



# CSS - 3

A Tradition of Excellence

IT Training | Consulting | Outsourcing

PART 1CSS Introduction:CSS (Cascading Style sheet)

→ CSS describes how HTML elements are to be displayed on screen, paper, or in other media.

HTML: what to Display → style.

CSS: How to Display ←

→ HTML की विविधता से Same या अन्य CSS द्वारा नियंत्रित की जाती है। (Art या Design) Apply की जाती है। website Design की ओर है। (CSS एवं ART हैं).

PART 2CSS Syntax:

Element      {  
 Property      :      Value  
 font-size: 12px;  
 color: blue;

PART 3How to Add CSS:

→ Three way to insert CSS

जो भी भी style लगाना होता है उस परिसर में inline नहीं, वह last वाले लगाने के लिए होता है।

External CSS

→ CSS file बनाये जो HTML 5 को link करें।

Style.css जोड़ दें।

```
  .p {  
    color: blue;  
  }
```

HTML में Head tag की Link लगाएं।

`<link rel="stylesheet"`

`type="text/css" href="style.css">`

इसका result paragraph में भी blue हो जाएगा।

Internal CSS

→ Head tag की style tag लगानी।

परंतु लाइफ लाइम।

<head>

<style>

p {  
 color: blue;  
 }

</style>

</head>

इसकी जो paragraph की

color भी blue हो जाएगी।

Inline CSS

→ HTML की विविधता

tag की attribute

style की जो भी लाइ

<p style="color: blue">

> OSCAR </p>

जो भी भी OSCAR

नो लाइ blue हो

जाएगा।

## PART 4

### CSS Selector

(HTML की वाले तो कौन से style उसी को select करता है).

#### Three type of Selection

##### Element selector

(HTML को समझता है और उसके साथ साथ style भी लेता है।)

##### Id Selector

(particular Element को style देना। Id का नाम है।)

##### Class Selector

(particular class को style देना। class का नाम है।)

उदाहरण HTML में

5 → Paragraph को उसके color (style) को, CSS में

P {

color: Red;

}

यापति यहाँ परिणाम की फॉर्म  
Red रंग आये।

(particular Element को style देना। Id का नाम है।)

change जैसा हो तो इसे

Id देना चाहिए जैसा है।

→ HTML में

<P id="p3"> OSCAR </P>

→ CSS में

# p3 {

color: green;

}

उदाहरण परिणाम-3 Green

इस तरीके

(particular Element को style देना। class का नाम है।)

→ HTML में

<h1 class="green-text">  
heading </h1>

<p class="green-text">

OSCAR </p>

→ CSS में

.green-text {

color: green;

}

इसी class का नाम है green-text यह ओरी

→ यह यह यह Element को कैसे Selector (Id, class, Element)

प्राप्त करता है जो कि पहले परिणाम की तरफ ID ने ले लिया है। Element को कैसे Selector करता है।

## PART 5

### Comments in CSS /\* Comments \*/

→ कुछ code को delete करना। Comment को कैसे करते हैं। कैसे करते हैं। कैसे करते हैं। कैसे करते हैं।

PART → G

## Colors in CSS

In CSS colors can be specified using

- Predefined color name

→ Modern browser support 140 named colors

(<https://htmlcolorcodes.com/color-names/>)

### ✓ RGB

→ rgb (red, green, blue)

→ Color value between 0 and 255

→ Black : rgb(0,0,0)

→ White : rgb(255, 255, 255)

→ Green : rgb(0, 255, 0)

### • RGBA (Red, Green, Blue, alpha)

→ alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all)

→ rgba(255, 99, 71, 0.5)

↳ opacity

### ✓ HEX

→ # RRGGBB

→ Value between 00 and ff (decimal 0-255)

→ Red : # ff0000

→ Black : # 000000

→ White : # FFFFFF

### • HSL (hue, saturation, lightness) (RED : hsl(0, 100%, 50%))

→ hue is degree on the color wheel from 0 to 360°.

0 is red, 120 is green, and 240 is blue.

→ saturation is percentage value 0%. means a shade of gray, and 100% is the full color.

→ lightness is also a percentage, 0% is black, 50%

is neither light or dark, 100% is white

### • HSLA (hue, saturation, lightness, alpha)

↳ transparent 212

→ hsla(0, 100%, 50%, 0.5).

Backgrounds

- ① Background में color लगाना है, (HTML में यह कैसे करते हैं?)  
background में color लगाना है (HTML में)

HTML में,

```
<p id="bg"> Paragraph </p>
```

CSS में,

```
# bg {
```

```
background-color: yellow;
```

```
}
```

- ② Background में image लगाना है,

```
# bg {
```

```
background-image: url('oscar/OSCAR.png');
```

```
background-repeat: repeat;
```

```
: no-repeat;
```

```
: repeat-x;
```

```
: repeat-y;
```

```
background-position: right-top;
```

```
right-bottom;
```

```
left-top;
```

```
left-bottom;
```

```
right-centre;
```

```
left-centre;
```

```
center-centre;
```

X-axis → 50px 100px ; → Y-axis  
5% 10% ;

background-size: cover; → same crop size.  
50px 100px;  
contain; → not crop full body.  
100% 100%; (full)

short  
def by  
order

background-attachment: scroll; (by default)  
fixed;

background: color, image, repeat, attachment, size;  
(order of application).

③ background colors of ~~multiple~~ Image (multiple background)  
SRMIE,

#bg {

background-color: yellow;

background-image: url('oscar/oscar.png');

background-repeat: no-repeat;

background-size: cover;

}

④ background ati ~~at~~ Image - Gollai MIE,

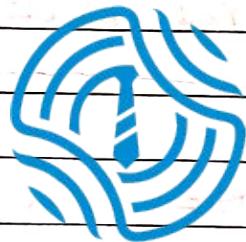
#bg {

background-image: url('oscar/1.png'),

url('oscar/2.png');

background-repeat: no-repeat;

background-position: left top ~~①~~, right bottom;



**Oscar**  
CAREER POINT

A Tradition of Excellence

PART → 8

## Border

# border {

- border-style: solid; → solid border
- : dotted; → dotted border
- : dashed; → dashed border
- : double; → double border
- : groove; → frame border
- : outset; → outward border
- : inset; → inward border
- : ridge; → groove of button
- : none; → by default or style

① : Solid dashed none dotted; →

: Solid dotted; →

. border-width: 5px; →

: 5px 10px; →

: 5px 7px 9px 10px; →

border-color: red; →

: red black; →

Short cut

border: width style color;

→ on all 4 sides of object property used simultaneously

border-top-style: solid;

border-top-width: 10px; →

border-top-color: red;

→ border at Edge & curve same side, (if different sides).

border-radius: 10px;

: 10px 20px;

: 10px 20px 30px 2px;

→ dots on Edge of border  $\rightarrow$   $10px$

border-top-left-radius: 10px  $\rightarrow$  

## PART 9

### Height and Width:

#### HTML $\rightarrow$

```
<div id="div1"> OSCAR CAREER POINT IS BEST  
TRAINING CENTER IN IT FIELD </div>
```

#### CSS $\rightarrow$

```
#div1 {
```

background-color: red;

height: 50%;  $\leftarrow$  body or 50%. height  $\rightarrow$  50px

width: 70%;

max-width: 500px;

max-height: 300px;

min-width: 100px;

min-height: 200px;

background color of 1 block  
set here  $\rightarrow$ , max width  
sets min height, width  
min max-height 200px  
width or width  $\rightarrow$  50px. or 70px.

min-width  $\rightarrow$  100px

## PART 10

### Padding $\rightarrow$ (content and border of div or gap of padding)

```
#div1 {
```

(value small + (top) or vertical).

border: 1px solid red;

padding-top: 50px;

padding-bottom: 10px;

padding-right: 10px;

padding-left: 20px;

padding: 20px 50px;  $\rightarrow$  left along 20px of padding

padding: 20px 50px;  $\rightarrow$  top-bottom  $\rightarrow$  left-right

padding: 10px 20px 25px 30px;  $\rightarrow$  clockwise padding

PART 11

Margin: (border of child of element અને  
space of margin સેટો).

