To design static web pages use HTML and HTML5

To give good looks and feel to the page use CSS

inline, external css, internal CSS

to use external CSS use link tag

to use internal CSS use <style> tag

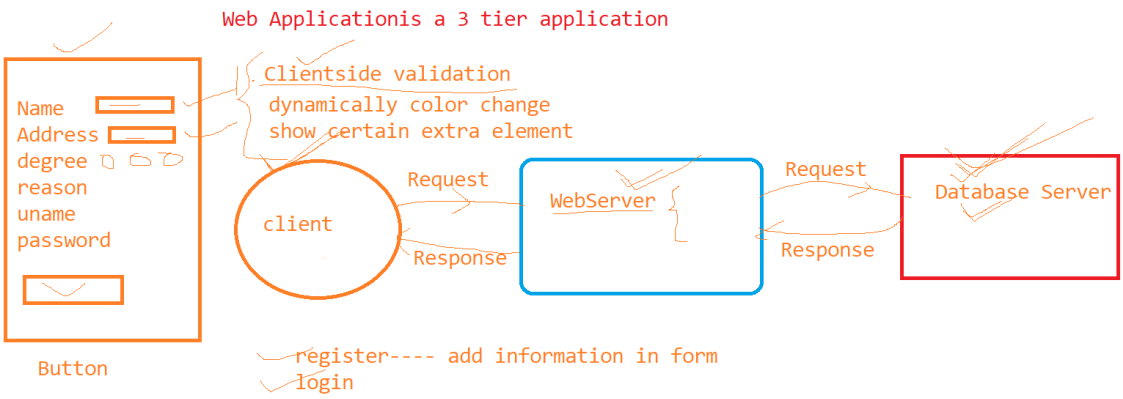
both <link> and <style> tag has to be added in <head> tag

to add inline CSS use style as an attribute

Rule

```
selector{
        property:value;
}
```

every property: value has to be separated by ;



JavaScript

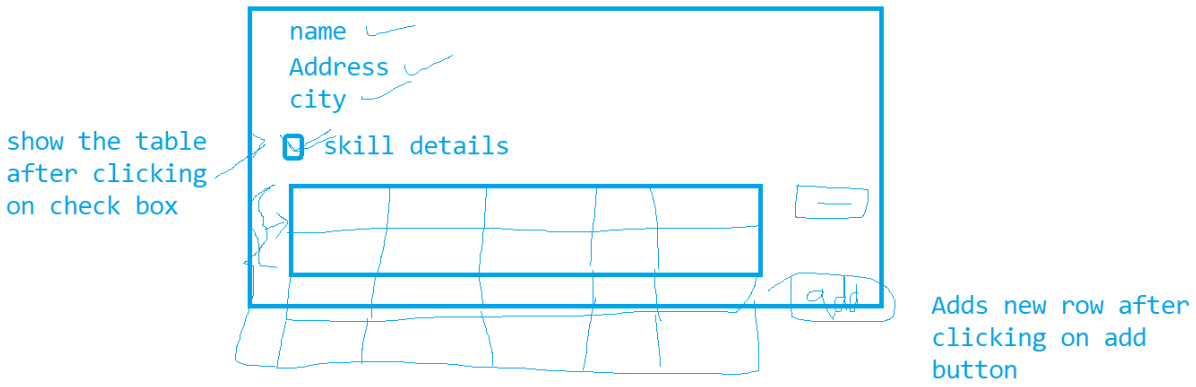
<script>

why to use Javascript

A. validation at client side
1. text box is empty
2. value of 2 textboxes is same or not
3. how check boxes have been checked
4. whether radio button is selected or not
5. email contains @ symbol
6. pattern matching

