if found
`str.split(" ");` /split a string at a delimiter (space) and store the substrings in an array
`str.match(regular expression or "substring");` //

Array methods

`arr.length` //returns the length of the array
`arr1.concat(arr2);` //to concatenate two arrays into one
`arr.sort();` // sorts a string array in alphabetical order
`arr.reverse();` //sorts an array in descending order
`arr.sort(function(a, b){return a - b});` //sorts a numeric array

Random Number generation:

`Math.floor(Math.random() * 5);` //generates a value between 0 and 4

Date() object

`const current = new Date();`
`const current = new Date(2023, 5, 15, 11, 30, 10, 0);`

Regular Expressions:

Syntax: `/pattern/modifiers;`

A regular expression is a string of characters that is used to specify a pattern matching rule.

Character Classes: (character sets, ranges)

<code>[abc]</code>	A single character of: a, b or c
<code>[^abc]</code>	A character except: a, b or c
<code>[a-z]</code>	A character in the range: a-z
<code>[^a-z]</code>	A character not in the range: a-z

[0-9]	A digit in the range: 0-9
[a-zA-Z]	A character in the range: a-z or A-Z
[a-zA-Z0-9]	A character in the range: a-z, A-Z or 0-9

Quantifiers

a?	Zero or one of a
a*	Zero or more of a
a+	One or more of a
[0-9]+	One or more of 0-9
a{3}	Exactly 3 of a
a{3,}	3 or more of a
a{3,6}	Between 3 and 6 of a

Meta characters (characters with a special meaning)

.	Any single character
\s	Any whitespace character
\S	Any non-whitespace character
\d	Any digit, Same as [0-9]
\D	Any non-digit, Same as [^0-9]
\w	Any word character
\W	Any non-word character

Anchors:

\G	Start of match
^	Start of string
\$	End of string

Modifiers:

i – case insensitive

g- perform a global match

m – perform multiple matching

Some simple examples:

```
const regexp=/abc/;  
regexp.test("check for abc"); //returns true if abc is found.
```

```
const regexp=/[br]ent/; // character classes  
regexp.test("bent"); //returns true.
```

```
const regexp=/\d+/; // meta character with a quantifier  
regexp.test("887"); //returns true.
```

```
const regexp=/^b/; // starting character of a string shall be  
regexp.test("ball"); //returns true, returns false for "dog"
```

```
var str="an, ant, at, ann, att, atn";  
str.match("an|at"); //returns the substring
```

Refer to [javascript regular expressions cheatsheets](#) for more.

Refer to [javascript best practices](#) for more.

ADP LAB -JAVASCRIPT - EXERCISES
(17-05-2023)

Learn to Use JavaScript **arrays, Objects, loops, strings**

1. Display the frequency distribution of numeric data entered in a text box (assuming that the data repeats)

Ex: if input array=[10,55,3,4,6,10,7,8,11,8] and frequency

Distribution is (10,2)(55,1)(3,1),..... (8,2).....

2. Identify the duplicates from a numeric array and display the distinct numbers.

3. Get a few names (comma separated) in a text box from user and create an input array. Order the names in the alphabetical order and length order, such that the names increase in their lengths.

Ex: if input=[html, head, body, div, pre, table, form, br, hr, tr, td]

Output=[br, hr, td, tr, div, pre, body, head, html, table]

4. Create any simple object and alter the data based on user input.

5. Design a regular expression for validating a string according to the following rules:

- Length of the string can be between 10 and 15
- String is alphanumeric and shall not have any two characters in succession
- String shall not have the following characters (0, o, 1, i, l,L)
