

12.00

## HTML Recording Lecture - Live

08/10/2023

01.00

### Browser

02.00

→ How Browser renders website ?? \*\*.

03.00

Most common idea:- We write an HTML file which has tags. The Browser consumes that HTML file and converts that HTML file into DOM, and then renders the webpage.

04.00

Detailed Explanation :-

05.00

What Browsers actually do?

06.00

It is a simple software that can load some files from your computer (HDD|SSD) or it can load some files from a remote server.

- The file's encoding must be understandable to the Browser.

✉ @ ☎

Browser then figures out, how to display your content.

If I have files with different formats, the Browser will take different strategies to display the content.

| MARCH 2017 |    |    |    |    |    |    |
|------------|----|----|----|----|----|----|
| S          | S  | M  | T  | W  | F  | S  |
| 10         |    | 1  | 2  | 3  | 4  | 5  |
| 11         | 6  | 7  | 8  | 9  | 10 | 11 |
| 12         | 13 | 14 | 15 | 16 | 17 | 18 |
| 13         | 20 | 21 | 22 | 23 | 24 | 25 |
| 14         | 27 | 28 | 29 | 30 | 31 |    |

(B) How Browsers compute "How to display" any file?

SATURDAY

2017

### Browser Engine

Browsers have an engine that decides

algorithmically how to display the content.

9th Week • 056-309

FEBRUARY

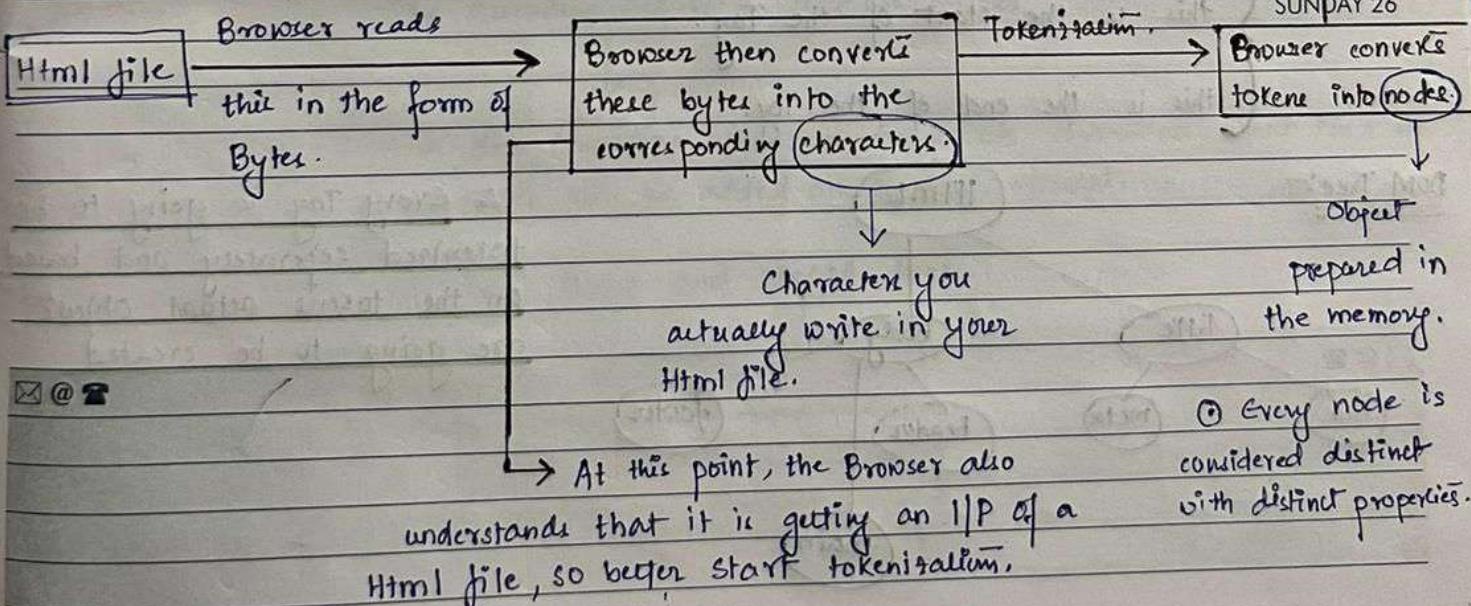
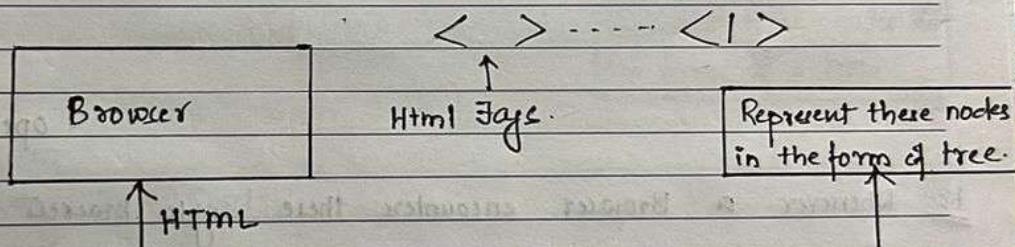
25

What Browsers can load and how to display all of these logic come to the Browser Engine.

\* Browser engine is a simple piece of software that (understands) or takes a decision of what it can display and algorithmically decide how to display that.

Examples of Browser Engines :- Gecko (Firefox); Blink (Chromium); Webkit (Safari)

(C) How Browsers load the files (HTML, CSS, JS)?



2017

MONDAY

27

10th Week • 058-307

FEBRUARY

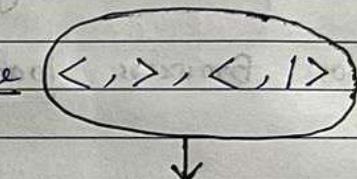
FEBRUARY 2017

| Wk | M  | T  | W  | T  | F  | S  | S  |
|----|----|----|----|----|----|----|----|
| 6  |    |    | 1  | 2  | 3  | 4  | 5  |
| 7  | 6  | 7  | 8  | 9  | 10 | 11 | 12 |
| 8  | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 9  | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 10 | 27 | 28 |    |    |    |    |    |

## Tokenization in HTML file:-

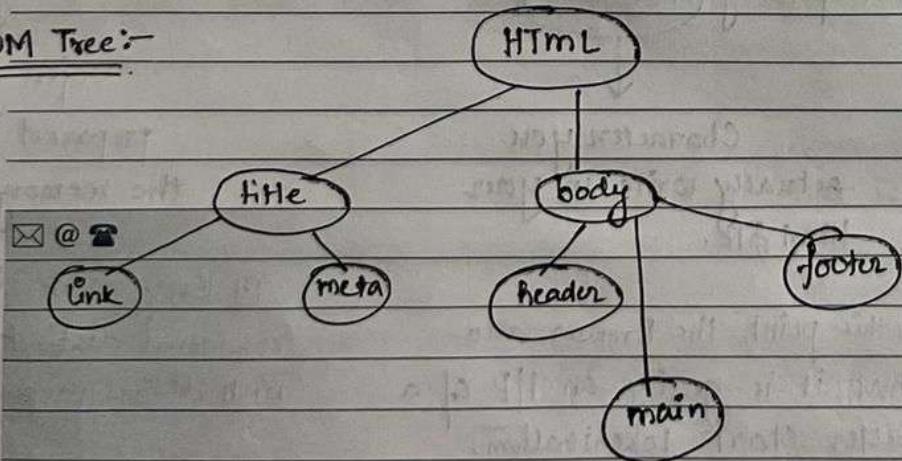
09.00

- ① <sup>1.00</sup> bunch of characters in our file, that produce the smallest meaningful piece of information.
- ② <sup>2.00</sup> The only thing that is usable for the Browser are the HTML tags.
- ③ <sup>3.00</sup> The whole file contains many such HTML Tags.
- ④ <sup>4.00</sup> ∵ The Browser tokenizes your whole file into dedicated tags. and separates everything.
- ⑤ <sup>5.00</sup> For a HTML file, the tokenizable characters are  $\langle, >, <, />$
- ⑥ <sup>6.00</sup> Whenever a Browser encounters these angle brackets it understands
  - { this is the start of the Tag.
  - } this is the end of the Tag.



Angle Brackets of these opening and closing tags.

## DOM Tree:-



∴ Every Tag is going to be tokenized separately and based on the token actual objects are going to be created

| MARCH 2017 |    |    |    |    |    |    |
|------------|----|----|----|----|----|----|
| S          | M  | T  | W  | T  | F  | S  |
| 10         | 1  | 2  | 3  | 4  | 5  |    |
| 11         | 6  | 7  | 8  | 9  | 10 | 11 |
| 12         | 13 | 14 | 15 | 16 | 17 | 18 |
| 13         | 20 | 21 | 22 | 23 | 24 | 25 |
| 14         | 27 | 28 | 29 | 30 | 31 |    |

In the DOM tree, each node represents an

Object, that object actually represents the token, the token is the Tag. The Tag is the one you

TUESDAY  
10th Week • 059-306

2017

28

FEBRUARY

prepared. Inside the tag you have a lot of attributes.

