

COMPLETE CSS COURSE



World Best CSS Course Ever

COURSE TOPICS



BASICS

- What is CSS?
- CSS rule structure
- How to use CSS in html?
- Types of Selectors?
- Comments in CSS
- Colors & Background Prop.
- CSS Units & Dimensions
- Fonts & Text Properties
- CSS Box Model
- Border, Padding & Margin
- CSS Display, Floats etc..
- CSS Positions & Z-index
- CSS Overflow

ADVANCED

- CSS3 Gradients
- CSS3 Shadows & Filters
- CSS Combinators
- CSS Pseudo Class & Element
- CSS3 Column Layout
- CSS3 Flexbox & Grid
- CSS3 Animations , Transform Transitions & Perspective
- CSS Specificity & Prefixes
- CSS Media & Cont. Queries
- CSS BEM, SMACSS etc..
- CSS3 Variables, Nesting etc..
- New CSS3 Features & Bonus

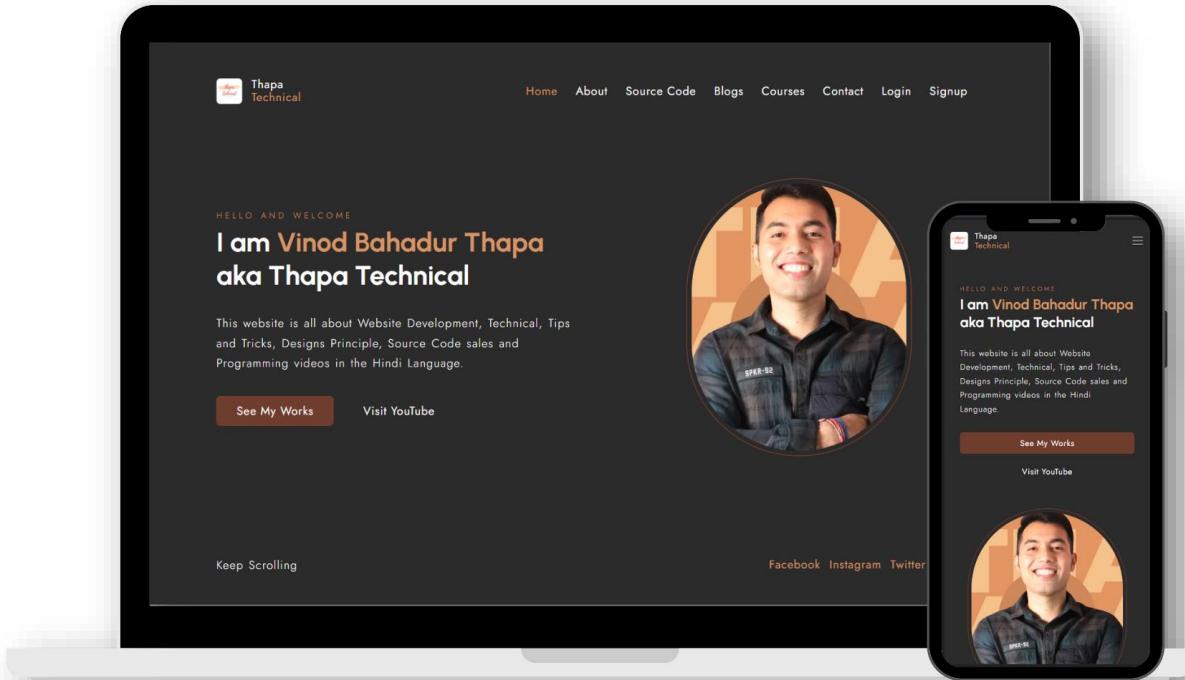
PRO LEVEL

- 50+ Interview Questions
- 30+ Tips & Tricks
- Complete Notes + Codes
- 100+ Animated Slides
- 10+ Animated Projects
- PROJECT** – Complete Multipage WEBSITE Using HTML & CSS 💚
- CSS Performance and Optimization
- SEO + Testing + LIVE HOSTING (free & Paid)



01

HOW WEBSITE WORKS?



[Subscribe Thapa Technical](#)

CLIENT VS SERVER



BUILDING BLOCK OF WEBSITE

HTML

Provides the structure and content of a webpage.

CSS

Styles and designs the appearance of the webpage

JS

Adds interactivity and dynamic behavior to the webpage.



REAL WORLD

Cascading Style Sheets

EXAMPLE



01

What is CSS?



css





css

Cascading Style Sheets



What is CSS?

- **Cascading Style Sheets (CSS)** is a stylesheet language used to describe the look & presentation of a document written in HTML or XML. It defines how elements are displayed on a web page.
- CSS makes websites visually appealing and user-friendly.



How to code CSS ?

```
h1 {  
    color: red;  
    font-size: 32px;  
}
```



Creators & Maintenance of CSS?

- CSS was created by a group of individuals known as the [World Wide Web Consortium](#) (W3C).
- Primary contributors include [Håkon Wium Lie](#) and [Bert Bos](#).
- CSS standards and specifications are [maintained by the W3C](#).
- A community of web developers, designers, and browser vendors also contribute to its evolution.
- Regular updates and new versions ensure CSS remains relevant and adaptable to changing web design needs.

CSS History (Version)

CSS to CSS3

01

CSS1 (1996)

Limited properties and selectors.