(Value + મળ - દ્વારા કિશે રાખી).

(Margin collapses કરતું) જી એવી  
માયો માન કે આજી જી અલો-અલો એલેમેન્ટ  
કે margin હોય ત્રૈન સ્પષ્ટ નથી  
margin હોય નથી. (જો  $\rightarrow$  top-bottom માં થાય).

→ # mary {

(left and right જી માન ગે + થાય).

margin: auto; → centre સ્થાનની.

margin-top: -10px;

margin-bottom: 10px;

margin-left: 20px;

margin-right: 25px;

margin: 10px 25px;

margin: 10px;

margin: 10px 15px 20px 25px;

નવી ફીલ

Padding ની ઠિકાતા

ડાનો margin  
છે છે.

PART 12

(Important)

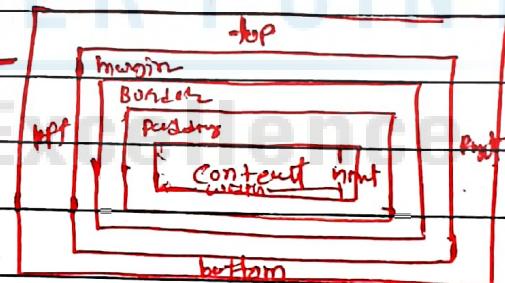
(જોયો આપણો height અને width  
Set કરીયો જીથી કે મારી content કુઝ

set થાય છે, અની પૂરી Element ઓ

set સ્થગ કરીયો padding, border,

અની margin કે (combine કરીયું).

Box Model :-



⇒ Total <sup>element</sup> width = width + left padding + right padding  
+ left border + right border  
+ left margin + right margin

⇒ Total Element height ⇒ Height + top padding + bottom padding  
+ top border + bottom border  
+ top margin + bottom margin

Contact: +91 9157929797

Email: info@osccareerpoint.com

Web: www.osccareerpoint.com

Address: 404, Shubh Square, Opp Venus Hospital, Lal Darwaja, Surat

FOLLOW US ON:



# text {

color: red;

text-align: left; → ~~left align~~: right; → ~~right align~~: center; → ~~center align~~: justify; → (All line Complete align, extra space with  
new paper align)direction: rtl; → (Text to left align nہیں اور  
right align)

→ Img کے ساتھ text on cuiy align کیا جائے گا  
text کے form of top, middle, bottom align کیا  
گا اسی nہیں

css nہیں

img {

vertical-align: top;

: middle;

: bottom;

}

→ text کی decoration باریکے css ہیں

text-decoration: underline; → oscar: overline; → oscar

: none; → oscar

: line-through; → ~~oscar~~

text-transform: uppercase; → All Capital

: lowercase; → small alphabet

: capitalize; → word کی نہیں

one word Capital ہے

line-height: 30px; → call the text gap 30px;

text-indent: 50px; → when write 50px in gap first  
will get.

letter-spacing: 10px; → each letter of text 10px in  
gap apart.

word-spacing: 10px; → each word in text 10px in  
gap apart.

white-space: pre; → HTML text will remain  
like or which ever.

white-space: nowrap; → this text will not wrap to next line.

word-wrap: break-word; → when text is too long  
word wrap will wrap text to next line  
like this text will wrap to next line.

→ HTML Shadow (Text):

# shadow {  
 text-shadow: 5px 10px 5px red;  
}

→ Horizontal      Vertical      Blur      color

→ HTML

<p id="text"> lorem ipsum </p> → OSCAR

CSS

# text {

width: 500px;

border: 1px solid black; →

writing-mode: vertical-rl;

→ vertical-rl;

## PART 014

### LINKS

→ HTML में Link होने से Blue रंग का Underline होता है.

Ex. `<a href="https://www.oscarcareerpoint.com"> OSCAR </a>`  
 → `OSCAR`

→ a {

`text-decoration: none;`

}

]  
 ] → Underline off होता है.

→ Link की चर्चा 4 STAGE में होती है.

1. Normal (Link) → नियमित रूप से open होते हैं।

2. Visited → कोई विषयालय की जाहिरतों में इसकी रुपरेखा नहीं होती है।

3. Hover → Link पर मासिक रूप से होती है।

4. Active → जब Link पर click होता है।

↳ Order maintain होती रहती है.

→ css का

`a:link {`

`color: yellowgreen;`

}

→ OSCAR (Yellowgreen)

`a:visited {`

`color: blue;`

}

→ OSCAR (Blue → VISIT की बात)

`a:hover {`

`color: orange;`

}

→ OSCAR (orange → HOVER की बात)

`a:active {`

`color: red;`

}

→ OSCAR (Red → CLICK की बात)

$\rightarrow$  PS

font-size : 30px; → 118.5 128.011

Font - weight: Regular; → not bold  
                  : SOFT; → bold & soft.  
                  : Bold; → Bold font

font-style: italic; → 斜体字 (n. 12)

font-variant: small-caps; → Oscar ⇒ OSCAR

\* Website in different font use size 21 :-

Method-1 → Google font [click here](#)

Google → Google font → one or one to many  
font use (select in style)

```

graph TD
    SF[Selected family] --> E[Embedded]
    SF --> CL[Copy link]
    E --> PHS[PASTE in head section in HTML]
    CL --> CCS[Copy CSS code<br/>(one font family)]
    CCS --> PDCS[PASTE in different CSS Section.]
  
```

## Method - 2

# Free font download

Google → download free font for → open website

PASTE in ← [Copy file] → Extract ← download font  
folder ...

Ex. Folder name → front

filename → Aclonica.HF

css 21

@font-face {

font-family : abc;

`src = "481('font/Aclonica.ttf');`

三

→ ये फॉन्ट नये CSS selector जैसे (font-family: abc;) PASTE करें।  
Font बदलना चाहिए।

→ એસ એફ કેલેન્ડર ફોન ફોન-ફેમિલી ની જીવાન  
Font Family Fall back હોય.

Ex.

Font-family : "abc"; "Poppins", sans-serif;

user font

open font (Google)

જોણાં જુન ફોન એટે નથે વારે લે જોણ ફોન હોય હોય,

ચ) સાંકેદ કોઈ browser font support ન હે તો ફોન family ની  
એસ એફ એફ ફોન નાખાની હોય તો જીવાન હોય.