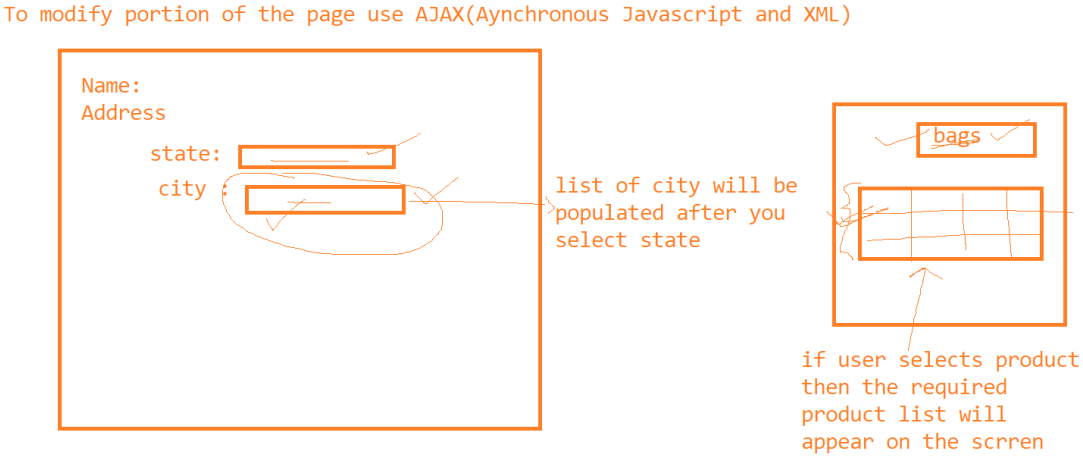
B. DOM manipulation (Document Object Model manipulation)
    1. we want to change the background color based on event
    2. display error message

3. hide or show portion of web page
4. after clicking on button add a row in the table



C. To send AJAX request
AJAX is used for populating portion of the page, to increase the speed of the website
in AJAX request Data comes in text or in JSON(Javascript object notation) format

XML –extensible markup language. it is used to store data.



How to use Javascript

JavaScript is Client side technology as well as server side technology

But we are currently learning Client side javascript

To Javascript, It is of 2 types of scripts

1. Internal Javascript
2. External javascript

Internal javascript

- it is a script written inside HTML page, and added in <script> tag
- <script> tag can be added in <head> or <body> or outside body tag

- any number of script tags can be added in HTML page
  <script> Javascript code…….</script>

External JavaScript

- If you write separate file with extension .js and add that file in HTML
  <script src=”myjavascript.js”></script>

To display messages
then use

document.write	Display message in browser window
console.log	Display message in console window
alert, prompt, confirm	Display message in popup windows, these are window object functions

alert – it is a function to show popup with ok button
confirm --- it is a function to display popup window with 2 buttons ok and cancel
prompt ---- it is a function which displays popup window with text box and button

----To execute javascript as soon as page gets load then add script tag in side Body tag and write javascript code outside function

-----But to execute javascript based on users action then it is called as event handling

and for event handling put code inside a function and call the function at appropriate place

----javascript is event driven programming

In javascript you can declare variables to store data

var sum=0; //global varible

let sum=0; //blocked scope

Rules to declare variables

1. variable should start with alphabet or _
2. it can be combination of alphabet, digits or _
3. it is case sensitive
4. no official limit on length of variable name, but it must fit in one line
5. cannot include space or any other punctuation characters

For variables data type will be managed internally by javascript. It is dynamically typed language

----if in the program the data type is decided after assigning the value at run time then it is called as dynamically typed

var a=10; // number

a=”Kishori” ///string

a={id:123,name:”kishori,”desg:”manager”,sal:2345} //object

a=12/12/2000 ///date

a=[12,13,14,”aaa,”bbb”] //array

Basic datatype in javascript

- Number (4,4.345)
- String (“this is string”, ‘This is string’)
- Boolean (true/false)
- object ({id:123,name:”kishori,”desg:”manager”,sal:2345})
- array ([12,13,14,15] or new Array(12,13,14,15))
- null to show absence of value