02

CSS2 (1998)

CSS 2 was released and work on CSS 3 began

03

CSS 2.1 (2011)

CSS 2.1 was released, which fixed the errors found in CSS 2



02

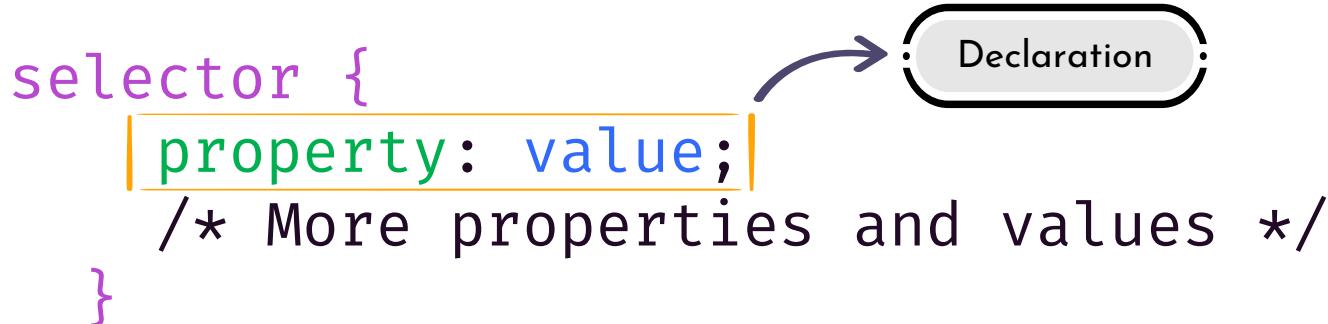
CSS Rule Structure



CSS Rule Structure

- A **CSS rule** consists of a **selector** and a **declaration block**.
- **Declarations** are used to define the style properties for the selected elements.
- Each declaration consists of a property and a value.

```
selector {  
    property: value;  
    /* More properties and values */  
}
```

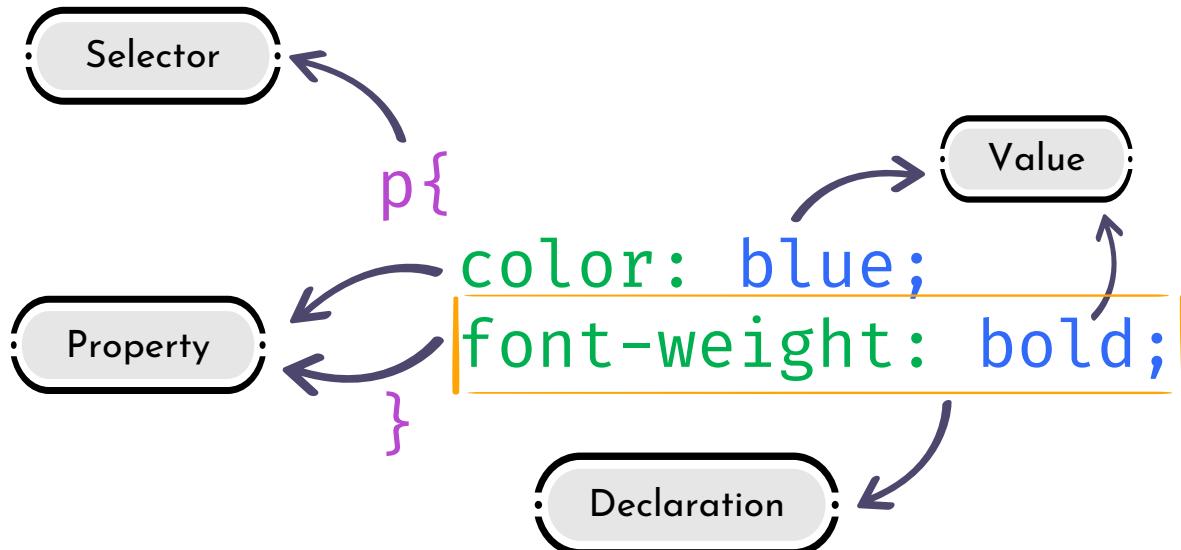


The diagram illustrates the structure of a CSS rule. It shows a code snippet with a purple 'selector {' followed by a yellow 'property: value;'. An arrow points from the yellow box to a rounded rectangle labeled 'Declaration'.

CSS Rule Structure



Syntax – CSS



03

How to use css in HTML?



3 Ways

01

02

03

Inline CSS

Style applied directly to an HTML element using the `style` attribute.

Internal CSS

Styles defined within the `<style>` tag within the HTML document's `<head>`.

External CSS

Styles stored in `separate .css files` linked to the HTML document using the `<link>` element.