09.00

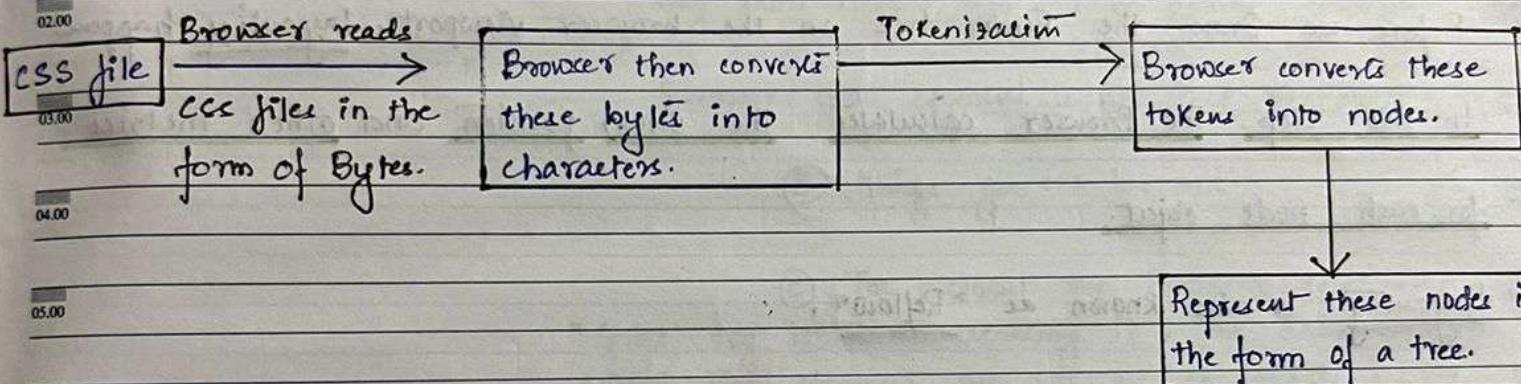
10.00 ∵ till now, the Dom tree represents the Html.

11.00 i.e. But preparing the Dom, doesn't render the Html on the webpage.

12.00 (D) CSSOM :- CSS Object Model. (also a tree data structure that is prepared by the Browser).

01.00

02.00



\* Note \*

"CSSOM"

→ "CSS" :- Cascading Style Sheet  
↳ Cascading algorithm that CSS follows.

The cascading algorithm helps CSS determine, what kind styles will be added on a HTML element.

(E) Now the Browser has DOM and CSSOM tree.

Now, what the Browser does is.

DOM + CSSOM → Render Tree.

201 / WEDNESDAY

01

10th Week • 060-305  
MARCH

This Render Tree contains info. about all

Visible DOM content → Html element

→ How it should be styled

| Wk | M  | T  | W  | T  | F  | S  |
|----|----|----|----|----|----|----|
| 10 |    |    | 1  | 2  | 3  | 4  |
| 11 | 6  | 7  | 8  | 9  | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 13 | 20 | 21 | 22 | 23 | 24 | 25 |
|    | 27 | 28 | 29 | 30 | 31 |    |

~~ex:~~ In CSS, you can mention `display: none;`

10.00 In the Browser, the Html element is hidden.

11.00 What it actually means is that, the Html element will be present in

12.00 the Dom tree, but not be applied on the Render Tree.

01.00

(B)

02.00 Before we draw the Render tree on the browser viewport, layouting happens.

03.00 In this step, the Browser calculates the size, position and other metrics

04.00 for each node object.

05.00 Layouting is also known as "Reflow".

06.00

(G)

07.00 Then Painting occurs. { i.e. Based on the information the Browser received it starts painting }.

on the Browser.

→ What happens when we have a JS file in your HTML file?

Does that creates any problem?

Having JS integrated with your HTML file is a costly affair.

✉ @ ☎

Why is it a "costly affair"?

→ We primarily use a "Script" tag to integrate our JS file into our HTML file.

APRIL 2017

| S  | M  | T  | W  | T  | F  | S  |
|----|----|----|----|----|----|----|
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 15 | 3  | 4  | 5  | 6  | 7  | 8  |
| 16 | 10 | 11 | 12 | 13 | 14 | 15 |
| 17 | 17 | 18 | 19 | 20 | 21 | 22 |
| 18 | 24 | 25 | 26 | 27 | 28 | 29 |
|    | 30 |    |    |    |    |    |

Now, when your Browser (engine) encounters a

"Script" Tag, the whole DOM construction is paused.

THURSDAY

2017

10th Week • 061-304

MARCH

02

When it is being read? → Well since we integrated our Script tag inside our HTML file.

∴ While construction of the Dom, the Browser encountered this "script" tag.

Example! —

HTML

Script

Body

Inside the script tag, if you are trying to access any DOM element, you will be getting undefined.

Because, that element is not yet prepared

(B) Things go one by one.

(C) The moment the Browser encounters "Script" tag it starts executing it. The Body and the actual HTML elements are not yet prepared. So, you won't be able to access these elements.

∴ "Costly" in terms of, it can slowdown the overall DOM loading process inside our browser.

① Script tag can also interact with our CSSOM too.

Scenario! — We encounter a "Script" tag.

Script Tag

→ DOM construction gets Halted.

→ In most engines, JS is halted until CSSOM is constructed.

2017

FRIDAY

03

10th Week • 062-303  
MARCH

## MODULE: II → Student Registration

MARCH 2017

| Wk | M  | T  | W  | T  | F  | S  | S  |
|----|----|----|----|----|----|----|----|
| 10 |    |    | 1  | 2  | 3  | 4  | 5  |
| 11 | 6  | 7  | 8  | 9  | 10 | 11 | 12 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 13 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 14 | 27 | 28 | 29 | 30 | 31 |    |    |

Requirements ① Make the whole form center aligned.

② Upon clicking on the fields, the cursor should

start pointing to the corresponding I/P field.

(B) ↴

Ex: First Name :

(max 40 characters a-z and A-Z)

If I click on this  
"Name"; cursor

should start blinking at the assigned I/P field.

③ Last Name :

(max 40 characters a-z and A-Z)

Same constraints as compared to ②.

④ Date of Birth :

⑤ Email :

⑥ Mobile Number :

(10 digit no. only)

⑦ The distance between 2 consecutive fields should be 10 units.

⑧ Gender : Male: Female: Transgender: 

⑨ Address :

⑩ Pincode :

(6 digit no. only)

⑪ Country :

⑫ Hobbies: Singing  Reading  Dancing  Others

APRIL 2017  
M T W T F S S  
1 2  
3 4 5 6 7 8 9  
10 11 12 13 14 15 16  
17 18 19 20 21 22 23  
24 25 26 27 28 29 30

(15) All the radio button / checkboxes should be wrapped as close as possible.

SATURDAY

2017

(14) Qualification:-

Please add your data:

10th Week • 063-302

MARCH

04

| SNO. | Examination | Board / University | Percentage | YOP |
|------|-------------|--------------------|------------|-----|
| 1    | Class X     |                    |            |     |
| 2    | Class XII   |                    |            |     |
| 3    | Graduation  |                    |            |     |
| 4    | Masters.    |                    |            |     |

(15)

Submit

Reset

Upon clicking of "Submit" the details must be submittable

Upon clicking of "Reset" the whole preferences should be gone.

(16) A dropdown for a list of available courses.

Approach! —

| Req. No.  | Workaround  |
|-----------|---|
| 07.00 15. | For Submit:- Since, we are looking forward to submit the user's details to the Business logic place. It makes sense to wrap the whole content inside FORM Tag and use type of "Submit". |
|           | For Reset:- Wrap the whole content inside FORM Tag, and use type of "Reset".  |

1. To make the whole content "center-aligned" we can make use of TABLE. "Table" has a property of align (values it can take center, right).

Note\*

tr → denotes row in a table

td → column values to be inserted at each row.

06

11th Week • 065-300  
MARCH

|    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|
| 11 | 6  | 7  | 8  | 9  | 10 | 11 | 12 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 13 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 14 | 27 | 28 | 29 | 30 | 31 |    |    |

Reg. No.

Workaround.

2.

`<td> <label for="firstName"> First Name </label> </td>`  
`<td> <input type="text" name="firstname" id="firstName" > </td>`

(2A) Since, we put `(id)` attribute inside `(input)` field. This `(id)` is getting mapped with `(for)` value in the `(label)` field.

Note:- For (2A) to be success `(id)` value and `(for)` value should be same.

(2B) `<td> <p> max 40 characters a-z and A-Z </p> </td>`.

3.

Same as Rep. No. (2).

4.