If you try to use variable without assigning the value to the variable then by default value is undefined

var a=null;

Operators in javascript

- unary operator(+,-,++,--)
- Arithmetic Operator (+,-,*,/,%)
- Assignment operator(=,+=,-=,*=,/=,%=)
- comparison operator(==,!=,<,>,<=,>=,===,!==)
    - //== ---checks value and === is strict checking it checks value as well as data type
    - //!= check value and !== id strict checking it checks value as well as data type
- Boolean operator (&&, ||, !)
- Bitwise operators(&, |,^,<<,>>,>>>) /// >>> this will right shift the number and also add zeros on left side
- String operators (=, +, +=)
- ternary operator condition ? true:false

if statement

if(condition){
 	
}
	if(condition){
	
	}else{
	
	}
		switch(variable){
		case val1:
		break;
		case val2:
		break;
		.
		..
		…

		default: }

---In javascript every function has it own array with name arguments and all values which you pass to the function gets stored in arguments array

----for any event(user action) occurs then JavaScript generates the object with name event and to get extra information about event use event object.

for(initialization;condition;increment/decrement){ }	To repeat statements

```
<form>
        Number1: <input type="text" name="num1" id="num1" placeholder="enter number 1"
autofocus> <br> Number2: <input type="text" name="num2" id="num2" placeHolder="enter
number 2"><br> Result : <input type="text" name="result" id="result" placeHolder="enter
number 2"><br>         Action :
        <input type="radio" name="op" id="add" value="add"><label for="add">
addition</label><br>
        <input type="radio" name="op" id="mul" value="mul"><label for="mul">
multiply</label><br>
        <input type="radio" name="op" id="div" value="div"><label for="div">
division</label><br>
        <input type="radio" name="op" id="sub" value="sub"><label for="sub">
subtract</label><br>
        <button type="button" id="btn" name="btn" onclick="doAction()">Do action</button>
        <button type="reset" id="reset" name="btn">Do action</button>
    </form>

function doAction() {
        //retrieve text box value
        var n1 = parseInt(document.getElementById("num1").value);
        var n2 = parseInt(document.getElementById("num2").value);
        var op=document.getElementsByName("op")
        for(var i=0;i<op.length;i++){

        }
```

Textbox,checkbox,radio button,select	value
P, div, span,h1,h2,h3,h4,h5,h6,ol,li,table	innerHTML

If you can use the tag to accept data from user in the form then use property .value otherwise use .innerHTML

temple operator-
It is used for displaying formatted string
var a=12;
var b=24;
value of a is 12 and value of b is 24

“value of a is “+a+” and value of b is”+b; // using +(concat) operator

` value of a is ${a} and value of b is ${b}` //using temple operator

Built in functions
Number function

isNaN	It returns false if value contains only digits otherwise it returns true if value contains alphabets or alphanumeric data	var n=123 isNan(n) --→ false var n=abc or n=abc1323 isNaN(n)-→ true
parseInt()	To convert string to int	var n=”123” n=parseInt(n)
parseFloat()	To convert string to float	var n=”123.67” n=parseFloat(n)
Math.ceil	Return the next number	Math.ceil(5.76)= 6 Math.ceil(0.3)=1
Math.round	Returns rounded to the nearest integer	Math.round(0.785)=1 Math.round(-3.05)=-3)
Math.min	To find minimum of numbers	Math.min(2,3,4)=2
Math.pow()	To find power of a number	Math.pow(5,3)=125 Math.pow(4,0.5)=2

To check data type of any variable
var num=123
document.write(“type of num :”+typeof(num));
using temple operator
`type of num : ${ typeof(num)}`

topics for tomorrow’s session

difference between var and let
regexp
String functions in javascript
validations in javascript
arrays and functions in arrays and array operators
String function
objects in javascript