04

Types of Selectors

CSS Tag / Element Selector

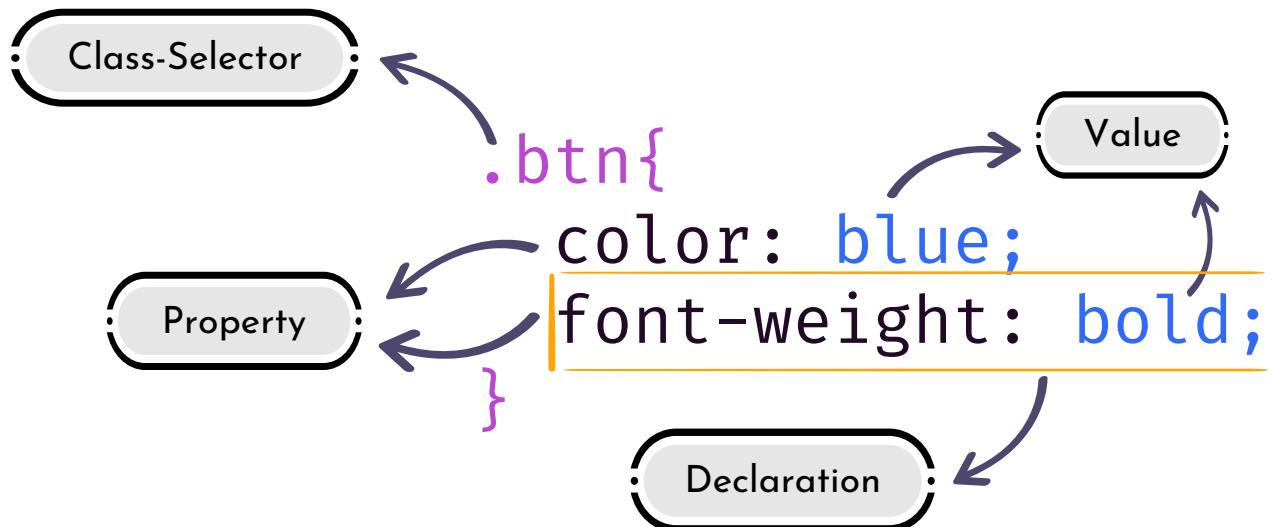


Class Selector – HTML / CSS

```
<section>
  <button class="btn">
    Subscribe Thapa Technical
  </button>
</section>
```



Class Selector - CSS

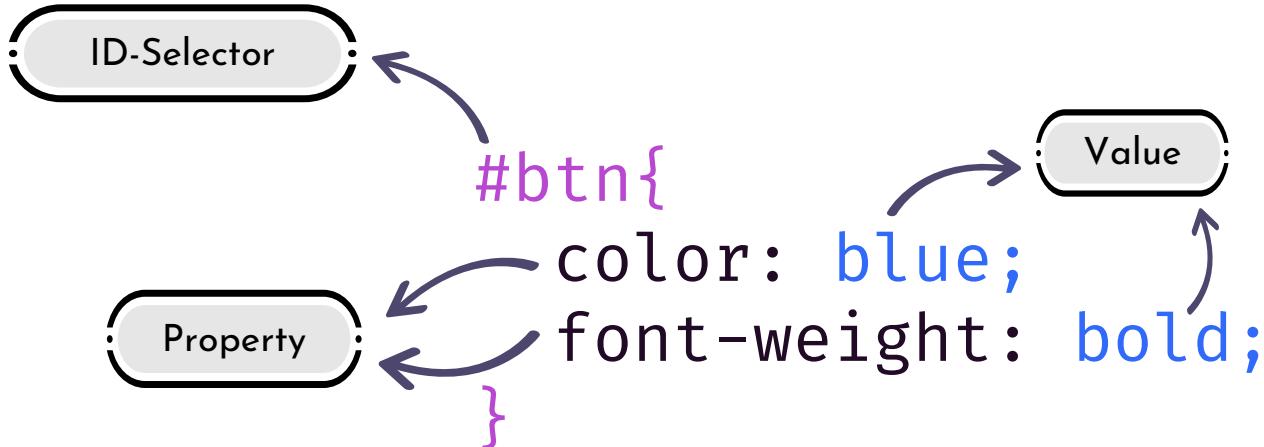


ID Selector – HTML / CSS

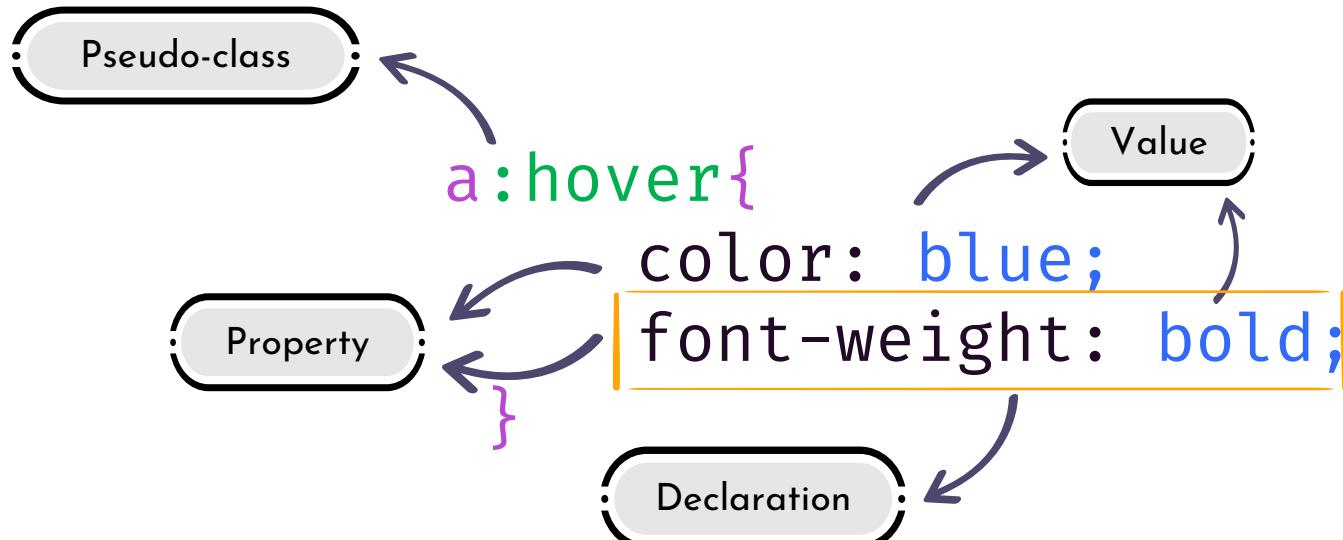
```
<section>
  <button id="btn">
    Subscribe Thapa Technical
  </button>
</section>
```

