

CSS AND ITS TYPES

- * CSS stands for cascading style sheets.
- * It saves a lot of work.
- * It is a stylesheet language.
- * It can control layout for multiple webpage at once.
- * With CSS you can control font, color, size of object, space b/w elements etc.
- * Types of CSS:

There are three types of CSS:

1) In-line CSS

2) Internal CSS

3) External CSS

⇒ In-line CSS:

* ~~Inte~~ By using style attributes in HTML elements we define In-line CSS.

<!DOCTYPE>

Eg: <HTML>

 <HEAD>

 <TITLE> IN-LINE CSS <TITLE>

 <HEAD>

 <BODY>

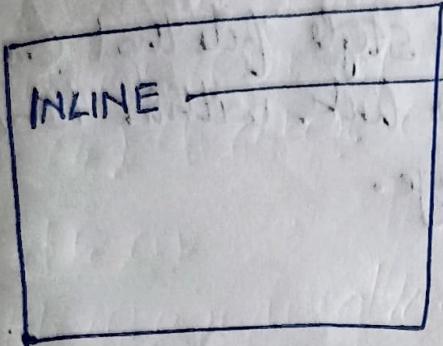
 <P style="color: green; font-size: 50px; text-align: center;">> IN-LINE

 </P>

 </BODY>

</HTML>

Outputs



Text color is

green.

⇒ Internal CSS:

* By using `<style>` element in `<head>` > `<section>` we define Internal CSS.

Eg: `<!DOCTYPE>`

`<html>`

`<head>`

`<title> Internal CSS </title>`

`<style>`

`body {`

`font-size: 20px;`

`font-color: green;`

`}`

`</style>`

`</head>`

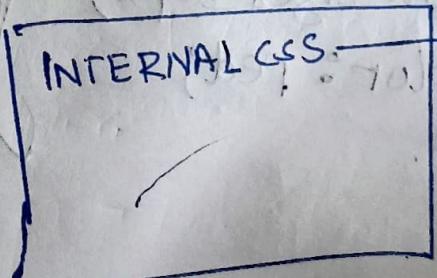
`<body>`

`<p> INTERNAL CSS </p>`

`</body>`

`</html>`

Output:



green
Color.

⇒ EXTERNAL CSS

* It is a separate CSS style file that can be accessed by creating a link within "head" section of the webpage.

Ex :- HTML file :-

```
<html>
<head>
<title> External CSS </title>
<link rel="stylesheet" href="style.css">
</head>
<body>
<h1> EXTERNAL CSS </h1>
</body>
</html>
```

CSS file :-

```
h1 {
    font-size: 30px;
    color: red;
}
```

This is

HTML file
Save this file
with extension
.html.

This is CSS file

Save this file with
extension .css.

#JAVASCRIPT CONTROL STATEMENTS

⇒ Control statements:

- * It allows structured control of sequence of applications, statements such as loops and conditional tests.
- * It allows switching input stream to another file.
- * It allows opening and closing of files.

⇒ Conditional Statement:

- * To perform different actions for different decisions, there are four types of conditional statements:

1) If

2) Else

3) Else if

4) Switch

1) If:

- * If is used to specify a block of code to be executed, if the specified condition is true.

2) else

Eg: <!DOCTYPE>

<html>

 <body>

 <head>

 <script>

 var a=20;

 if(a>10){

 document.write("value of a is greater than 10")

 }

 </script>

 </html>

Output:

Value of a is greater than 10

Else:

Else condition is used specify a block of code to be executed, if the same condition is false.

Eg:-

```
<!DOCTYPE>
```

```
<html>
```

```
<body>
```

```
<script>
```

```
Var a=20;
```

```
if(a%2==0){
```

```
document.write("a is even number");
```

```
}
```

```
else{
```

```
document.write("a is odd number");
```

```
}
```

```
</script>
```

```
<body>
```

```
<html>
```

Output:-

a is even number

3) else if

Else if condition is used to specify a block of code to be executed.

Else if condition is used to specify a new condition to test, if first condition is false.

Eg: <!DOCTYPE>
<html>

<body>

<script>

Var a=20;

if (a==10){

document.write("a is equal to 10");

y.

if (a==15){

document.write("a is equal to 15");

y.
else {

document.write("a is equal to 20");

y.

<script>

<body>

<html>-

Output:
a is equal to 20

4) Switch:

Switch is used to specify many alternative blocks of code to be executed.

Eg:

<html>

<body>

<script>.

Var grade = 'B';

Var result;

switch(grade){

Case 'A':

result = "A grade";

break;

Case 'B':

```
result = "B Grade";
break;
```

Case 'C':

```
result = "C Grade";
```

```
break;
```

default:

```
result = "No Grade";
```

}

```
document.write(result);
```

```
</script>
</body>
</html>
```

Javascript Loops:

* If we want to run ~~code~~ same code again and again , each time with different value we use ~~function~~ loops.