String Functions
If you want to perform any operation on string

Function		
charAt	Returns character at the specified position	var name=”Rajas” name.charAt(0)=R name.charAt(1)=a name.chartAt(name.length-1)=s
length	Property which gives length of string	var name=”Rajan” name.length -→ 5
indexOf	Returns the position of the string’s first occurrence	Str=”hello everybody, everyone” Str.indexOf(“every”)--→6 Str.indexOf(“how”)-→-1 Str.indexOf(“every”,7)-→17
lastIndexOf	Returns the position of the string’s last occurrence	Str=”hello everybody, everyone” Str.lastIndexOf(“every”)--→17 Str.lastIndexOf(“how”)-→-1
replace	-Replace first occurance of the old string with new string and it returns new string -It is case sensitive	Str=”How are you,where are you,are You going for trip?” Str.replace(“you”,”everyone”) -→ How are everyone,where are everyone,are You going for trip?” Str=”How are you,where are you,are You going for trip?” Replace using regexp Str.replace(/[Yy]ou/g,”everyone”) -→ How are everyone,where are everyone,are everyone going for trip?” g is flag for globally , to replace all occurrences
slice	It returns portion of the string in given indexes(end is excluded) Negative number selects	Str=”How are you,where are you,are You going for trip?” Str.slice(0,5)----→ how a Str.slice(5)----→ re you,where are you,are You going for trip?

	the end of string end is excluded If start>end then it gives empty string	
split	It breaks the string into array at given delimiter	Str=”How are you, where are you, are You going for trip?” Myarray=str.split(“,”) Myarray =[=”How are you”,” where are you”,” are You going for trip?”]
startsWith endsWith	To check whether string starts with or endswith given string	Str=”How are you, where are you, are You going for trip?” Str.startsWith(“how”)-→true Str.endsWith(“trip?”)--→true
substring	It extracts characters from given start position but end index is exclusive	Str=”How are you, where are you, are You going for trip?” Str.substring(2,9)-→ w are y Str.substring(2)-→ w are you, where are you, are You going for trip?
toUpperCase toLowerCase	Convert string into upper or lower case	Str=”hello” Str.toUpperCase() --→HELLO Str.toLowerCase()--→hello
toString	Covert data into string format	var num=123; num.toString() --→”123”
trim	Removes spaces from both sides of the string	var str=” Hello “ var data=”welcome” str+data-→ = Hello welcome str.trim()+data--→Hellowelcome

Regular expression in Javascript
/pattern /modifier
g---global
i---ignore case
m ----multiline

^---beginning of the string
$-end of the string

.----any characters

[^ AB] ---except A and B character any other character
(x|y) ---- either X or Y expression
?-----o or 1 occurrence of preceding character
+ ----- 1 or more occurances
* --------0 or more occurance
[aeiou] -----any vowel
\w ------word character -----[A-Za-z0-9_]
\W -----non word character
\d -----digit character [0-9]
\D ------non digit character [^0-9]
\s -------white space character
\S ----- non whitespace character

^n matches string startswith n
?=n matches anystring that is followed by specific string n
?!n matches any string that is not followed by a specific string

----regular expression to check password

^(?=(.*[a-z]){3,})(?=(.*[A-Z]){2,})(?=(.*[0-9]){2,})(?=(.*[!@#$%^&*()\-__+.]){1,}).{8,}$

^ start anchor
(?=(.*[a-z]){3,}) lowercase letters. {3,} indicates that you want 3 of this group
(?=(.*[A-Z]){2,}) uppercase letters. {2,} indicates that you want 2 of this group
(?=(.*[0-9]){2,}) numbers. {2,} indicates that you want 2 of this group
(?=(.*[!@#$%^&*()\-__+.]){1,}) all the special characters in the [] fields. The ones used by regex are escaped by using the \ or the character itself. {1,} is redundant, but good practice, in case you change that to more than 1 in the future. Also keeps all the groups consistent
{8,} indicates that you want 8 or more
$