ID-Selector

ID Selector – CSS



Pseudo-Class – CSS





05

css Comments

CSS Comment

```
/* This is a comment */
```



06

COLORS IN CSS



RGB – CSS

01

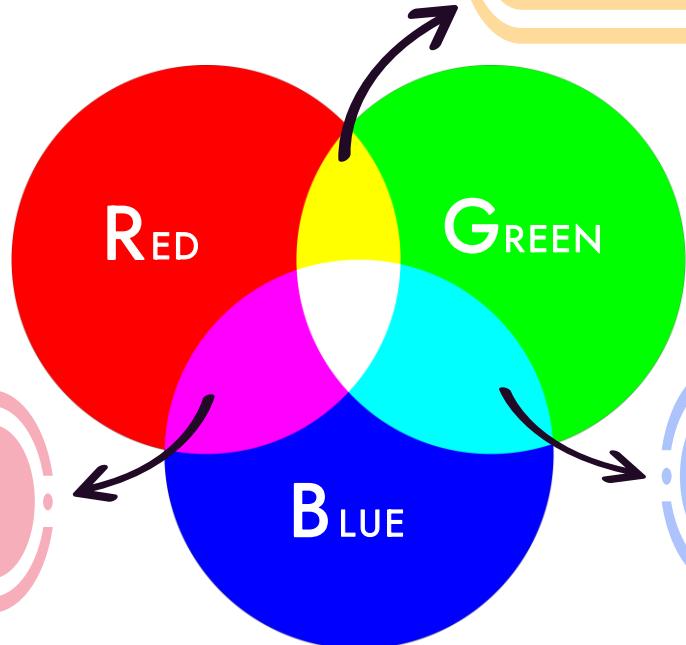
RED – #FF 00 00
255, 0, 0

02

GREEN – #00 FF 00
0, 255, 0

03

BLUE – #00 00 FF
0, 0, 255



YELLOW
RGB = 255, 255, 0
HEX = #FF FF 00

Magenta
RGB = 255, 0, 255
HEX = #FF 00 FF

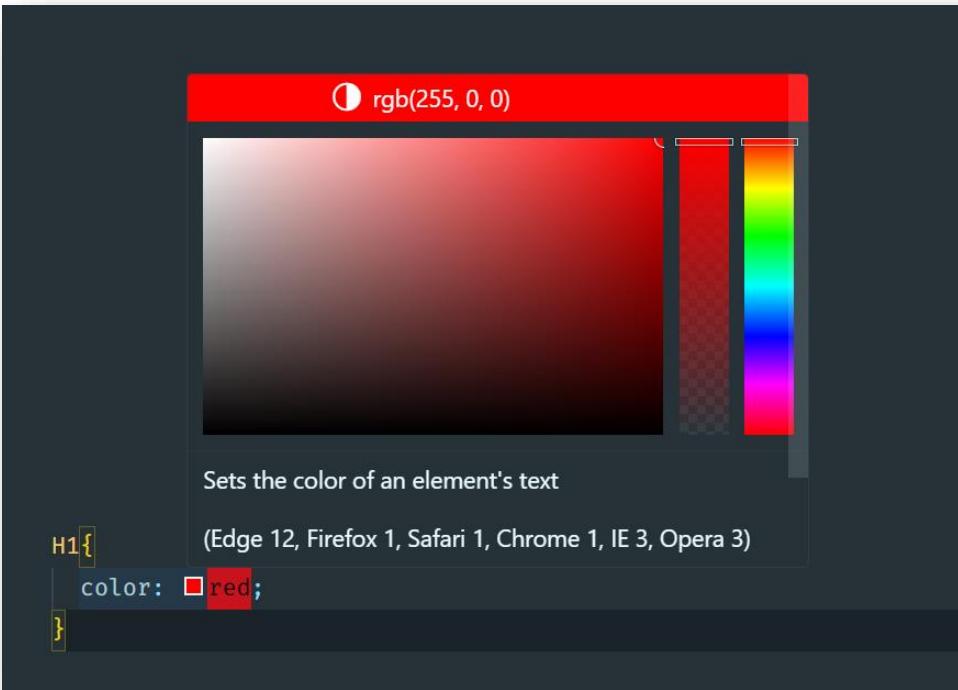
CYAN
RGB = 0, 255, 255
HEX = #00 FF FF



COLORS (BONUS)