ચ) Font-family ની ફોન એ નાખી Single word ની હોય હોય  
તેને (" ") એ હોય હોય ન હોય તી એવી વાલે. એની  
Single word એવી એવી એ હોય હોય (" ") ની જ નહીં.

→ Short hand for Font :

font : italic Small-Caps 600 20px "abc";  
↑      ↓      ↑      ↑      ↓  
Style      Variant      weight      size      font name

(order change ન હોય, અજરૂરી ન હોય તો  
દુઃખી શક્યાનું હોય).



Oscar  
CAREER POINT

A Tradition of Excellence

Contact: +91 9157929797

Email: info@osccareerpoint.com

Web: www.osccareerpoint.com

Address: 404, Shubh Square, Opp Venus Hospital, Lal Darwaja, Surat

FOLLOW US ON:

PART 16

## CSS Units

(Length measure साल एवं अमेरिकी एवं)

→ We use length in width, height, margin,  
padding, font-size.

→ There are two type of unit in CSS



### Absolute

(स्थिति नहीं बदलती है)

1. CM
2. MM
3. Inch (5 in)
4. Px ✓ (1 inch = 96 px)
5. Pt (1 inch = 72 pt)
6. Pc (1 pc = 12 pt)

### Relative

(स्थिति बदलती है)

1. Em (1em → parent की साल)
2. Rem (em की साल वा इसकी)
3. Vh (1% of view port height)
4. Vw (1% of view port width)
5. % (relative to parent)

PART 17

## CSS Cursor

क्षेत्र छोड़ा है तभी यह बदलता है।  
पांख Hand वाली शब्द, ताकू  
सारे वाली शब्द लेता है।

- Cursor : auto; → link → Hand, cursor: pointer; → ↑, Text → default line
- : all-scroll; → ↗
- : Cell; → +
- : col-resize; → ↔
- : crosshair; → +
- : e-resize; → ↔
- : grab; → ⌂
- : help; → ↗
- : not-allowed; → ❌
- : pointer; → ⚡
- : progress; → ↗
- : wait; → ⏳
- : zoom-in; → 🔎

google mail  
बैठा खिला दिया जा सकता है।  
धृति - अंगूष्ठ।

PART 18

## CSS Important

[CSS में Priority (order) नुम्बर वाले होते हैं, अब तक सीधे मापदण्ड के साथ CSS लिगाचर्ट होते हैं जो उनमें से अपने लिए Important मापदण्ड का style लगाते हैं ] [जूदा नहीं हो सकता, 3rd party का code होता है और काम करता है ].

→ `div {`

`border: 1px solid red !important;`

`}`

`div {`

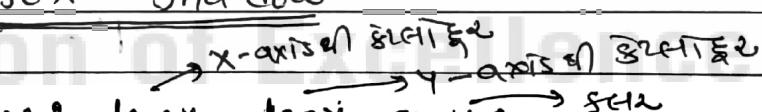
`border: 1px solid green;`

`}`

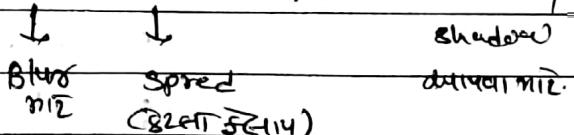
→ output : border की colour → red अपर्याप्त होले green priority में नहीं आवर्त होय.

PART 19

## Box Shadow

→ `box-shadow: 10px 10px 15px 3px gray;` 

`: 10px 10px 15px 3px gray;` → अपर्याप्त

  
 blur  
spread  
अपर्याप्त

`: 10px 10px 15px 3px gray inset;` → अपर्याप्त.

`: 10px 10px 15px 3px gray, -10px 10px 15px 3px red;`

↳ multiple shadow अपर्याप्त

PART 20

## CSS Opacity

(सिंह एलेमेंट अंतर्गत  
अपारिया स्टील)

Opacity = 0 → Full transparent

Opacity = 1 → No transparent

→ Color-box की opacity 0.5 में से 1 लिख त्रापशील हो।

opacity : 0.5;

मिला.

→ Image की transparency कैसे करें।

→ Image की hover स्टेट में CSS में,

img : hover {

    opacity : 0.3;

}

Image normal or व्यापक रूप

स्टेट में उपरोक्त लाइन 0.3 की

opacity की लिखें।

→ यहाँ की Colorbox की opacity का लिए उसका overlap होता है।  
इसकी overlap की mix-color कीजिए।

## CSS Filter

(Filter कोई Damage  
दर्शाने का वाला है, mobile की  
photo की Filter करने का)

img {

    filter : blur(10px);

    : brightness(50%);

    : contrast(70%);

    : drop-shadow(8px 8px 5px red);

    : grayscale(100%); → Black & white स्टेट।

    : hue(90deg); → Color circle जैसा effect

    : invert(100%);

    : opacity(30%);

    : saturate(30%);

    : sepia(30%);

यहाँ की अपेक्षा एक विवर

effect का लिए देखिए।

filter : blur(10px), hue(90deg);

g



PART → 22

## Image Sprites

प्राप्त दिया Image को सीखें  
सीधे Image करवाने, CSS में  
एक दिया Image को बनाएंगे  
सीधा टॉप Image sprite.

→ एक background Image में से  
चाहिए है।

→ जिस विभिन्न दिया Image website पर लोड होने  
के अपने browser में hand scaling करने वाले हैं।  
website के load time के बहुत कम होता है।  
Image के User ने भावे Combine set के लिए Image को  
न लेना। Coordinate (Photoshop के मद्देनज़र कोरिन) की  
रूपी ने इसके लिए Image को छापे हुए तो Display  
कर दिया है।

→ Ex. Image combine →

India	2
AUS	1
ENG	1

CSS में

# sprite {

height: 144px; → India flag

background-position: 0px -288px; → AUS flag

: 0px -396px; → ENG flag

→ colors ने smoothly  
mix कर दिया।

PART → 23

## CSS Gradients

2 types :-

① Linear Gradients

② Radial Gradients

→ ① Linear Gradient:

# div {

height: 500px;

border: 1px solid red;

background-image: linear-gradient(green, blue); →

green  
blue

: linear-gradient(green, blue, red); →

green  
blue  
red

: linear-gradient(to right, green, blue); →

: linear-gradient(to bottom right, green, blue); →

: linear-gradient(160deg, green, blue);

## (2) Radial - Gradient

10 → **examples**

background-image : radial-gradient (red, green);  
                          : radial-gradient (red, green, blue);  
                          : radial-gradient (circle, red, green, blue);  
                          : radial-gradient (circle, red 20%, green 30%, blue 50%);

PART 24

## CSS Overflow

(Block Element की height

में से कौन सा बोर्डर होगा)

Container से कौन Context

आयामों की विधि क्या है?

# overflow {

background-color : whitesmoke;

border : 1px solid red;

width : 500px;

height : 300px;

overflow : visible; → by default value or style.