`<td> <label for="dob"> Date of Birth : </label> </td>`

`<td> <input type="date" name="dob" id="dob" > </td>`

5.

`<td> <label for="email"> Email Address </label> </td>`

`<td> <input type="email" name="email" id="email" > </td>`

6.

`<td> <label for="celno"> Phone Number </label> </td>`

`<td> <input type="number" name="celno" id="celno" max="10" > </td>`

`<td> <p> 10 digit only </p> </td>`.

7.

In the Table tag, there is an attribute known as "cellpadding"

∴ `<table align="center" cellpadding="10">`

|    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|
| 3  | 4  | 5  | 6  | 7  | 8  | 9  |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 |

TUESDAY

2017

11th Week • 066-299

MARCH

07

No.

Workaround.

AND 13

Best practices to follow:

- ① If you have multiple radio buttons, wrap all of them up in one **single `(td)`**.

The advantage we get out of this is proper alignment and spacing.

Method-01:-

```
<td> <label for="gender"> Gender </label> </td>
```

```
<td> Male: <input type="radio" name="gender" value="male">
```

```
Female: <input type="radio" name="gender" value="female">
```

```
others: <input type="radio" name="gender" value="others">
```

</td> *method 1*

In the Server

Method-02:- *method 2*

lets say you select "Female"

gender: Female

```
<td> Gender </td>
```

```
<td> <label for="male"> Male </label> <input type="radio" name="gender"
```

value = "Male" id = "male" >

```
<label for="female"> Female </label> <input type="radio" name="gender"
```

value = "female" id = "female" > </td>

Pursuing Method-02, will let you focus on the corresponding radio

option if even you clicked on the text "Male / Female".

13

Same as 8 & 13.

```
<td> <label for="country"> Country </label> </td>
```

```
<td> <input type="text" name="country" value="India" id="country"
```

</td>

2017

WEDNESDAY

08

11th Week • 067-298

MARCH

| Wk | M  | T  | W  | T  | F  | S  | S  |
|----|----|----|----|----|----|----|----|
| 10 |    | 1  | 2  | 3  | 4  | 5  |    |
| 11 | 6  | 7  | 8  | 9  | 10 | 11 | 12 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 13 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 14 | 27 | 28 | 29 | 30 | 31 |    |    |

Ref. No.

Workaround.

09.00

14. Seems like we need to incorporate a table inside a table.

10.00

thead :- To provide headers for our respective cols.

11.00

tbody :- Body of the Table

12.00

colspan :- Merge cols. ; Rowspan :- Merge rows.

01.00

To provide the header "Please add your data"

02.00

&lt;thead&gt;

03.00

&lt;tr&gt; → colspan = "5"

04.00

&lt;th&gt; please add your data &lt;/th&gt;

05.00

&lt;/tr&gt;

06.00

&lt;/thead&gt;

&lt;tbody&gt;

07.00

Rest of the details

08.00

&lt;/tbody&gt;

09.00

16.

&lt;label for = "courseID"&gt; Choose a course &lt;/label&gt;

→ name = "course Opted For"

&lt;select id = "courseID"&gt;

→ Clicking on this word, the dropdown box gets highlighted.

(ui)

Choose a course

Select a course ✓

Server Response:- <option value = " " > Select a course </option>

courseOptedFor: Physics &lt;option value = "physics" &gt; Physics &lt;/option&gt;

name attribute inside &lt;option value = "chemistry" &gt; Chemistry &lt;/option&gt;

&lt;option value = "Comp.Sci" &gt; Computer Science &lt;/option&gt;

&lt;/select&gt;

APRIL 2017

| SUN | MON      | TUE      | WED      | THU      | FRI | SAT |
|-----|----------|----------|----------|----------|-----|-----|
| 14  |          | 1<br>2   |          |          |     |     |
| 15  | 3<br>4   | 5<br>6   | 7<br>8   | 9        |     |     |
| 16  | 10<br>11 | 12<br>13 | 14<br>15 | 16       |     |     |
| 17  | 17<br>18 | 19<br>20 | 21<br>22 | 23       |     |     |
| 18  | 24<br>25 | 26<br>27 | 28<br>29 | 29<br>30 |     |     |

THURSDAY

2017

11th Week • 068-297

MARCH

09

## MODULE-12: KICKSTARTING CSS

① [color.adobe.com/explore/](http://color.adobe.com/explore/) :- Color palette.

10.00 ② [fontpair.co/all](http://fontpair.co/all) :- Font exploration.

11.00 ③ google fonts

css Resources.

12.00 ④ ⑤ [css-tricks.com](http://css-tricks.com) :- Good CSS Documentation.

01.00 ⑥ mdn.css/docs

01.00 ⑤ [css-gradients.io](http://css-gradients.io) → Gradient picker.

02.00 ① CSS :- Cascading Style Sheet

??

03.00 CSS uses something known as "Cascade Algorithm".

04.00 In CSS, we can have multiple properties for different HTML elements that we can assign.

05.00 How, exactly those properties are applied, resolved; all of these things are calculated by a Cascading algorithm.

06.00 Cascading Algorithm has a order of precedence.

- Let's say, we are trying to style an HTML element, and there are 2-3 different code pointers which are actually styling your HTML element.

07.00 Which style needs to be applied is calculated by Cascading Algorithm.

How to apply CSS?

• Inline CSS

• Internal CSS

• External CSS.

✉ @ ☎

2017

FRIDAY

10

11th Week • 069-296

MARCH

(A) Inline CSS :-

Let's say, we have any HTML element such as  
div, h2, span etc.

All these elements have a special attribute called "Style".

Eg:-

`<h2 style="color: gold;">`

Elite Frontend Course

① You can also provide valid Hex code.

`</h2>`

② OR, use a rgb format.

Eg:-

`<h2 style="color: gold; font-size: 50px;">`

Some Random Text.

`</h2>`

(B) Internal CSS :-

Putting Style Tag inside our Head Tag.

And inside our Style Tag we can write our CSS.

UseCase, I want to increase the font-size of all the Span Tags.

`<Style>`

So, basically inside the Style Tag we

`span {`

can put the element name that we want

`font-size: 20px;`

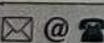
to Style → this mechanism is called as

`}`

"Selectors".

`</Style>`

Note :- Why do we call it "Internal CSS"?



Because the CSS is still present in the same HTML file only.

| APRIL 2017 |    |    |    |    |    |    |
|------------|----|----|----|----|----|----|
| S          | M  | T  | W  | T  | F  | S  |
| 1          | 2  | 3  | 4  | 5  | 6  | 7  |
| 8          | 9  | 10 | 11 | 12 | 13 | 14 |
| 15         | 16 | 17 | 18 | 19 | 20 | 21 |
| 22         | 23 | 24 | 25 | 26 | 27 | 28 |
| 29         | 30 |    |    |    |    |    |

## (C) External CSS:-

In external CSS, we write our CSS outside of our HTML file.

SATURDAY

2017

11th Week • 070-295

MARCH

Post that we have to link our HTML file with CSS file.

∴ Inside Head Tag,

```
<link rel = "stylesheet" href = "style.css">
```

The rel property actually represents that we are going to connect a style sheet.

- Represents the link where our style sheet is currently present.

Sometimes it might be a case that our style.css file can be hosted somewhere in the internet. Then, we would have to provide URL inside href.

We should never opt for inline CSS. Also, if possible avoid internal CSS as well.

The Biggest disadvantage for inline CSS, is that it makes the HTML code messy.

Advantage for inline CSS is that it is easy to implement.

For Internal CSS, similar problem of bloating happens.

SUNDAY 12

As per the "Separation of Concern" we are making the whole code messy.

It states that we should have only relevant piece of code in our file.

Now, HTML is meant to give us structure. So, implementation of styling makes it messy.

Advantage is that, we can collectively style a group of elements.

↓  
via **Selectors**

2017 MONDAY

13

12th Week • 072-293

MARCH

| W  | M  | T  | W  | T  | F  | S  | S  |
|----|----|----|----|----|----|----|----|
| 10 |    |    | 1  | 2  | 3  | 4  | 5  |
| 11 | 6  | 7  | 8  | 9  | 10 | 11 | 12 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 13 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 14 | 27 | 28 | 29 | 30 | 31 |    |    |

For external CSS, all benefits of Internal CSS are there.

Apart from that,

External CSS won't bloat our code

(And)

We can use the same style sheet in different HTML files.

② Selectors:- Selectors are mechanism that CSS provide us, using which we can collectively select a bunch of elements and style them together.

③ Universal Selector:-

If we want to apply styling to everything in our document.

(Selecting each and every element specifically)

We use "Universal Selector".

Syntax:-

\* {

background-color: # FFD393;

}

\* Note \*

① The best practice says that do not put background color inside universal selector.

② The properties we should be putting inside Universal selector are related to Box model.

③ Type Selector:-

In Type Selector, we can mention which HTML element we want to style in a particular way.