GRAPHIC DESGINER – CSS



1

Hue: A pure color

2

Tint: A pure color with just white added

3

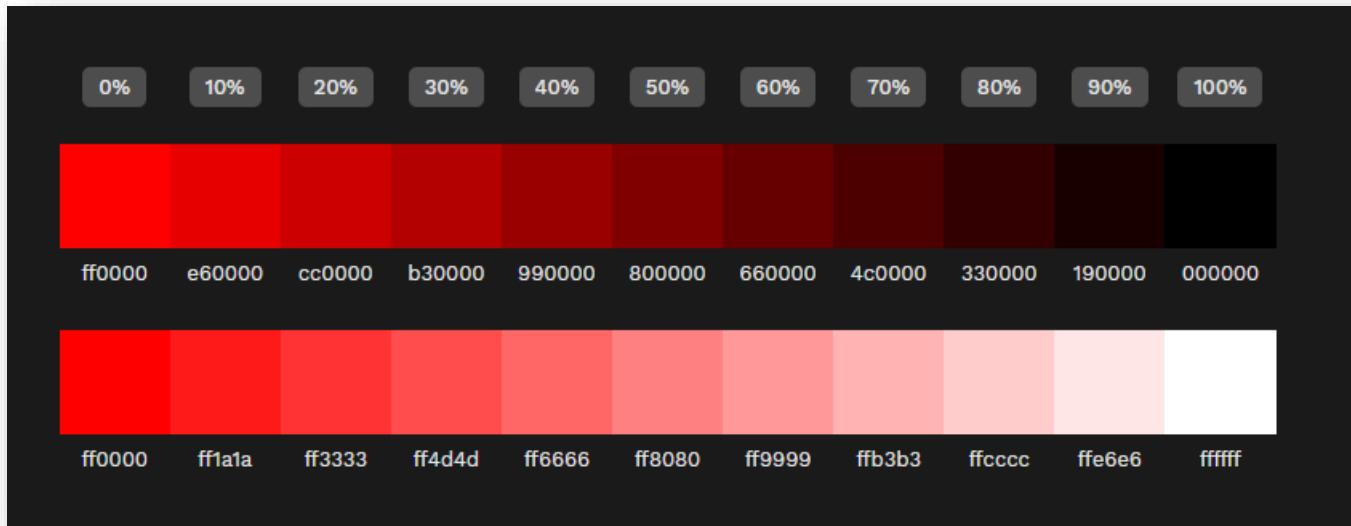
Shade: a pure color with just black added

4

Tone: A pure color with just grey added



GRAPHIC DESGINER – CSS



07

BACKGROUND IN CSS



BACKGROUND PROPERTIES – CSS

- 01 background-color
- 02 background-image
- 03 background-repeat
- 04 background-position
- 05 background-size
- 06 background-attachment

08

UNITS IN CSS



Absolute Lengths

- 01 Pexels (px)
- 02 Inches (in)
- 03 Points (pt)

Relative Lengths

- 04 em
- 05 Root em (rem)
- 06 Percentage (%)

08

FONTS IN CSS



FONT PROPERTIES – CSS

- 01** Font Family
- 02** Font Size
- 03** Font Weight
- 04** Font Style
- 05** Font Variant

09

TEXTS IN CSS



TEXT PROPERTIES – CSS

01

Text Alignment

02

Text Decoration

03

Text Transform

04

Text Spacing

05

Text Shadow

TEXT SHADOW - CSS

text-shadow: 2px 2px 1px blue;