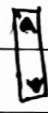
(content, container of child element).

overflow : hidden; → (content, container of child element will not be visible).

overflow : scroll; → (scroll x and y axis will be visible).

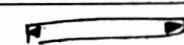
overflow-x : hidden; } →

overflow-y : scroll; } →



overflow-x : scroll; } →

overflow-y : hidden; } →



overflow : auto; (self axis का मार्ग scroll करता है तो जल्दी

stop हो जाता है).

PART 25

### CSS Resizable

(only block element).

CSS API,

(overflow: visible or all scrollable area)

[~~styling~~ user-resize  
scrollable].

resize: both; → x-axis & y-axis both scrollable  
resize set also in style.

resize: horizontal; → only horizontal or.

resize: vertical; → only vertical.

resize: none; → no resize set or.

PART 26

### CSS Lists

↳ Ordered list of numbers  
change scroll style;

↳ Unordered list of icons  
change scroll style;

OL {

list-style-type: upper-alpha;  
: lower-alpha;  
: lower-roman;  
: upper-roman;  
: decimal; → 1.  
: decimal-leading-zero; → 01

}

UL {

list-style-type: none;  
: disc; → ●  
: circle; → ○  
: square; → □

}

→ Icon of scroll bar image

scroll by click,

\* UL {

list-style-image: url('oscar.png');  
border: 1px solid black;  
list-style-position: outside; → • OSCAR  
: inside;

}

• OSCAR

UL {

list-style-image: url('oscar.png');  
}

(Image of size will be controlled)

PART 27

## CSS Tables

→ HTML की दूसरी चर्किती है।

→ CSS की,

table {

border: 1px solid black;

border-collapse: collapse; → cell वर्षे border (mix) हो।

: separate; → cell border एकत्र नहीं हो।  
By default collapse हो।

border-spacing: 15px 25px;

empty-cells: hide; → अंतर्गत डेटा को नहीं दिखाए।

→ table-layout: auto; → better view हो।

: fixed; → each column नियन्त्रित हो।

width: 100%; → each column नियन्त्रित हो।

caption {

border: 1px solid black;

Caption-side: bottom; → caption नियन्त्रित हो।

tr {

border: 1px solid black; → border एकत्र हो।

th {

border: 1px solid black;

height: 30px;

padding: 15px;

text-align: right;

Vertical-align: middle;

: top;

: bottom;

3

PART 2

## CSS Function

Functions  
rgb()  
url()

① calc function :

→ HTML

```
<div id="fun"></div>
```

CSS

```
#fun{
```

```
background-color: red;  
height: 200px;
```

```
width: calc(100% - 200px);
```

output :



css area body of

other attr of this div

(calc attr of this)

calc function eff.

②

Var (variable) Function :

(अपनी सी लोर चेंज करने की क्षमता  
सी अपनी सी लोर चेंज करने की क्षमता, तो  
वारियल कोडिनग में सी लोर चेंज करने  
की क्षमता उपलब्ध है।)

CSS

```
:root{
```

```
--primary-color: red;
```

```
--btn-color: lightgreen;
```

```
--text-color: black;
```

}

```
#fun{
```

```
background-color: Var(--primary-color);
```

```
height: 200px;
```

```
width: calc(100% - 200px);
```

}

a{

```
padding: 10px 30px;
```

```
margin: 20px;
```

```
background-color: Var(--btn-color);
```

```
color: Var(--text-color);
```

```
outline: none;
```

variable का  
CSS की define की।

PART 29

## Box Sizing

→ CSS में height वाले width का अपना नहीं है, padding वाले border वाले तो box का size नहीं बढ़ाते।  
यहाँ, padding वाले border वाले content वाले include  
होते हैं। Box Sizing property का उपयोग किसी भी विशेष रूप से नहीं होता।

CSS 21

# div {

background-color: blue;

width: 600px;

height: 150px;

padding: 50px;

border: 10px solid black;

box-sizing: border-box; → height & width  
content वाले विशेष रूप से नहीं होते।

: Content-box; → height & width  
content वाले विशेष रूप से नहीं होते।

PART 30

## Initial vs Inherit

by default  
CSS

by percent  
CSS

HTML 21

```
<div id="green">
    Green Text <br>
    <a href="#" id="abc">link 1 </a> <br>
    <a href="#" id="initial">link 2 </a> <br>
    <a href="#" id="inherit">link 3 </a>
</div>
```

CSS 21

# green {

color: green;

}

# abc {

color: red;

}

# initial {

color: initial;

}

# inherit {

color: inherit;

}

Output:

Green text → Green

link 1 → red

link 2 → blue (by default)

link 3 → Green (by parents).

PART 3)

## object-fit

[image & video all property (H10)]

img {

border: 1px solid red;

width: 500px;

height: 400px;

object-fit: contain; → image will resize w.r.t aspect ratio change itself while border will fit it.

: fill; → borders will Image original w.r.t aspect ratio change.

: cover; → fit in border, Aspect ration not change but img crop.

object-position: top centre; → Img top w.r.t original.

: ~~top~~ left; → Img left w.r.t original (contain ni).

PART 3)

## Pseudo-classes

(: → 1 colon).

Element all Special stat of Pseudo-class selector.

Ex. :link (on mouse over), :hover, :active, :first-child, :last-child, :nth-child(4), :nth-child(odd), :nth-child(3n+1), :nth-child(2n+1) etc.

Selector: Pseudo-class {

}

↳ Ex. a :link {

a : hover {

}

}

→ Element & Selector List all having the nth-child - nth-child List of nth-child - nth-child

Style defined by,

li : first-child { → when list will change }

: last-child { → when list will change }

: nth-child(4) { → 4th item in list will change }

: nth-child(odd) { → odd number in style } even

: nth-child(3n+1) { → math formula will give }

: nth-child(2n+1) { → even all odd in change }

color: red;

→ Focus selector

[SIC] Input text currency by default border click gradient border border color change

input: focus {

border: 3px solid red;

}

→ कृपया नो Pseudo classes ↗ और Google की Search से।

### PART 33

#### Pseudo - Element

(:: → 2 colon)

Specify the style in special insert से CSS through J.

#### Syntax

Selector :: pseudo-element {

}

⇒ P :: first-letter {

→ नेटवर्क style से।

:: first-line {

→ नेटवर्क line of style से।

font-size: 35px;

color: red;

}

⇒ h1 :: before {

→ CSS से

Content: url(images/oscar.png);

image h1  
नेटवर्क  
insert  
से।

{

⇒ h1 :: after {

Content: "added after"; → text add  
गर्ने मिले।

{

⇒ :: Selection {

color: red;

Selection or selection

background: yellow;

all style की तरह।

{

PART 3

## Display

### Visibility

HTML की ट्रॉपी नो प्रोपर्टी हैं

Rectangular box वाले, एक box के लिए

behaviour) जैसा कि Display में Depend है

block	inline	inline-block	
✓	✗	✗	New line की START होती है
✓	✗	✗	Full width होती है
✓	✗	✓	height & width Apply होती है

HTML में

```
<div class="display">
    This is a div 1
</div>
```

```
<div class="display">
    This is a div 2
</div>
```