Search	Searches the pattern in string and gives position where the pattern exists	Str.search(/pattern/)
Replace	Search and replace the pattern in string with new string and returns new string Will not change the original string	Str.replace(/pattern/g,new string)
match	Search the pattern in string and returns the array of strings where matches Returns null if not found	Str.match(/pattern/) Str="How are you, how is your class going on?" Mymatch=Str.match(/H.*w/gi) [How, how]
To create a object of regular expression		regexp=new RegExp("s.*?e","gi") myreg=/s.*?e/gi vat s="something is there somewhere"; regexp.test(s) will return true or false matcharray=regexp.exec(s)

test	Search the pattern in given string and returns true or false	Pattern.test(string)
		regexp=new RegExp("s.*e","gi");
		vat s="something is there somewhere";
		regexp.test(s) ///true

exec	It will find matchobjects, which gives string as well as position and also groups	Pattern.exec(string)---to get string and position both Regexp.exec(s) regexp=new RegExp("s(.*?)e","gi"); vat s="something is there somewhere"; var result=regexp.exec(s) console.log(result)

Array function

Array in javascript are objects whose size is dynamically growable
and you can store different types of values in an array

filter	It returns new array which contains elements that matches the condition	Arr=[12,13,5,3,10]
		function check(element,index,Arr){
		return element>10;
		}
		
		Arr.filter(check) ----→ 12,13
		
		Arr.filter(x=>x>10 && x%2==0);
		Arr.filter((x,index)=>{console.log(x+"-→"+index);
		return x>10 && x%2==0);};

find	It finds the first value from array which satisfies the condition	Arr=[12,13,5,3,10]
		function check(element,index,Arr){
		return element>10;
		}
		Arr.find(check) ----→ 12
		Arr.find(x=>x>10)

findIndex	It finds the position of value which matches the given condition	Arr=[12,13,5,3,10]
		function check(element,index,Arr){
		return element>10;
		}
		Arr.findIndex(check) ----→ 0
		Arr.findIndex(x=>x>10)---→0

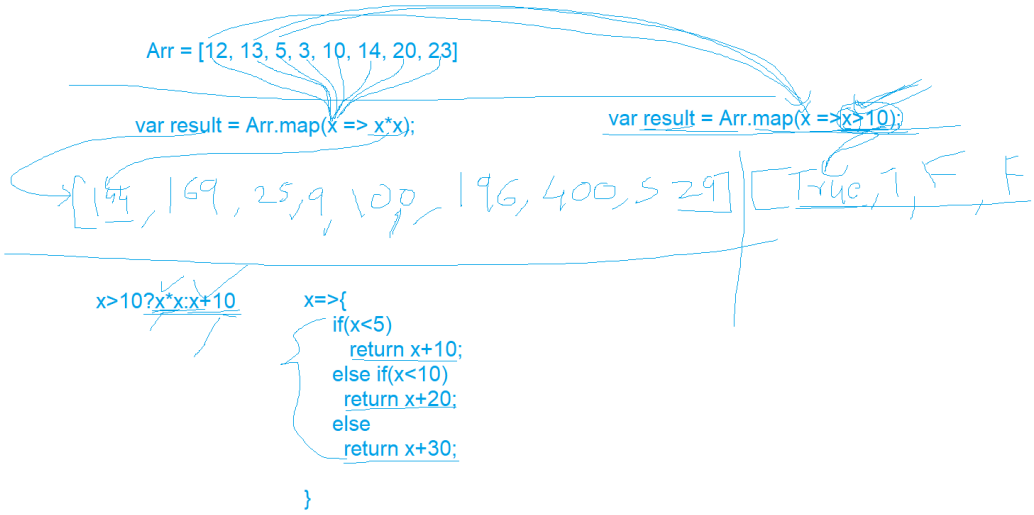
indexOf	It gives position of the given value	Arr=[12,13,5,3,10] Arr.indexOf(13)-→1
map	It applies the given function on every value	Arr=[12,13,5,3,10] var result = Arr.map(x => x*x); console.log(result)
Push	Add a new element at the end of existing array	Arr=[12,13,14] Arr.push(20) [12,13,14,20]
Pop	Remove and returns the last value of existing the array	Arr=[12,13,14] Arr.pop() [12,13]