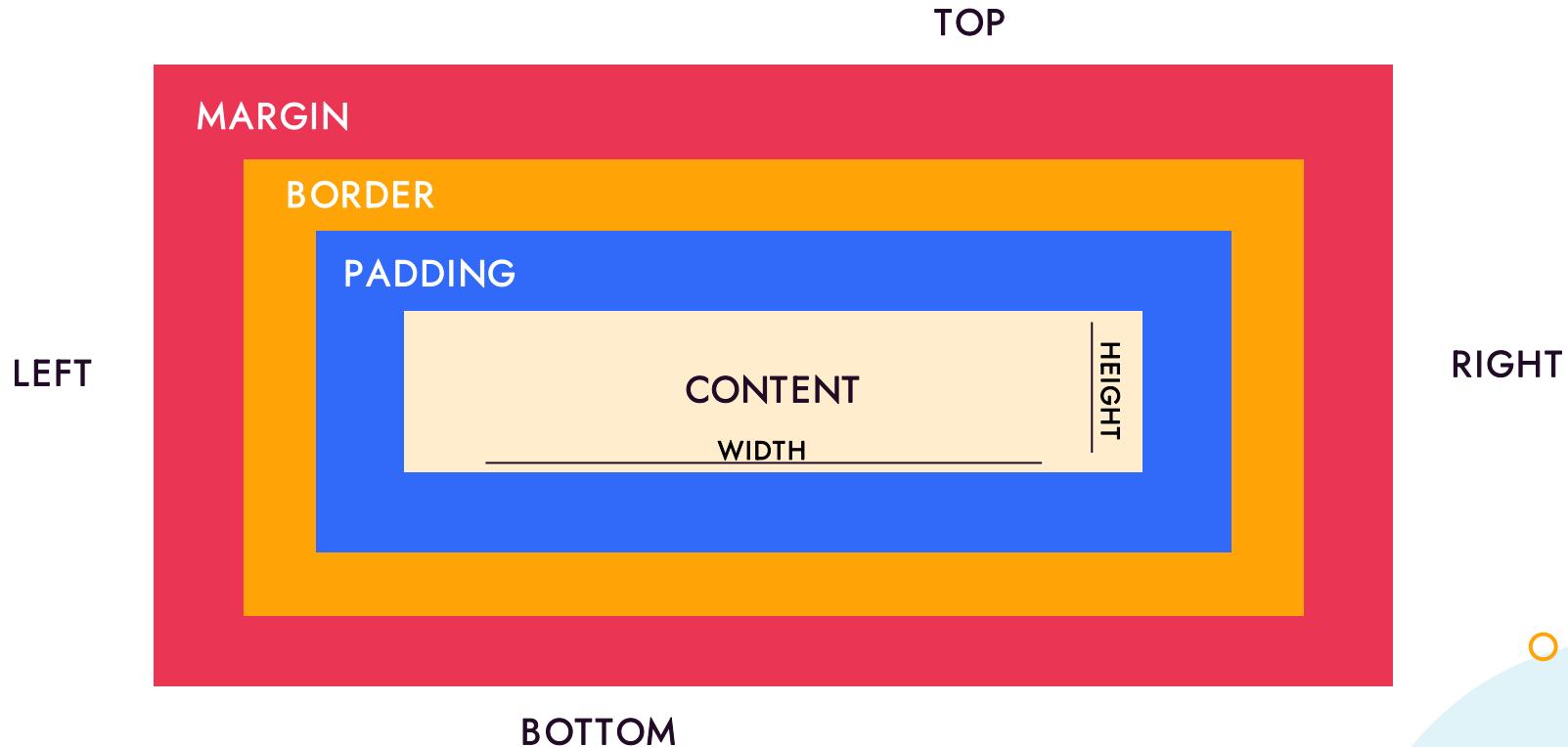
Offset-Y

Blur-radius

Offset-X

Color

BOX MODEL – CSS



10

BORDER IN CSS



Border – CSS

```
border: 2px solid blue;
```

Border-Style

Border-Color

Border-Width

PADDING IN CSS



Padding – CSS

padding: 2px 4px 3px 5px;