\* Note \*

What can be the possible O/P of this syntax considering the fact that the user before this styling applied a background-color in universal sel?

✉ @ ☎ Syntax:- h2 {

font-size : 12px;

background-color : white;

}

The h2 element properties will over-write whatever was applied by universal sel.

| APRIL 2017 |       |       |       |       |       |             |
|------------|-------|-------|-------|-------|-------|-------------|
| S          | M     | T     | W     | T     | F     | S           |
| 14         |       | 1 2   | 3 4   | 5 6   | 7 8   | 9           |
| 15         | 10 11 | 12 13 | 14 15 | 16 17 | 18 19 | 20 21 22 23 |
| 16         | 24 25 | 26 27 | 28 29 | 30    |       |             |

### Note \* (Best Practice) :-

IF we have to apply any property that needs to be applied on the whole doc. go for Body Tag.

TUESDAY

2017

12th Week • 073-292

MARCH

14

The best usecase of providing "Background-color"

body {

background-color: white;

}

header {

background-color: red;

}

What happens is Body along with its children will have background-color of white.

Header along with its children will have background-color of red.

Note\* Let's say we have applied "background-color" to the header tag.

And inside header tag there exists a child tag of (div). We want to

target this tag and give a certain background color of (blue).

div {

background-color: blue;

}

∴ What it signifies is that we can also target nested child and apply property on them.

Even if previously we targeted their parents and applied properties on them.

This method is also termed as "Specificity".

Scenario:-

Let's say we have a header tag and inside that we have 3 children. (The user applied a background-color property using universal sel.) I want to make background color "purple" to this header?

header {

background-color: purple;

This syntax won't apply "purple" background color to the selim (meaning the children won't be having this property rather they will have property of universal sel.)

2017

15

This is happening because universal selector (targets each and every tag).

WEDNESDAY

MARCH 2017

| Wk | M  | T  | W  | T  | F  | S  | S  |
|----|----|----|----|----|----|----|----|
| 11 | 6  | 7  | 8  | 9  | 10 | 11 | 12 |
| 12 | 13 | 20 | 21 | 22 | 23 | 24 | 25 |
| 13 | 27 | 28 | 29 | 30 | 31 |    |    |

When I am just targeting header one are failing to target children of it.

12th Week - 074-291 If we want to target the children we will have to manually target them. Hence, writing background-color inside

### (C) Class Selectors:-

universal selector is not recommended.

Class Selectors say that all the elements having same class name will be selected.

Syntax:- .class-name {

font-weight: bolder;

font-size: medium;

}

### (D) ID-selectors:-

ID Selectors are used to target one unique html element.

If we put, duplicate IDs then whatever is the first element that will be picked.

Syntax:- # id-name {

text-decoration: underline;

}

### (E) Attribute-Selector:-

If we want to select all the HTML elements which are having a particular attribute.

✉ @ ☎

usecase:- I want to select all the `INPUT` tags which are disabled. and apply background color of Beige.

| APRIL 2017 |    |    |    |    |    |    |
|------------|----|----|----|----|----|----|
| M          | T  | W  | T  | F  | S  | S  |
| 14         |    |    | 1  | 2  |    |    |
| 15         | 3  | 4  | 5  | 6  | 7  | 8  |
| 16         | 10 | 11 | 12 | 13 | 14 | 15 |
| 17         | 17 | 18 | 19 | 20 | 21 | 22 |
| 18         | 24 | 25 | 26 | 27 | 28 | 29 |
|            |    |    |    |    |    | 30 |

input[disabled] {  
background-color: beige;  
}

THURSDAY

2017

12th Week • 075 - 290

MARCH

16

09.00

### (F) Selector List:-

10.00

11.00

Selector list helps you to select a list of elements together and apply styling on them.

12.00

Eg:- I want to select all the div, h2 and span tags and apply a background color.

01.00

02.00

03.00

04.00

05.00

06.00

07.00

08.00

09.00

10.00

11.00

12.00

13.00

14.00

15.00

16.00

17.00

18.00

19.00

20.00

21.00

22.00

23.00

24.00

25.00

26.00

27.00

28.00

29.00

30.00

31.00

32.00

33.00

34.00

35.00

36.00

37.00

38.00

39.00

40.00

41.00

42.00

43.00

44.00

45.00

46.00

47.00

48.00

49.00

50.00

51.00

52.00

53.00

54.00

55.00

56.00

57.00

58.00

59.00

60.00

61.00

62.00

63.00

64.00

65.00

66.00

67.00

68.00

69.00

70.00

71.00

72.00

73.00

74.00

75.00

76.00

77.00

78.00

79.00

80.00

81.00

82.00

83.00

84.00

85.00

86.00

87.00

88.00

89.00

90.00

91.00

92.00

93.00

94.00

95.00

96.00

97.00

98.00

99.00

100.00

101.00

102.00

103.00

104.00

105.00

106.00

107.00

108.00

109.00

110.00

111.00

112.00

113.00

114.00

115.00

116.00

117.00

118.00

119.00

120.00

121.00

122.00

123.00

124.00

125.00

126.00

127.00

128.00

129.00

130.00

131.00

132.00

133.00

134.00

135.00

136.00

137.00

138.00

139.00

140.00

141.00

142.00

143.00

144.00

145.00

146.00

147.00

148.00

149.00

150.00

151.00

152.00

153.00

154.00

155.00

156.00

157.00

158.00

159.00

160.00

161.00

162.00

163.00

164.00

165.00

166.00

167.00

168.00

169.00

170.00

171.00

172.00

173.00

174.00

175.00

176.00

177.00

178.00

179.00

180.00

181.00

182.00

183.00

184.00

185.00

186.00

187.00

188.00

189.00

190.00

191.00

192.00

193.00

194.00

195.00

196.00

197.00

198.00

199.00

200.00

201.00

202.00

203.00

204.00

205.00

206.00

207.00

208.00

209.00

210.00

211.00

212.00

213.00

214.00

215.00

216.00

217.00

218.00

219.00

220.00

221.00

222.00

223.00

224.00

225.00

226.00

227.00

228.00

229.00

230.00

231.00

232.00

233.00

234.00

235.00

236.00

237.00

238.00

239.00

240.00

241.00

242.00

243.00

244.00

245.00

246.00

247.00

248.00

249.00

250.00

251.00

252.00

253.00

254.00

2017 FRIDAY

17

12th Week • 076-289

MARCH

| S  | M  | T  | W  | F  | S  | S  |
|----|----|----|----|----|----|----|
| 10 |    | 1  | 2  | 3  | 4  | 5  |
| 11 | 6  | 7  | 8  | 9  | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 13 | 20 | 21 | 22 | 23 | 24 | 25 |
| 14 | 27 | 28 | 29 | 30 | 31 |    |

## (H) child Combinator:-

09.00

Syntan:-  $A > B$

10.00

This combinator will select nodes which are a direct descendant of the first element.

11.00

• Immediate child.  
If direct descendant is not found, then mentioned properties won't be applied.

12.00

In the prev. example if we wrote something like

01.00

div > .extra {  
color: purple;  
}

Then color purple would not have been applied to class named as extra.

Because class extra is not the immediate child of parent div.

02.00

## (I) Sibling Combinator:-

03.00

Syntan:-  $A \sim B$  this basically means all (B) which are a sibling of (A).

04.00

usecase:-

<div>

target (h4) tag.

<h2> Portfolio </h2>

h2 ~ h4 {

<span> This is the portfolio </span>

font-width: bolder;

<h4> Newline </h4>

target1

</div>

<h2>

Part Two

<h2>

<p>

</p>

<h4> Newline 2 </h4>

target2

This basically means any (h4) which happens to be a sibling of h2. Target that.

Sibling of  
target1.

Sibling of  
target2.

APRIL 2017

| Wk | M  | T  | W  | T  | F  | S  | S  |
|----|----|----|----|----|----|----|----|
| 14 | 3  | 4  | 5  | 6  | 7  | 8  | 9  |
| 15 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 17 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |

SATURDAY

2017

12th Week • 077-288

MARCH

18

## (J) Adjacent sibling combinator:-

Syntax:-

A + B

The (+) combinator will capture (B) only if it is an immediate sibling of (A).

<div>

<h2> Portfolio </h2>

<span> This is the portfolio </span>

<h4> Target line </h4>

</div>

h2 + h4 {

color: red;

} cannot capture

(h4) because it is not an immediate sibling of (h2).

Remedy.

span

div + h4 {

color: red;

}

## (K) Pseudo classes and Pseudo elements:-

Pseudo classes:-

Use case:- Let's say, we want to apply some changes to an element based on some events.  
Events such as Hovering.

Scenario:- (A)

Software Engineer 2 @ Google  
Ex-software engineer @, LinkedIn.

When I am hovering over SUNDAY 19 such as Google or LinkedIn. I want the underline to go away.