Reverse	Returns new array with reverse of the array	Arr.reverse()
Reduce	Applies function on all values to reduce it to a single value	Arr.reduce(callback(accumulator,currentValue,currentIndex,array),initial value) Arr=[12,13,1410,20,2,3] Arr.reduce((x,y)=>x+y,0);

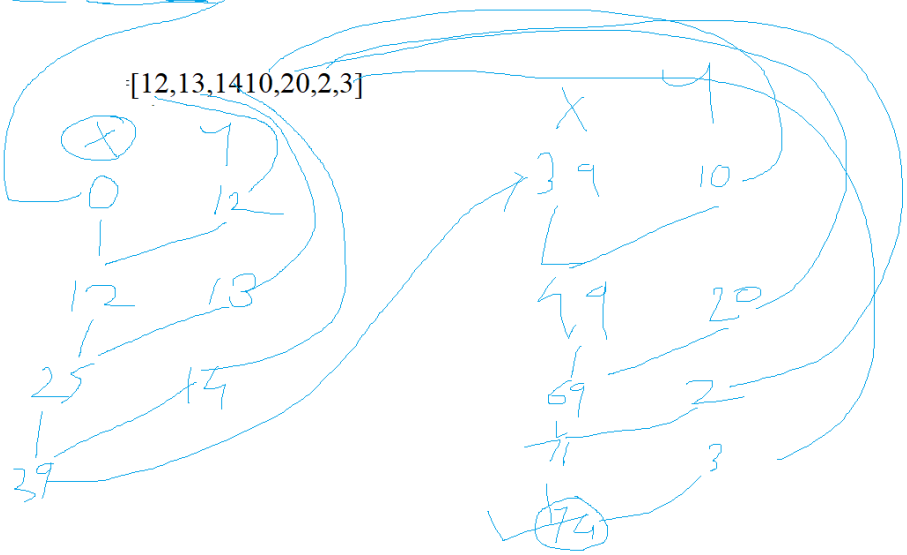
Splice	To add/remove elements to and from the existing array	Arr=[12,13,14,15,20]
		//removes 1 value from existing array
		Arr.splice(2,1) --→ [12,13,15,20]
		//remove 3 values starting from 2nd position
		Arr.splice(2,3) --→ [12,13]
		
		//remove value starting from 2 nd index remove 1 element and replace it with 10,20,30
		Arr.splice(2,1,10,20,30) [12,13,10,20,30,15,20]
		//add at 2 nd index position 10,20,30
		Arr.splice(2,0,10,20,30) [12,13,10,20,30,14,15,20]

Shift	Removes the value from the beginning of the existing array	Arr=[12,13,14] Arr.shift() [13,14]
Unshift	Adds the value at the beginning of the existing array	Arr=[12,13,14] Arr.unshift(20) [20,12,13,14]
Join	Combines all values of the array into a string separated by given delimiter	Arr[“aaa”,”bbb”,”cccc”] Arr.join(“:”); aaa:bbb:cccc

Sort	To sort the given array	Arr.sort() //will sort in ascending order ascii sort
		Arr.sort((a,b)=>a-b); /// sort in ascending order numeric sort
		Arr.sort((a,b)=>b-a); /// sort in reverse order numeric sort



Arr=[12,13,1410,20,2,3]
Arr.reduce((x,y)=>x+y,0);



Arr=[12,13,1410,20,2,3]
Arr.reduce((x,y)=>x+y,0);

```
arr.reduce((x,y)=>x*y,1)
arr.reduce((x,y)=>x+y,10)
```

Initial value