CSS में

display :

border : 1px solid red;

width : 50%;

height : 100px;

display : inline; → [ ] ;

: inline-block; → [ ] [ ]

(नहीं height & width)  
(comment की तरफ)

: inline-block; → (नहीं height & width)  
अद्वितीय रूप से Apply होती है

→ कोई ट्रॉपी element की अवधारणा नहीं, इसलिए यहाँ Use of each  
ट्रॉपी, CSS में,

display : none;

→ कोई ट्रॉपी element के लिए hide or बर्दाची रूप से Use  
Use गए अपनी CSS में,

• Visibility : hidden;

PART 3

## Position

web page ni element જેણું  
એવું બન્ધું તૈયાર કરી લે એવું  
સર્ટ

Static -> (HTML ના બદા Element ની position static છે) (by default)

**relative** (બાસાની જગતાની center ની level એન્ટ્રીઝ અને અંદર એલેમેન્ટ્સ  
ને તૈની જગતાની પ્રાપ્તિકા વર્ગે હોય)

**Absolute** (ને તૈની જગતાની પ્રાપ્તિકા વર્ગે હોય)

~~fixed~~

બાસાની જગતાની તૈના

parent position element

(જેણું position: relative, Absolute

~~fixed~~ માનુષ હોય તેણે

જ હશ્વયાનું).

(થ્રીડ ની 3 position)

આપણી હોય તો જ લાગશે.

રહ્યાય શી હોય, કો parent position  
ન હોય તાં તેણી (થ્રીડ ની parent નીચે).  
જો કોઈ ન હોય તો છેલ્લે બાયાં ના  
દિશાય શી હોય એવો હોય જેણાય  
તેણી એવી નો એલેમેન્ટ આવી જાય).

(એ) element ની  
viewport ની રાખાય થાં

Fix કરી લ્યાય તો, લાલ

scroll થાય થળા ત લાની

position છોડી ન હોય)

Example → HTML નિ parent Div નિ 3 Div element હાયા.

CSS નિ

# parent {

border: 1px solid red;

padding: 20px;

width: 80%;

position: relative;



# one {

width: 100px;

height: 100px;

background-color: green;

display: inline-block;

position: relative;  
top: 50px;  
left: 50px;

# two {

width: 100px;

height: 100px;

background-color: blue;

display: inline-block;

position: absolute;  
top: 0px;  
left: 0px;

# third {

width: 100px;

height: 100px;

background-color: black;

display: inline-block;

position: fixed;

bottom: 10px;

right: 10px;

### CSS Layers and Z-index:

→ जोड़ी Relative, Absolute वाले त्वारे Element विशेष  
ज्ञान Layer बनावें, जोड़ी ज्या Element ने जोड़ी  
(यदि इनपर दो तोनों Z-index में से) वह आपापामा  
आवेंगे.

CSS 21),

z-index: 1;

: 2;

: 3;

→ जो Layer ओर (यदि उसके दो तोनों Z-index में से)

Layer 2 वाले गोण दिवाने  
position: relative top & left set  
जारी.

(order जैसा नहीं होता).

PART 3

css Float and clear

→ block element ને અને પ્રાંત આં કોઈ ને  
block નિર્માણ (Inline નિ) Float કરીય છે.

Ex. ① Image & paragraph in combination section

② Navigation bars configuration (34).

HTML आइ	CSS आइ
<ul>	ul {
<li> Home </li>	background-color: green;
<li> About </li>	list-style-type: none;
<li> Gallery </li>	overflow: auto;
</ul>	}
	li {
left side of green element white color.	color: white;
	padding: 5px 10px;
	position: left;
	z-index: 1;

Outputs | Home About Gallery

→ float: right; → Right side of each element  
Testing in mobile view.)

clear → જોએ float કરીએ હોય હેઠળાનું ન લોડશે હોય  
 તેણે આ float લાગે હોય, clear ની પ્રાપ્ત હોય  
 થાણે હોય & floated element ની વોણું હોય  
 (left - right) એ કિસ્માની સુધીની અભિવ્યક્તિ હોય,  
 by default આ floated element ની વોણું હોય  
 હોણી space નાં હોય હોય એવું એ clear:both  
 હોણું હોય તો left & right ની એ કિસ્માની  
 નિયમો હોય હોય.

Ex

→ HTML में 3 Div दिये गए।

→ CSS में;

# Content {

width : 70%;

background-color : green;

height : 200px;

float : left;

}

# Sidebar {

width : 90%;

background-color : red;

height : 150px;

float : left;

}

# Footer {

background-color : yellow;

clear : both;

}

→ Content, Sidebar वाले

div का left & Right margin

मात्रा दिए गए हैं अतः

इनमें से बड़ी होगी।

→ clear : right; → बैंग दिये गए हैं अतः इनमें से बड़ी होगी।

float : right; मात्रा दिये गए हैं।

यह दिया गया है।

Right margin मात्रा भरते हैं, तो

clear : right; मात्रा दिये गए हैं।

इसके बारे में Right margin नहीं।

मात्रा दिया गया है। Right margin नहीं।

इसके बारे में (Right margin नहीं।)

PART - 88

## CSS 2D Transformations

(Inline element or all style child of it).

Block element,  
inline block element  
move, rotate, Scale,  
skew all will change.

(or style child all this)

→ block element,

inline-block element all

Transformation say us.

→ HTML all say we (Button) of all property much say.

### css all

display: inline-block;

→ transform: skew(20deg, 10deg);

: skewX(20deg);

: skewY(20deg);

→ transform: origin: 0% 100%;

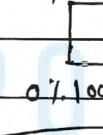


Skewer of rotate

say all

0%, 0%

100%, 0%



0%, 100%

100%, 100%

→ say button in hover say property say,

a: hover {

display: inline-block;

transform: translate(50px, 50px);

: rotate(20deg); → up, down

: scale(2, 3); → width×2, height×3

: scaleX(2); → width×2

: scaleY(3); → height×3

→ say div in matrix of transform say;

transform: matrix(scaleX(), skewY(), skewX(), scaleY(),  
translateX(), translateY());

→ transform: matrix(1, -0.2, 0, 1, 0, 0);

output:

Google (show)

PART #39

### CSS 3D Transforms

→ 3d transforms, descenders.

→ HTML ai,

com / perspective.

<div id = "cover" >

<div id = "content" > </div>

</div>

→ css ai

# Cover {

border : 2px solid red;

height : 200 px;

width : 200 px;

margin : 150 px;

}

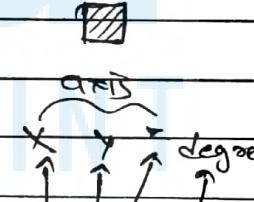
# Content {

height : 200 px;

width : 200 px;

background-color : yellow;

transform : perspective(300 px) rotate3d(0, 0, 0, 20deg)



: perspective(300px) rotate3d(50, 0, 0, 20deg) →

: perspective(300px) rotate3d(0, 50, 0, 20deg) →

: perspective(300px) rotate3d(0, 0, 50, 20deg) →

: perspective(300px) translate3d(0px, 0px, 0px); →

: perspective(300px) translate3d(50px, 0px, 0px); →

: perspective(300px) translate3d(0px, 50px, 0px); →

: perspective(300px) translate3d(0px, 0px, -50px); →

PART → 40

## CSS Transition

ऐसे Element के property से start वे (hover) के fast change की तरह smooth scroll & animation से all होते हैं।

→ HTML में div के hover के सभी transition property होते हैं।

CSS में

# div {

height: 100px;

width: 100px;

background-color: red;

margin: 100px;

transition-property: width; → क्या अपने हैं?