Pseudo class.

a::hover

State

text-decoration: none;

Scenario:- (B)

Whenever I click on a link, it → a:visited {

color: red;

will show me (red) link there after.

2017 MONDAY

20  
MARCH

13th Week • 079-286

| Wk | M  | T  | W  | T  | F  | S  | S  |
|----|----|----|----|----|----|----|----|
| 10 |    |    | 1  | 2  | 3  | 4  | 5  |
| 11 | 6  | 7  | 8  | 9  | 10 | 11 | 12 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 13 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 14 | 27 | 28 | 29 | 30 | 31 |    |    |

## Pseudo Elements! -

09.00

### Scenario! -

10.00

<div>

<p>...</p>

11.00

<h2> Subhadip </h2>

12.00

<span> This is some random text </span>

01.00

<h4> Target line </h4>

<p>...</p>

</div>

...

Subhadip

This is some random text

Target line

...

02.00 I want to add ... before and after my content without incorporating it in the HTML doc.

03.00

div :: before {

No Spaces

content: "..." ;

}

div :: after {

No Spaces

content: "..." ;

}

06.00 :: before and :: after are not actual HTML elements, these are pseudo  
elements.

07.00 Scenario! I want to style the first child of every div.

div : first-child {

color: blue;

}

div (space) :first-child {

}

This basically implies apply styling to the first child of every div.

APRIL 2017

| S  | M  | T  | W  | T  | F  | S  |
|----|----|----|----|----|----|----|
| 14 |    | 1  | 2  |    |    |    |
| 15 | 3  | 4  | 5  | 6  | 7  | 8  |
| 16 | 10 | 11 | 12 | 13 | 14 | 15 |
| 17 | 17 | 18 | 19 | 20 | 21 | 22 |
| 18 | 24 | 25 | 26 | 27 | 28 | 29 |
|    | 30 |    |    |    |    |    |

TUESDAY

2017

13th Week • 080-285

MARCH

21

Scenario:-Software engineer 2 @ google

I want to apply hover effect while hovering over google.

Ex Software engineer @ linkedin.

&lt;ul&gt;

&lt;li&gt; Software engineer 2 @ &lt;a href = "google.com target = "\_blank"&gt;

Google&lt;/a&gt;&lt;/li&gt;

&lt;li&gt; Ex Software engineer @ &lt;a href = "linkedin.co.in" target = "\_blank"&gt;

linkedin&lt;/a&gt;&lt;/li&gt;

&lt;/ul&gt;

ul :first-child a:hover {

• NO Spaces.

text-decoration: none;

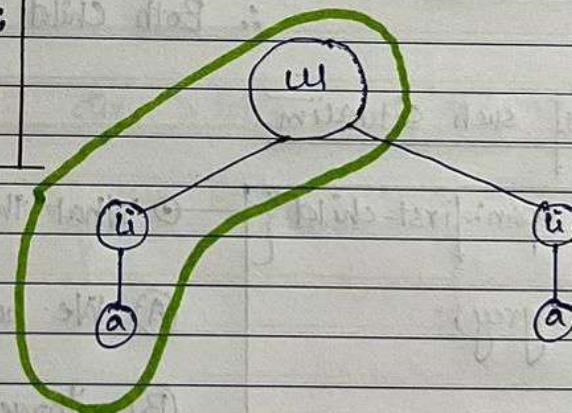
∴ Only when I hover over Google's hyperlink will I be able to not see the underline.

}

What this syntax is saying is that

We have a parent (ul).

which has multiple children.



We are interested in the first-child of ul



&lt;li&gt; Software engineer 2 @ &lt;a href = "www.google.com target = "\_blank"&gt; Google&lt;/a&gt;&lt;

Inside (which) there exists an (anchor) tag.

So, target that anchor tag and apply pseudo class :hover

2017 WEDNESDAY

22

13th Week • 081-284

MARCH

| Wk | M  | T  | W  | T  | F  | S  |
|----|----|----|----|----|----|----|
| 10 |    | 1  | 2  | 3  | 4  |    |
| 11 | 6  | 7  | 8  | 9  | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 13 | 20 | 21 | 22 | 23 | 24 | 25 |
| 14 | 27 | 28 | 29 | 30 | 31 |    |

Scenario for Child Combinator :-

09.00  
`<div>`  
10.00      `<span> Part One <span> Val One </span></span>`  
              `<span> Part Two <span> Val Two </span></span>`  
11.00      `</div>`

12. If I want to target let's say the first child?

01.00      `div > span {`  
02.00          `color: grey;`  
03.00          `}`  
04.00

What this query will do is that it will select all the immediate children of type Span of parent div.

And apply color grey to all of them.

∴ Both child will be targeted.

05.00 ∴ Remedy of such situation.

06.00      `div > span:first-child {`  
07.00          `color: grey;`  
             `}`

① What this query is saying is that

(A) We have a parent div

(B) Target the immediate children of div

having the type span and also it happens to be the first-child.

2017 FRIDAY

MODULE-13: ADVANCED CSS.

24

13th Week • 083-282

MARCH

| Wk | M  | T  | W  | T  | F  | S  | S  |
|----|----|----|----|----|----|----|----|
| 10 |    | 1  | 2  | 3  | 4  | 5  |    |
| 11 | 6  | 7  | 8  | 9  | 10 | 11 | 12 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 13 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 14 | 27 | 28 | 29 | 30 | 31 |    |    |

Box Model:-

09.00

Box Sizing:- This property helps us to decide Box Model.

10.00

① Box Model:-

11.00

lets say, we have a div tag and inside it we have some content.

12.00

<div> ② The content inside an element is rendered in the form of a Box.  
in CSS.

01.00

:

02.00 </div>

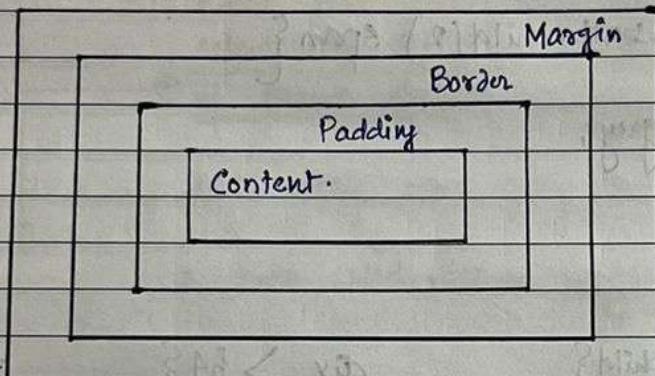
03.00

04.00

05.00

06.00

07.00



③ Padding:- Internal Spacing we give (applied in between the content and Border)

↓  
content Box.

④ Content Box:- Actual content such as text, images is displayed. The width and height of Content Box can be changed via "width" and "height".

⑤ Margin:- Space outside our box.

Eg:- Imagine there are 2 separate boxes then margin is the distance b/w them.

✉ @ ☎

APRIL 2017

| W  | M  | T  | W  | T  | F  | S  |
|----|----|----|----|----|----|----|
| 14 |    |    | 1  | 2  |    |    |
| 15 | 3  | 4  | 5  | 6  | 7  | 8  |
| 16 | 10 | 11 | 12 | 13 | 14 | 15 |
| 17 | 17 | 18 | 19 | 20 | 21 | 22 |
| 18 | 24 | 25 | 26 | 27 | 28 | 29 |
|    |    |    |    |    |    | 30 |

SATURDAY

2017

13th Week • 084-281

MARCH

25

## ② Content Box V/S Border Box!—

09.00

Let's say, we say width of 200px. How this 200px is going to get distributed.

11.00

Ⓐ In Content Box, the whole width is going to get applied on the content and padding will take "x" unit and border will take "y" unit.

01.00

Ⓑ In Border Box, including the padding and the Border we apply a total of 200px.

04.00

Now, in that 200px, we give  $\textcircled{X}$  unit for Border.

05.00

Ⓐ we give  $\textcircled{Y}$  unit for Padding.

06.00

Ⓑ We give  $[200 - (x+y)]$  unit for Content.

07.00

∴ In one way, the whole content gets the width.

In another way, border + padding + content gets the width.

SUNDAY 26

③ box-sizing : content-box;

width: 100%

Parent Container

Child Container

∴ We are using content-box.

⇒ All the 100% width will be applied

on the content only. Here border width is 0px & padding width is 0px.

Here, content width = 100% of parent.

2017 MONDAY

27

14th Week • 086-279

MARCH

| MARCH 2017 |    |    |    |    |    |    |
|------------|----|----|----|----|----|----|
| W          | M  | T  | W  | F  | S  | S  |
| 10         |    | 1  | 2  | 3  | 4  | 5  |
| 11         | 6  | 7  | 8  | 9  | 10 | 11 |
| 12         | 13 | 14 | 15 | 16 | 17 | 18 |
| 13         | 20 | 21 | 22 | 23 | 24 | 25 |
| 14         | 27 | 28 | 29 | 30 | 31 |    |

- ① `box-sizing: content-box;`  
09.00 `width: 100px; %;`  
10.00 `border: solid red 10px;`  
`padding: 5px`

Parent Container

child Container

∴ We are using content-box and a width of 100%.