* There are three types of loops:

1) for loop.

2) while

3) Do While.

1) For Loop

* It loops through a block of code number of times.

Eg:- <html>

<body>

<script>

```
for(i=1; i<=5; i++) {
```

`document.write (i + "
");`

3

```
<script>  
</body>  
</html>
```

Output:

1
2
3
4
5

2) while:

i) It loops through a block of code if specified condition is true.

Eg:

```
<!DOCTYPE>  
<html>  
<body>  
<script>  
var i=11;  
while (i<=15);
```

Output:

11
12
13
14
15

```
document.write (i + "<br>");
```

i++;

}

```
<script>  
</body>  
</html>
```

3) DO WHILE

It loops through a block of code if specified condition is true.

Ex:-

<!DOCTYPE >..

<html>

<body>

<script> ..

Var i=21;

do {

document.write (i + "
");

i++;

}.

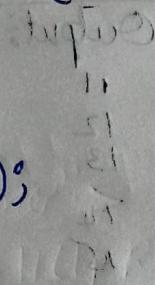
while (i <= 25);

</script>

</body>

</html> -

Output:-	
21	
22	
23	
24	
25	



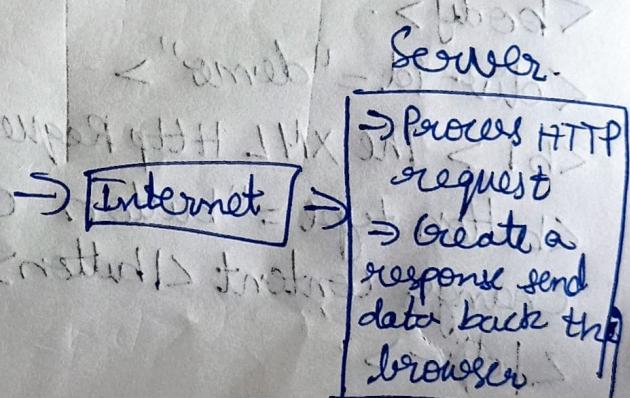
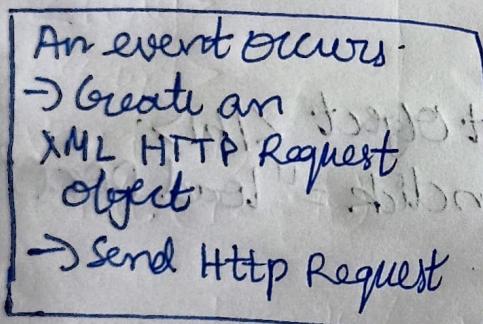
{ ("
" + i)}

i++

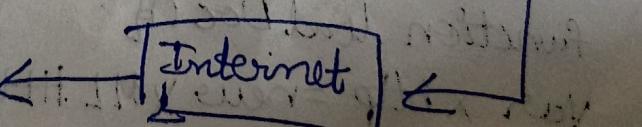
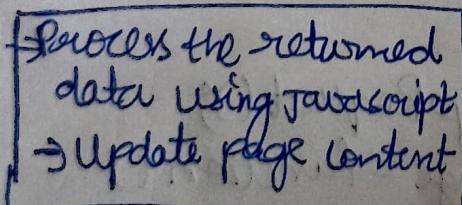
AJAX

- * AJAX = Asynchronous JavaScript and XML.
- * AJAX is not a programming language.
- * AJAX uses a combination of:
 - A browser built-in XML Http Request Object (to request data from a web server)
 - JAVA SCRIPT and HTML DOM (to display or use data)
- * AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes.
- * This means that it is possible to update parts of a webpage, without reloading the whole page.

Browser.



Browser



- 1) An event occurs in a webpage.
- 2) An XML HTTP request object is created by Javascript.
- 3) An XML HTTP request object sends a request to a web browser.
- 4) The server processes the request.
- 5) The server sends a response back to webpage.

- 6) The response is read by Javascript.
- 7) Proper action is performed by Javascript.

The XML HTTP Request Object

Syntax:

```
var xhttp = new XMLHttpRequest();
```

- All browsers supports XML HTTP Request Object.
- The XML HTTP Request Object is used to exchange data with a server behind the scenes.

Example:

```
<!DOCTYPE html>
<html>
<body>
<div id="demo">
<h1>The XML HTTP Request Object</h1>
<button type="button" onclick="loadDoc()>
Change Content</button>
</div>
</body>
```

```
function loadDoc() {
```

```
var xhttp = new XMLHttpRequest();
```

```
xhttp.onreadystatechange = function () {
```

```
if (this.readyState == 4 && this.status == 200)
```

```
document.getElementById("demo").innerHTML = this.responseText;
```

```
xhttp.open("GET", "agent-info.txt", true);  
xhttp.send();
```

3.

```
</script>
```

```
</body>
```

```
</html>.
```

Output:

The XMLHttpRequest Object

Change Content