: width, background-color;

: All;

transition-duration: 500 ms; → क्या अपने हैं?

transition-delay: 500 ms; → क्या अपने हैं?

transition-timing-function: ease-in-out; → start slow, End fast

(अथवा start fast, end slow  
→ Start Fast, between)

except all property हैं।

transition: width 2s, height 3s; → shortcut है।

g

# div : hover {

height: 300px;

width: 200px;

g

PART 04

## CSS Animation

JS की जरूरी → स्क्रिप्ट की परवानगा  
 HTML Animation क्या है?  
 CSS 3 में यह दिया गया है, तो  
 [CSS 1, CSS 2 में ऐसा नहीं]

CSS की key frame → define सकते,  
 किसे कैसे सकते? → self से property change  
 सकते.  
 → self से self.  
 → self duration या time से.

1) HTML में वाले div को भी और background-color  
 दिया width के Animation कर सकते.

→ CSS में

@keyframes changeColor {

0% { background-color: red; } → start (from)  
 25% { background-color: blue; }  
 50% { width: 300px; }  
 75% { width: 250px; }  
 100% { background-color: yellow; }

}

# animate {

background-color: red;

height: 200px;

width: 200px;

animation-name: changeColor; → कैसे रोमांच

animation-direction: 4s; → इसे कैसे बदलें?

animation-iteration-count: infinite; } → असीमीय  
 : 3 ; } संख्या से.

}

→ यहाँ यहाँ सभी वाले div (EX - 2 div) को किसी दिव-2 में,

animation-direction: reverse;

: alternate;

: alternate-reverse;

} एक दिव में

एक दिव में एक बार।

जाएगा।

animation-timing-function: ease;

: ease-in;

: ease-out;

: ease-in-out;

: linear;

} दूसरी तरफ

timing function

ऐसा रोमांच।

Q) HTML में से 1 DIV में multiple style में Animation कैसे करें,

@ keyframes changecolor {

```
0% { background-color: yellow; border-radius: 0%; }
10% { background-color: green; }
20% { background-color: red; }
30% { background-color: orange; }
40% { background-color: red; }
50% { background-color: olive; }
60% { background-color: orange; }
70% { background-color: blue; }
80% { background-color: black; }
90% { background-color: yellow; }
100% { background-color: red; }
```

}

# animate {

background-color: red;

height: 300px;

width: 300px;

animation-name: changecolor;

animation-duration: 2s;

animation-iteration-count: infinite;

animation-timing-function: linear;

}

→ यही विधि Animation परियों ने भी बताया है।

PART-42

## Print style in CSS

→ यहाँ के HTML में जैसे DIV ही  
जैसे Image वह परालेग्रफ ही, वही  
print में वह आवश्यक होगा Image  
जैसे ब्राउज़र, font size जैसे जॉड विल्युम 50% width जैसे

सभी CSS ही

@ media print {

    p {

        font-size: 30px;

    }

    img {

        display: none;

    }

    div {

        width: 50%;

    }

PART-43 :

## Responsive Web Design :

[Web page जैसे ही स्क्रीन बदलता होता है Screen की जरूरि अपडेट होती है।  
Screen की जरूरि अपडेट होती है।]

→ कोई mobile या APP जैसे ही भी रिसेप्टिव website होना चाहिए।

→ यह website responsive नहीं होने वाली Screen (mobile)  
में जॉड जॉड जॉड थाय जाए (Zoom जॉड जॉड)।

\* Website → Right click → Inspect → open pop-up → : → [ ] → [ ]

Select different phone → minimize  
check (responsive or not ?).

PART 44

## Viewport Meta Tag

→ Bootstrap ✓  
→ CSS Frameworks  
→ Pure CSS  
→ Transformation CSS.

"तरे CSS Framework ने उपयोग करके वेबसाइट को Responsive scroll करने के लिए 3 कदम दिये हैं।"

- ① Viewport meta tag
- ② Media Queries
- ③ Images & Videos in responsive environment.

⇒ ① Viewport Meta Tag :

HTML में

- ① Image      ② Paragraph ] → mobile की ओर से देखा जाएगा। देखा जाएगा। देखा जाएगा।

HTML में,

<head>

<meta name="viewport" content="width=device-width,  
initial-scale=1.0">

not zoom, not zoom-out

PART 45

② ⇒

## Media Queries

"जो लोगों को Mobile, Computer, Tablet वेबसाइट की width में परेंगे अपनी-अपनी width के लिए-अपनी अपनी style तो लगा सकते हैं।"  
[ ऐसे Device के लिए जबकि आप ऐसे Device के लिए जबकि ]

[ browser के size को देकर → www.onlinemutilitywebsite.com ]

→ HTML में एक Div Tag की heading वाली (large-font)

(1) Responsive scroll करें,

→ प्रत्येक Mobile, i-pod, desktop वा child एवं उपयोगी है,  
 max-width ← max. allowed width का उपयोग.  
 min-width ← min. allowed width का उपयोग.

CSS 3.1

→ @ media screen and (max-width: 768px) {  
 # div 1 {

width: 50%;

}

# div 2 {

width: 50%;

}

}

→ ~~width: auto;~~ @ condition media query द्वारा दिया गया,

@ media screen and (min-width: 400px) and (max-width: 768px);  
 {

# div 1 {

width: 100%;

}

# div 2 {

width: 100%;

color: white;

background-color: black;

}

}

→ ③ Img वा video के Responsive सेट हैं,

img {

width: 100%;

height: auto;

}

video {

width: 100%;

height: auto;

}

PART 46

## Advanced Selectors

### (3) Attribute Selector

- ① Type Selector
- ② Id Selector
- ③ Class Selector
- ④ Pseudo class & Pseudo Element

↳ `a [href = "https://oscar.com"] {`

`color: red;`

`}`

→ Web page all  $\rightarrow$  link OSCAR.COM  $\rightarrow$  Red  
 $\rightarrow$  `color: red;`

↳ `input [type = "text"] {`

`color: green;`

`}`

### (4) Universal Selector [ Use asterisk (\*) , testing all element Use \* ]

Ex

`*`

`color: red;`

`z`

Style of child of  $\rightarrow$  style

Color of all  $\rightarrow$  height of  $\rightarrow$

width of  $\rightarrow$  on no border

all  $\rightarrow$

### (5) Combinator Selector

#### ① Descendant combinator (child at any stage)

Ex.

`.red-p span {`

`red-p class of`

`color: green;`  $\rightarrow$  all depth child of

`y`

`color set by`

(Stylized level of child (say)).