① The whole width gets applied to the actual content.

② ∵ We are using border of 10px and padding of 5px.

This 10px + 5px gets added on top of 100%.

Hence, making the width of the child container wider than Parent Container.

① `box-sizing: border-box;`

07.00 `width: 100%;`

`border: solid red 12px;`

`padding: 5px;`

∴ We are using border box

Parent container

Child container

border width (12px) + padding (5px) + content width = 100% of parent.



box-sizing: border-box;

width: 100px; %;

Parent container

Child container

Border width (0px) + padding (0px) + content width = 100% of parent.

APRIL 2017

| S  | M  | T  | W  | T  | F  | S  | S  |
|----|----|----|----|----|----|----|----|
| 14 |    |    | 1  | 2  |    |    |    |
| 15 | 3  | 4  | 5  | 6  | 7  | 8  | 9  |
| 16 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 18 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |

TUESDAY

2017

14th Week • 087-278

MARCH

28

③ Nuances of resetting box-sizing via the old way:-

"Old-way" of resetting box-sizing is via the following:-

\* {

box-sizing: border-box;

}.

Now, let's consider an example:-

<div class = "component">

Now I want to apply "content-box"

<header>

for class component.

</header>

Now, if I write the following:-

</div>

.component {

box-sizing: component-box;

}

- ① The problem that will happen is that the header will still be having border-box and not component box. This is because at the start we used universal selector and mentioned border-box.

✉ @ ☎

2017 WEDNESDAY

29

14th Week • 088-277

MARCH

Remedy :- Box Sizing Inherit.

09.00

"Inherit" says that if your parent is changing how your box sizing should be then the child should inherit that same type of sizing.

10.00

html {  
    box-sizing: border-box;  
}

Apply a box-sizing of "border-box" to the whole html doc.

\*, \*:before, \*:after {

    box-sizing: inherit;

}

What this is saying is that, if in some scenario I want to change box-sizing of a parent, then its child will also inherit the same box-sizing type as that of its parent.

① Scenario:-

<div class = "new-test">

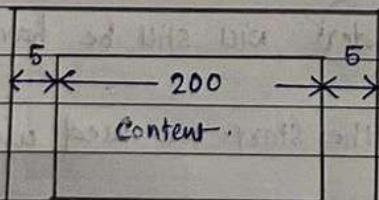
• new-test {

This is the content

</div>

    box-sizing: content-box;  
    width: 200px;  
    border: 5px solid blue;

}



Note \* If it was border-box . Then

including left border + Right Border +

Content = 200 px.

APRIL 2017

| S  | M  | T  | W  | F  | S  | S  |
|----|----|----|----|----|----|----|
| 14 | 1  | 2  |    |    |    |    |
| 15 | 3  | 4  | 5  | 6  | 7  | 8  |
| 16 | 10 | 11 | 12 | 13 | 14 | 15 |
| 17 | 17 | 18 | 19 | 20 | 21 | 22 |
| 18 | 24 | 25 | 26 | 27 | 28 | 29 |
|    | 30 |    |    |    |    |    |

THURSDAY

2017

14th Week • 089-276

MARCH

30

#### ④ Flex Box:-

09.00

What Flex Box, helps us to do is arrange elements on the whole html page.

10.00

In a Flex, there will always be a container (Parent element)

11.00

The children of parents are flex items.

01.00

<div class = "parent"> —————• Flex Container

02.00

<div> —————• Flex item  
child One

03.00

</div>

∴ Flex container will be the Html element that wraps the Flex items.

04.00

<div>  
child Two

Flex items will be the elements we want to arrange.

05.00

</div>

Child Three

When we write something like

06.00

</div>

Child Four

• parent {

border: solid red 10px;

07.00

</div>

display: flex;

</div>

When we mention the property display: flex. Then, for the container with class "parent"; flex box is enabled.

Post, that we will be happy to have command over all the flex-items.

(Defaceet direction is row-wise)

2017 FRIDAY

31

14th We  
MARCH

① Flex-wrap:- `flex-wrap: wrap;`

09.00

What happens is, every element gets ample amount of space without wrapping anything.

10.00

default value `(nowrap)`.

② Flex-direction:- `flex-direction: row-reverse;`

01.00 The flex items will be displayed in reverse manner.

02.00 If we want to apply flex-wrap and flex-direction together:-

03.00 `flex-flow`

04.00 `flex-flow: row wrap;`

→ `flex-wrap`.

05.00  
06.00  
`-flex-direction`

③ Main and Cross axis:-

07.00 Whatever is `flex-direction` → Main Axis

90° perpendicular to that → Cross Axis.

`flex-direction: row`

`flex-direction: column;`

Then main-axis → row

∴ Main-axis → column.

Cross-axis → coln

Cross axis → row.

Note\* The default nature of flex-items will be to take as big as space available of their parent.

MAY 2017

|    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|
| W  | M  | T  | W  | T  | F  | S  | S  |
| 19 | 1  | 2  | 3  | 4  | 5  | 6  | 7  |
| 20 | 8  | 9  | 10 | 11 | 12 | 13 | 14 |
| 21 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 23 | 29 | 30 | 31 |    |    |    |    |

SATURDAY

2017

14th Week • 091-274

APRIL

01

### (e) Justify - content:-

09.00

To arrange elements on the main axis.

### (f) Note

11.00

```
#parent {
```

12.00

```
border: solid red 10px;
display: flex;
height: 500px;
flex-direction: row;
```

01.00

```
}
```

C1

C2

C3.

02.00

```
# parent div {
```

03.00

```
border: solid blue 2px;
padding: 20px;
min-width: 100px;
```

04.00

```
}.
```

The children has flex-wrap: nowrap & no max-width parameter.  
Hence, they will acquire the whole width of their parent.

(1)

The children have no max-height property.  
Hence, they will take max-height of their parent.

(2)

### (g) Justify - content : start .



c1

c2

c3.

Here, in the

```
# parent div {
```

```
max-width: 200px;
max-height: 100px;
```

```
}
```

also added.

2017 MONDAY

03

15th Week • 093-272

APRIL

APRIL 2017

| Wk | M  | T  | W  | T  | F  | S  |
|----|----|----|----|----|----|----|
| 14 |    |    |    |    |    | 1  |
| 15 | 3  | 4  | 5  | 6  | 7  | 8  |
| 16 | 10 | 11 | 12 | 13 | 14 | 15 |
| 17 | 17 | 18 | 19 | 20 | 21 | 22 |
| 18 | 24 | 25 | 26 | 27 | 28 | 29 |
|    |    |    |    |    |    | 30 |

③ justify-content: end :-

|       |  |  |    |    |     |  |
|-------|--|--|----|----|-----|--|
| 10.00 |  |  | C1 | C2 | C3. |  |
| 11.00 |  |  |    |    |     |  |
| 12.00 |  |  |    |    |     |  |
| 01.00 |  |  |    |    |     |  |

④ justify-content: space-around :-

|       |   |    |   |    |   |    |
|-------|---|----|---|----|---|----|
| 03.00 |   |    |   |    |   |    |
| 04.00 | ↔ | C1 | ↔ | C2 | ↔ | C3 |
| x     |   | x  |   | x  |   | x  |
| 05.00 |   |    |   |    |   |    |
| 06.00 |   |    |   |    |   |    |
| 07.00 |   |    |   |    |   |    |

The Browser decided  
left of C1 x  
right of C3 x  
In between them also x

⑤ justify-content: space-between :-

|       |   |    |   |    |   |    |  |
|-------|---|----|---|----|---|----|--|
|       | ↔ | C1 | ↔ | C2 | ↔ | C3 |  |
| x     |   | x  |   | x  |   | x  |  |
|       |   |    |   |    |   |    |  |
|       |   |    |   |    |   |    |  |
| ✉ @ ☎ |   |    |   |    |   |    |  |

MAY 2017

| Wk | M  | T  | W  | T  | F  | S  | S  |
|----|----|----|----|----|----|----|----|
| 19 | 1  | 2  | 3  | 4  | 5  | 6  | 7  |
| 20 | 8  | 9  | 10 | 11 | 12 | 13 | 14 |
| 21 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 23 | 29 | 30 | 31 |    |    |    |    |

TUESDAY

2017

15th Week • 094-271

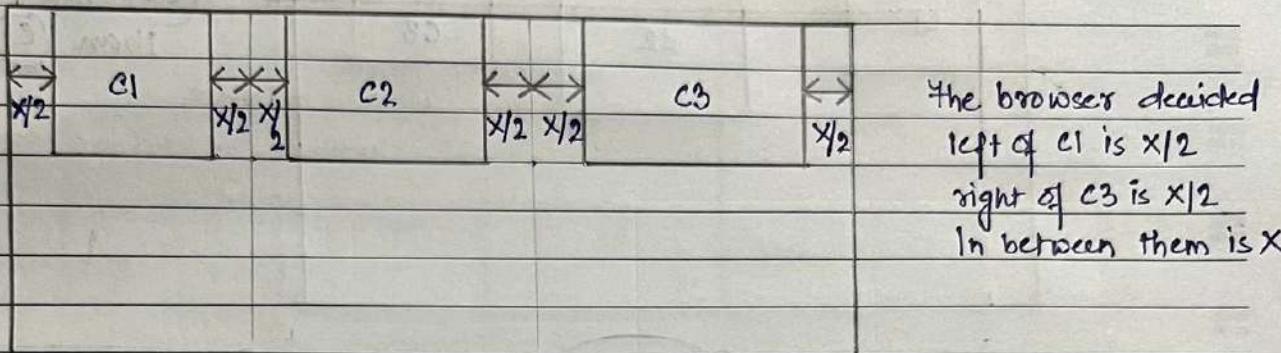
APRIL

04

09.00

(E) justify-content: Space around :-

10.00



01.00

(F) justify-content: center !