Padding-Right

Padding-Left

Padding-Top

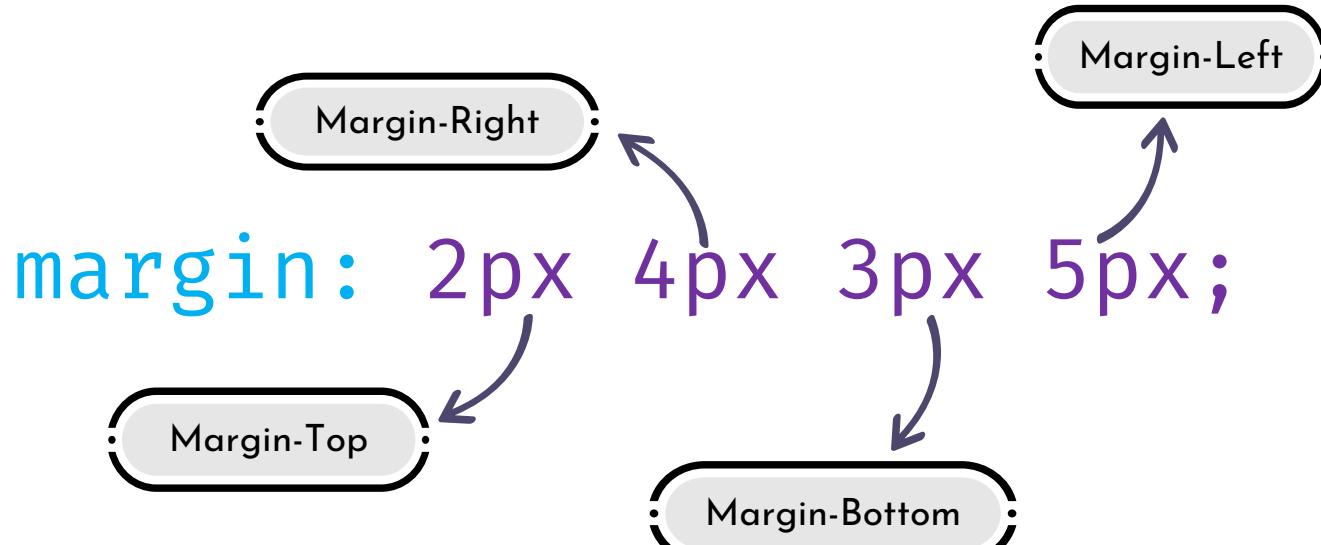
Padding-Bottom

12

MARGIN IN CSS



Margin – CSS

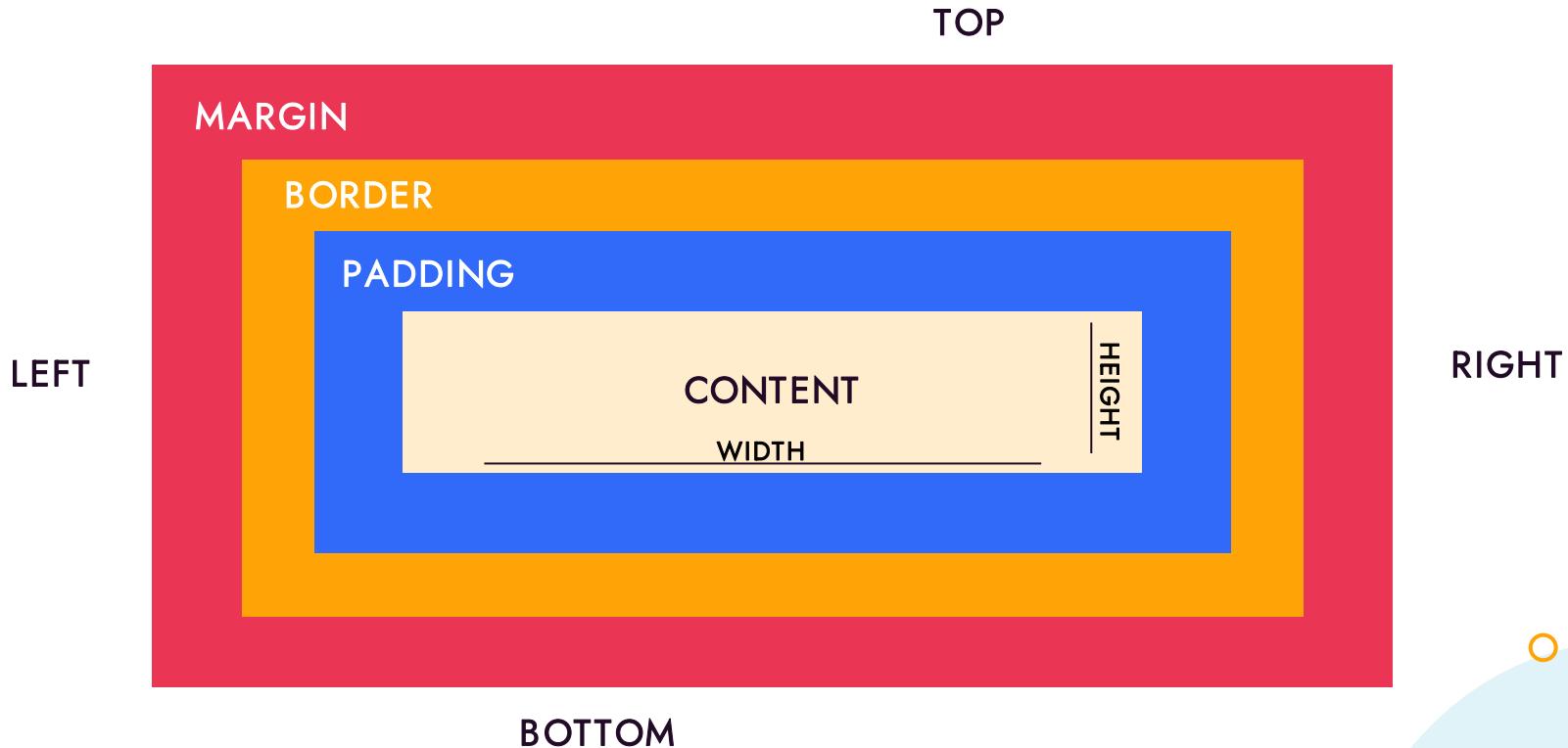


13

BOX-MODEL IN CSS



BOX MODEL – CSS

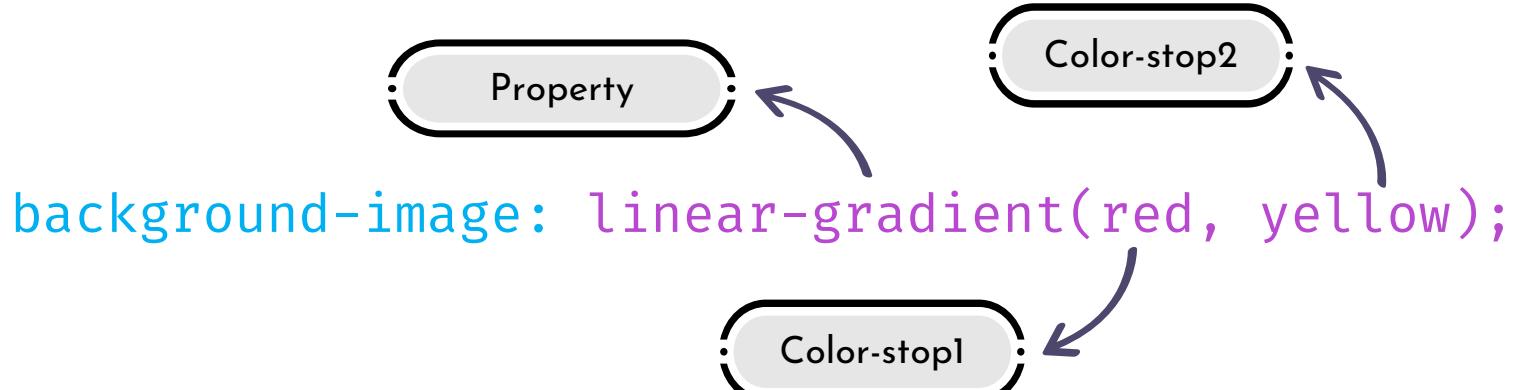