#### ② Child combinator (Direct child)

`.red-p > span {`

`color: green;`

`y`

### ③ Adjacent combinatory (right after sibling)

- red-p + h1 {

color: green;

}

(Same level element)

(not parent-child)

### ④ General sibling Combinatory (all sibling)

- red-p ~ h1 {

color: green;

}

[Same level all child]

[h1 at style will be]

### ⑤ Add Selector

[multiple at same time style will be]

- red-p, h1 {

color: green;

;

PART 47

### css Specificity

⇒ If one Element is styled by more than one rule, which style will be applicable?

Ans ↳ more specific rule will get more value & apply style.

#### → General Rule:

↳ ! important always has a higher value to apply.

↳ ! important & style at inline css apply first.

↳ also External css will Specificity check so.

### Specificity Rule

↳ (#) Id selector has highest value

↳ (\*) Universal selector has lowest value.

## Specificity Rule Trick (point system)

- ↳ 10,000 point → ! Important
- ↳ 1000 point → Inline
- ↳ 100 point → Id
- ↳ 10 point → class, attribute or pseudo-class
- ↳ 1 point → Element Selector & Pseudo-element
- ↳ 0 point → Universal Selector

→  $\text{P} \{$  → 1 point

color: green;

}

.red-P { → 10 point

color: red;

{

div.red-P { → 11 point ←

color: blue;

}

Apply  
CSS

→ Specificity same के बीच CSS Apply किया जाएगा.

PART 4

Multiple columns (like newspaper)  
more columns.

HTML में डिव में डिव heading डिव paragraph आदि.

→ CSS में,

div {

column-count: 3;

column-gap: 50px;

column-rule-style: solid;

column-rule-width: 1px;

column-rule-color: red;

column-rule: 1px solid red;

column-width: 300px;

}, h1 {

column-span: all;

}

output:

OSCAR CAREER POINT

Shortcut



A Tradition of Excellence

Date \_\_\_\_\_  
Page \_\_\_\_\_

PART 49

### Flexbox

(2009)

अंतर्गत एक विकल्प है जो वेब डिजाइन में फ्लॉट और पोजिशन के लिए अपनी ओपनी को बढ़ावा देता है।

Flex property परसेट्स कि एकाधिक सभी child के flexible size दृष्टि [width और height की नहीं] automatic control के लिए flex-direction: nowrap विकल्प दिया जाता है।

HTML में, यह div parents की ओर div है।

```
<div class = "Flex-container">  
  <div class = "Flex-child">one</div>  
  <div class = "flex-child">Two</div>  
  <div class = "flex-child">Three</div>  
  <div class = "flex-child">four</div>  
</div>
```



Oscar  
CAREER POINT

A Tradition of Excellence

Contact: +91 9157929797

Email: info@osccareerpoint.com

Web: www.osccareerpoint.com

Address: 404, Shubh Square, Opp Venus Hospital, Lal Darwaja, Surat

FOLLOW US ON:

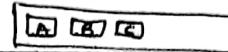
### css all

.flex - container {

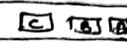
background-color : Yellow;

display : Flex; → अंदरूनी ने Flex दिलाया। child of flexible गोपनीय.

Flex-direction : row; →



: row - reverse; →



: column; →



(child of height दिलाया).

: column - reverse; →



flex - wrap : ~~flex~~ nowrap; → (width block of अंदरूनी गोपनीय नहीं रखा। Adjust करेंगे).  
width block of अंदरूनी गोपनीय नहीं रखा। Adjust करेंगे).

: wrap; →



(width अंदरूनी देता है इसकी).

: wrap - reverse; →



(लेटे खड़े जाएं).

Flex-flow : row wrap; →

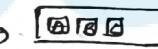


(row ने अपने अपने wrap को सजाया).

justify-content : center; →



: flex-start; →



: flex-end; →



: space-around; →



(Element of child) (अंदरूनी गोपनीय space).

: space-between; →



(Element of child) (अंदरूनी गोपनीय space).

height : 400px; →



(box of child अंदरूनी गोपनीय).

align-items : center; →



: flex-start; →



: flex-end; →



: stretch; →



Vertical  
work  
से

align-content : space-between; →



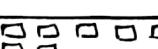
(empty area share) (अंदरूनी गोपनीय)

: space-around; →

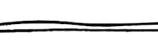


(empty area share) (अंदरूनी गोपनीय)

: flex-start; →



: flex-end; →



→ If any element is centered then all the other elements will be centered as well.  
Also, now Flex is very strong when it comes to center element.

justify-content: center; } {

height: 70vh; → (in body, height is set to 100% so view will be flexible).



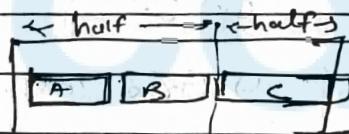
→ child element in order change will have the same effect as child element of CSS in order: 1, order: 2; and so on order-wise output will be.

→ Flex-grow child of any parent div of Extra space will cover flex.

Child-1 will → Flex-grow: 1;

Child-2 will → Flex-grow: 1;

Child-3 will → Flex-grow: 2;

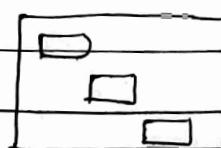


↓ opposite

→ Flex-shrink child of any parent and child will take width of parent and shrink if any child has overflow.

Child-1 will → Flex-shrink: 2; → (any child has overflow)

→ If child has flex-basis: 300px then it will be 300px width wise.



Child-1 will → align-self: flex-start;

Child-2 will → align-self: center;

Child-3 will → align-self: flex-end;

• flex-child {

background-color: wheat;

margin: 20px;

padding: 20px;

width: 100px;

PART 50

## CSS Grid (Oct, 2017)

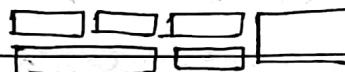
→ CSS Grid is the new layout model in CSS.

⇒ Difference between

→ Flex Layout (one dimension at a time change size & position).



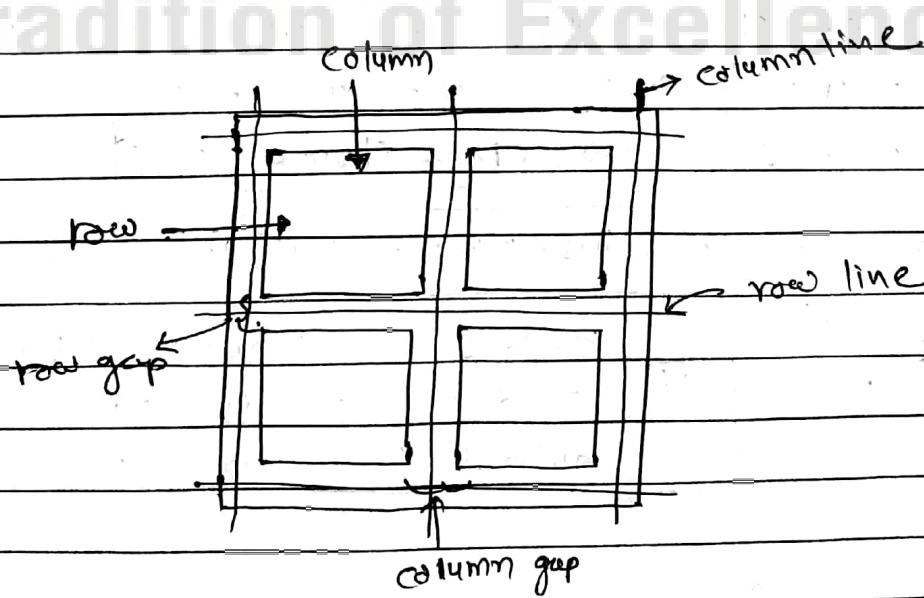
→ Grid layout (row & column or mix of both).  
(two dimensions used)



→ Both flexbox and CSS grid has its own importance and should be used for designing a web page.