03.00

04.00

05.00

06.00

(G) Align items:-

To arrange things on the cross-axis.

# parent {

border: solid red 10px;

display: flex;

height: 500px;

flex-flow: row nowrap;

justify-content: space-around;

# parent div {

border: solid blue 2px;

padding: 20px;

min-width: 100px;

max-width: 200px;

max-height: 100px;

APRIL 2017

| Wk | M  | T  | W  | T  | F  | S  | S  |
|----|----|----|----|----|----|----|----|
| 14 |    |    |    |    |    |    | 1  |
| 15 | 3  | 4  | 5  | 6  | 7  | 8  | 2  |
| 16 | 10 | 11 | 12 | 13 | 14 | 15 | 9  |
| 17 | 17 | 18 | 19 | 20 | 21 | 22 | 16 |
| 18 | 24 | 25 | 26 | 27 | 28 | 29 | 17 |

2017 WEDNESDAY

05 APRIL

15th Week • 095-270

## ① align-items: start:-

09.00

|       |    |    |    |  |  |  |
|-------|----|----|----|--|--|--|
| 10.00 | c1 | c2 | c3 |  |  |  |
| 11.00 |    |    |    |  |  |  |
| 12.00 |    |    |    |  |  |  |
| 01.00 |    |    |    |  |  |  |
| 02.00 |    |    |    |  |  |  |

∴ We did not provide min-height in children; align-items: Start

is pushing all the child items to the top. thereby, reducing extra

height of each child. → Remedy provide min-height: 100px to children.

## ② align-items: stretch:-

05.00

|       |    |    |    |  |  |  |
|-------|----|----|----|--|--|--|
| 06.00 | c1 | c2 | c3 |  |  |  |
| 07.00 |    |    |    |  |  |  |
| 08.00 |    |    |    |  |  |  |
| 09.00 |    |    |    |  |  |  |
| 10.00 |    |    |    |  |  |  |

What stretch would do is it will start from the top and will give max-height to each and every child.

✉ @ ☎

MAY 2017

| W  | M  | T  | W  | T  | F  | S  | S  |
|----|----|----|----|----|----|----|----|
| 19 | 1  | 2  | 3  | 4  | 5  | 6  | 7  |
| 20 | 8  | 9  | 10 | 11 | 12 | 13 | 14 |
| 21 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 23 | 29 | 30 | 31 |    |    |    |    |

THURSDAY

2017

15th Week • 096-269

APRIL

06

① align-items: end :-

09.00

10.00

11.00

12.00

01.00

02.00

03.00

04.00

05.00

06.00

07.00

c1

c2

c3.

Because of the absence of min-height in children; align-items: end

is pushing all the child items to the end. thereby reducing extra height

of each child. → Remedy provide minheight: x42 to children.

② align-items: center :-

c1

c2

c3.

③ Align-content:- Helps to arrange elements on cross-axis.

Space on the cross-axis is being handled.

✉ @ Note\*

Align items → Where item should be actually present in cross axis.

Align content → Distribution of Space along cross axis.

2017 FRIDAY

07

15th Week • 097-268

APRIL

| Wk | M  | T  | W  | T  | F  | S  |
|----|----|----|----|----|----|----|
| 14 |    |    |    |    |    | 1  |
| 15 | 3  | 4  | 5  | 6  | 7  | 2  |
| 16 | 10 | 11 | 12 | 13 | 14 | 8  |
| 17 | 17 | 18 | 19 | 20 | 21 | 15 |
| 18 | 24 | 25 | 26 | 27 | 28 | 29 |
| 19 |    |    |    |    |    | 30 |

# parent {

border: solid red 1px;  
display: flex;  
height: 400px;  
flex-direction: row;  
flex-wrap: wrap;  
justify-content: start;  
align-items: start;

}

# parent div {

border: solid blue 2px;  
padding: 20px;  
min-width: 100px;  
max-height: 100px;

}.

① align-content: currentOp:-

03.00

|       |    |    |  |
|-------|----|----|--|
| 04.00 | c1 | c2 |  |
| 05.00 |    |    |  |
| 06.00 |    | c3 |  |
| 07.00 |    |    |  |

② align-content: start:-

|       |    |    |  |
|-------|----|----|--|
| ✉ @ ☎ | c1 | c2 |  |
|       |    | c3 |  |

| W  | M  | T  | W  | T  | F  | S  | S  |
|----|----|----|----|----|----|----|----|
| 19 | 1  | 2  | 3  | 4  | 5  | 6  | 7  |
| 20 | 8  | 9  | 10 | 11 | 12 | 13 | 14 |
| 21 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 23 | 29 | 30 | 31 |    |    |    |    |

`flex-flow: row wrap;`  
`justify-content: end;`  
`gap: 2rem;`

SATURDAY 2017

15th Week • 098 - 267

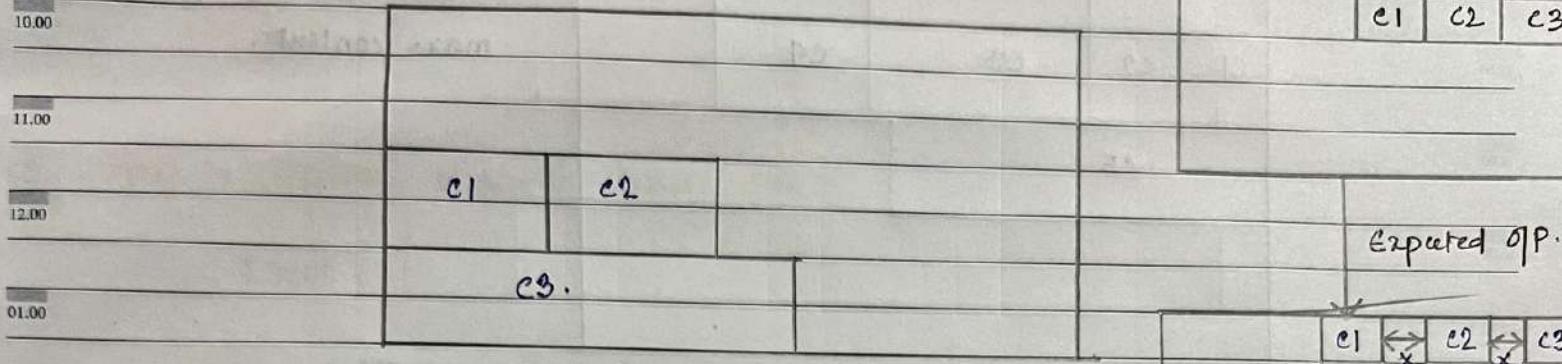
APRIL

08

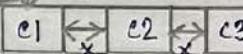
① align-content: end :

}

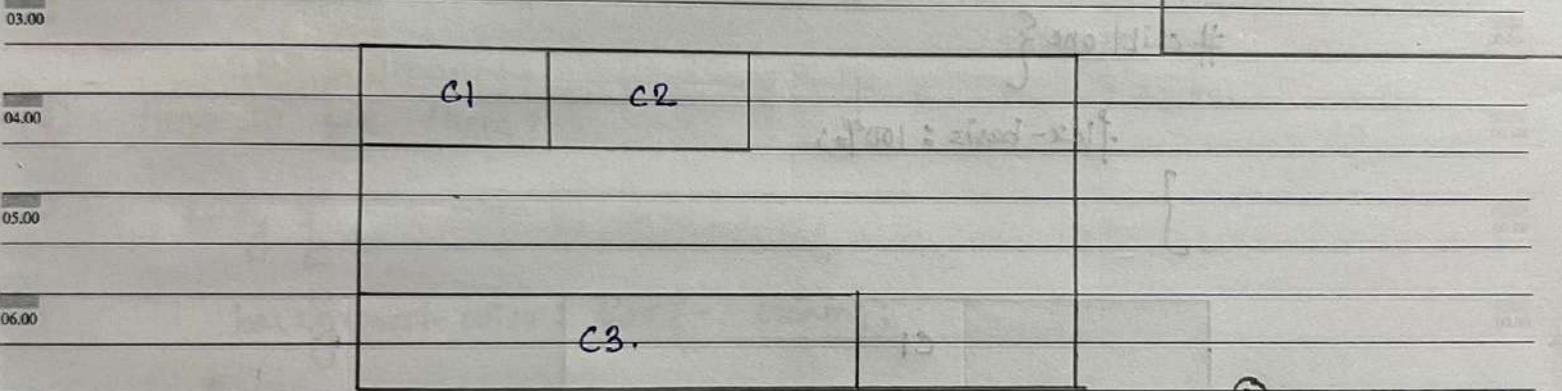
Scenario:-



Expected Output



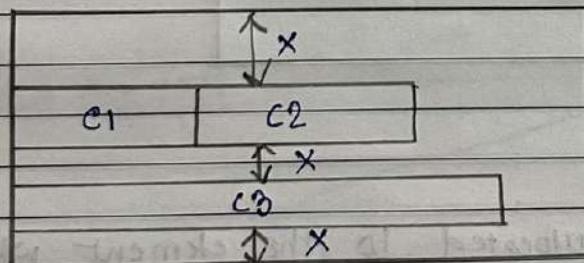
② align-content: space-between.



Main axis

SUNDAY 09

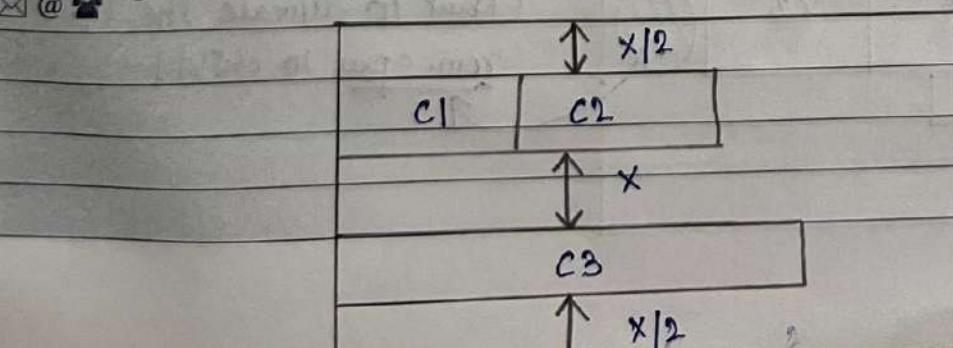
③ align-content: space-around:-



Flex-direction → direction  
 justify-content → space distri.  
 align-items → direction

align-content → space distri  
 align-items → direction

④ align-content: space-around:-



Cross Axis.

APRIL 2017

| Wk | M  | T  | W  | T  | F  | S  |
|----|----|----|----|----|----|----|
| 14 |    |    |    |    |    | 1  |
| 15 | 3  | 4  | 5  | 6  | 7  | 8  |
| 16 | 10 | 11 | 12 | 13 | 14 | 15 |
| 17 | 17 | 18 | 19 | 20 | 21 | 22 |
| 18 | 24 | 25 | 26 | 27 | 28 | 29 |

2017

MONDAY

10

16th Week • 100-265

APRIL

For a responsive design, "wrap" is preferable.

### (F) Flex-basis:-

09.00

|       |    |    |    |    |  |  |
|-------|----|----|----|----|--|--|
|       |    |    |    |    |  |  |
| 10.00 | c1 | c2 | c3 | c4 |  |  |
| 11.00 |    |    | c5 |    |  |  |
| 12.00 |    |    |    |    |  |  |
| 01.00 |    |    |    |    |  |  |

Requirement:- c1 needs to have 100% width of its parent.

03.00

# child-one {

04.00

flex-basis: 100%;

05.00

}

06.00

c1

07.00

c2 c3 c4

c5

(G)

### Flex-Grow:-

The remaining spaces will be allocated to the element which has flex-grow.

grow:

|     |    |    |    |  |  |
|-----|----|----|----|--|--|
| ✉ @ | c1 | c2 | c3 |  | I want to allocate the rem. space to child 1 |
|     | c4 |    |    |  |  |

| Wk | M  | T  | W  | T  | F  | S  | S  |
|----|----|----|----|----|----|----|----|
| 19 | 1  | 2  | 3  | 4  | 5  | 6  | 7  |
| 20 | 8  | 9  | 10 | 11 | 12 | 13 | 14 |
| 21 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 23 | 29 | 30 | 31 |    |    |    |    |

TUESDAY

2017

16th Week • 101-264

APRIL

# childOne {

09.00

flex-grow: 1;

10.00

}

11.00

⑤ How to declare variables inside CSS?

12.00

: root {

01.00

-- cream: Hexcode;

02.00

-- deepgreen: Hexcode;

03.00

}

⑥ How to use them?

body {

background-color: var(-- cream);

07.00

}

<body>  
<div>

this is a div tag.

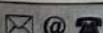
← This is a div tag

</div>  
</body>

body {

margin: 0;  
padding: 0;

Resolution



2017 WEDNESDAY

12

16th Week • 102-263

APRIL

| Wk | M  | T  | W  | T  | F  | S  | S  |
|----|----|----|----|----|----|----|----|
| 14 |    |    |    |    |    |    |    |
| 15 | 3  | 4  | 5  | 6  | 7  | 1  | 2  |
| 16 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 18 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |

### ③ Adding Margin Vertically:-

10.00

This is a H2 Tag.

11.00

This is a P tag.

12.00

Let's say, for H2 tag we added a margin of 10px → M1

02.00

" P " " " " " " 15px → M2.

03.00

Then, a resultant Margin of 15px will be applied in between 2 boxes.

04.00

∴ Resultant Margin in Vertical direction

$$\max(m_1, m_2)$$

05.00

But,

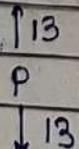
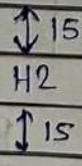
On the other hand if we apply padding of 15px of H2

06.00

and 13px of P2.

07.00

Then,



MAY 2017

| W  | M  | T  | W  | T  | F  | S  | S  |
|----|----|----|----|----|----|----|----|
| 19 | 2  | 3  | 4  | 5  | 6  | 7  |    |
| 20 | 8  | 9  | 10 | 11 | 12 | 13 | 14 |
| 21 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 23 | 29 | 30 | 31 |    |    |    |    |

THURSDAY

2017

16th Week • 103-262

APRIL

13

## ⑨ Line Height:-

Distance b/w 2 lines of text. (Best practice: 1.5)

This will scale up/down as per the zoom in/out factor.

Random pixel values increases chance of collision.

## ⑩ List Style type:-

Note\*

list-style-type : none;

The margin and padding comes from ul or ol tags and not li tags.

Removes numbering from list.

## ⑪ margin : 2px; applied all directions.

Note\*

margin : 5% auto

↓      ↳ left | right.

① To align text inside content Box

top | down

left → text-align : left.

margin : 1em auto 2em;

↓      ↓      ↳ bottom.

top      left | right

right → text-align : right

center → text-align : center

margin : 2px 1em 0 auto;

↓      ↓      ↓      ↳ left.  
top      right      bottom

2017 FRIDAY

14

16th Week • 104-261

APRIL

| W  | M  | T  | W  | F  | S  | S  |
|----|----|----|----|----|----|----|
| 14 |    |    | 1  | 2  |    |    |
| 15 | 3  | 4  | 5  | 6  | 7  | 8  |
| 16 | 10 | 11 | 12 | 13 | 14 | 9  |
| 17 | 17 | 18 | 19 | 20 | 21 | 15 |
| 18 | 24 | 25 | 26 | 27 | 28 | 16 |
|    |    |    |    |    |    | 23 |
|    |    |    |    |    |    | 29 |
|    |    |    |    |    |    | 30 |

⑫ How to align a div at the center?

09.00

<section id="hire">

<h2> Hire me </h2>

<p> I am always eager to learn new things </p>

</section>

# hire {

width: 400px;

}

Hire me

I am always eager  
to learn...

?

What auto will do is that it  
will divide rem-area equally in left and

# hire {

width: 400px;

margin: 0 auto;

right side

Hire Me

I am always eager  
to learn

right

Also, current width remains unchanged.

∴ To align div at the center → margin 0 auto

Similar concept for padding.

⑬ Padding: 0  
margin: 0 } → for UI tag.

Text will start from extreme left.

✉ @ ☎

⑭ Responsive Web Pages! - For different type of screens we are going to get different type of CSS.