↳ Grid works for 2 dimensional alignment.

↳ Flexbox is good for space distribution.



### HTML 21

```

<div class="grid-container">
    <div class="grid-item">1</div>
    <div class="grid-item">2< id="grid-2">
        > 3 <
        > 4 <
        > 5 <
        > 6 <
        > 7 <
        > 8 <
    <div class="grid-item">9</div>
</div>

```

### CSS 21

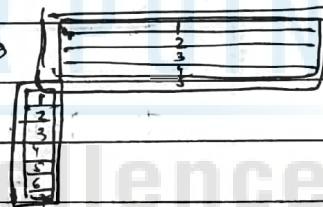
.grid-container {

background-color: Yellow;

padding: 20px;

display: grid; (by default) →

: inline-grid; →



grid-template-columns: auto 200px 50%; → 3列  
auto 200px 50%  
width分配  
宽度分配

width分配  
width分配

grid-template-rows: 100px 200px auto; (row分配)  
(row分配)

grid-column-gap: 30px; } column gap  
grid-row-gap: 20px; } row gap

justify-content: center;

: flex-start;

: flex-end;

: space-around;

: space-between;

: space-evenly;

} column container style

} row container style

Horizontal align

样式风格

样式风格

align - Content: center;  
: flex-start;  
: flex-end;  
: space-around;  
: space-between;  
: space-evenly;

~~Vertical~~  
~~align: center;~~

# grid-1 {

grid-column-start : 1; grid-column-end : 3; grid-column : 1 / 3;

grid-column : 1 / span 2;

OR

grid-column : 1 / column 3; grid-column : 1 / column 3; grid-column : 1 / column 3;

grid-1 at line-14  
line-3 column group  
mix sel & [ ].

column-1 line  
column-2 line  
column-3 line  
column-4 line

अगर Column एवं Row को एक ही दोष में combine करें

→ gold - Area 212

# gold-55

grid-area : 1 / 2 / 3 / 4

( ୧୦୯-୫ ଏ ୩୪୨ ଯୁଦ୍ଧୀ କମ୍ବିନେ ହେଲା ).

→ # dstd -5 {

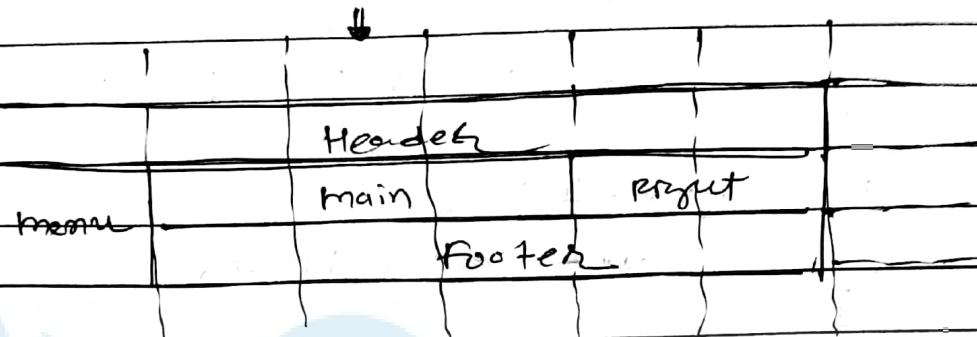
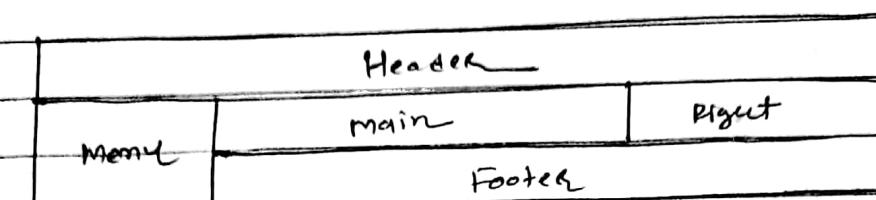
gold-area: 1 | 2 | Span 3 | Span 3;

(2 ଶତ ଏ ମେଟି 3 column ଏ କ୍ରମିକ  
କ୍ଷେତ୍ରରେ).

### • gold-items

```
background-color: wheat;  
border: 1px solid black;  
padding: 20px;  
font-size: 40px;  
text-align: center;
```

\* CSS grid & flex Layout Comparison.



HTML → 1 parent DIV with 5 child DIV (class with id).

CSS → .grid-container {

background-color: yellow;

padding: 20px;

display: grid;

grid-template-columns:

'header header header header header header'

'menu main main main right right'

'menu footer footer footer footer footer'

? grid-gap: 10px;

#grid-1 {

grid-area: header;

\* .grid-item {

background-color: wheat;

border: 1px solid black;

padding: 20px;

font-size: 40px;

text-align: center;

}

```
# grid-1 {  
    grid-area: header;  
}
```

```
# grid-2 {  
    grid-area: menu;  
}
```

```
# grid-3 {  
    grid-area: main;  
}
```

```
# grid-4 {  
    grid-area: right;  
}
```

```
# grid-5 {  
    grid-area: footer;  
}
```

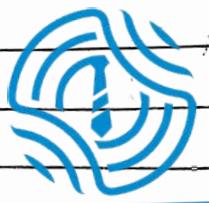
PART → 5

### CSS Validation

Google → CSS validation → W3 Validation

Congratulation ← No Error

URL
file upload
direct input



**Oscar**  
CAREER POINT

A Tradition of Excellence

PART 5

## Font Icons in CSS:

→ Font Awesome Icons:

Google → download font awesome



[fontawesome.com/download](https://fontawesome.com/download)



Free for web → (download this)

→ Add complete folder in project / fonts

→ Include fontawesome.css file (all.css)

```
<link rel="stylesheet" href="fonts/fontawesome/all.css">
```

→ How to use font awesome icons in HTML:

ex: <i class="fab fa-facebook-f"></i>



(Search font awesome facebook icon in google & copy HTML code & paste in your HTML file).

→ How to use font awesome icons in CSS:

ex: ↑ <span>mail @ google.com</span>

css: span::before {

font-family: "Font Awesome 5 Free";

or "Font Awesome 5 Brands";

content: "f0e0"; (you can search & find content from all.css file or from website)



## CONTACT US



+91 91579 29797



[oscarcareerpoint@gmail.com](mailto:oscarcareerpoint@gmail.com)



[www.oscarcareerpoint.com](http://www.oscarcareerpoint.com)



404, Shubh Square,  
Opp. Venus Hospital,  
Lal Darwaja, Surat-395004