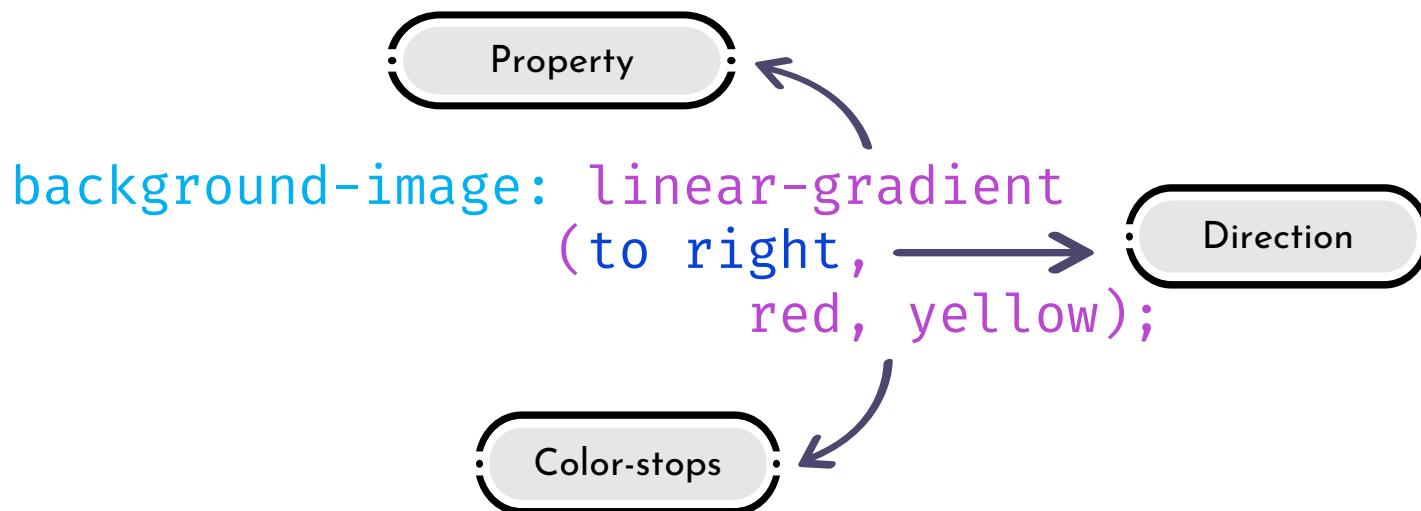
14

css Gradients

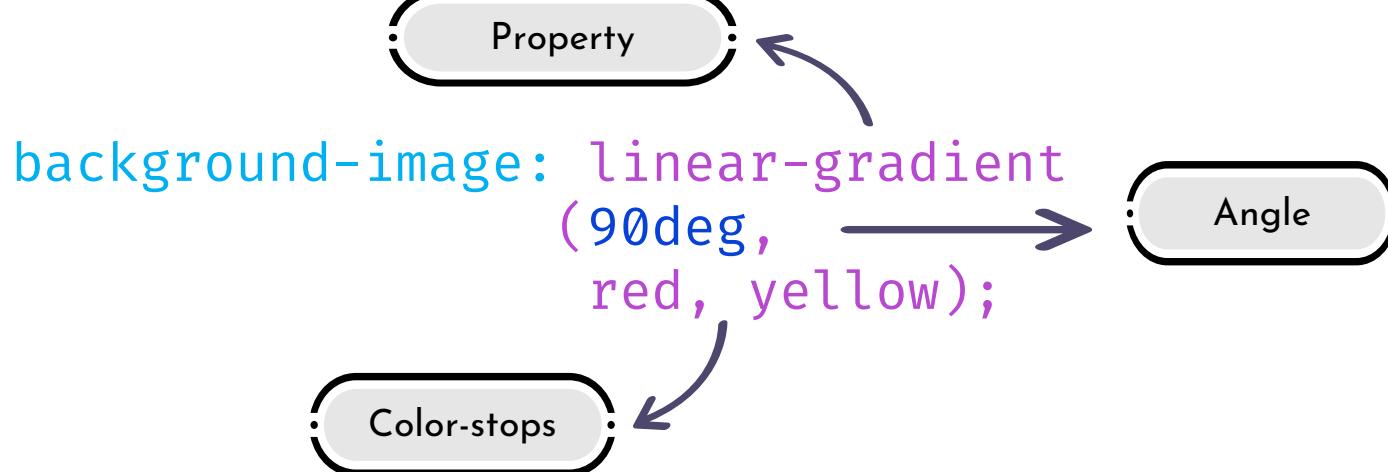
Gradient – CSS



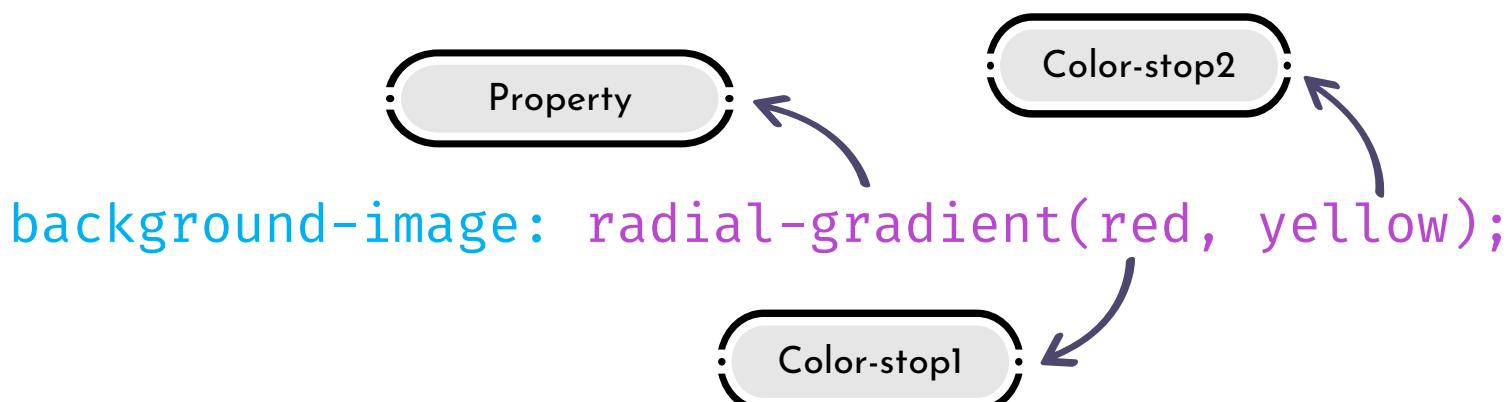
Gradient with Direction – CSS



Gradient with Angles – CSS



Radial Gradient – CSS



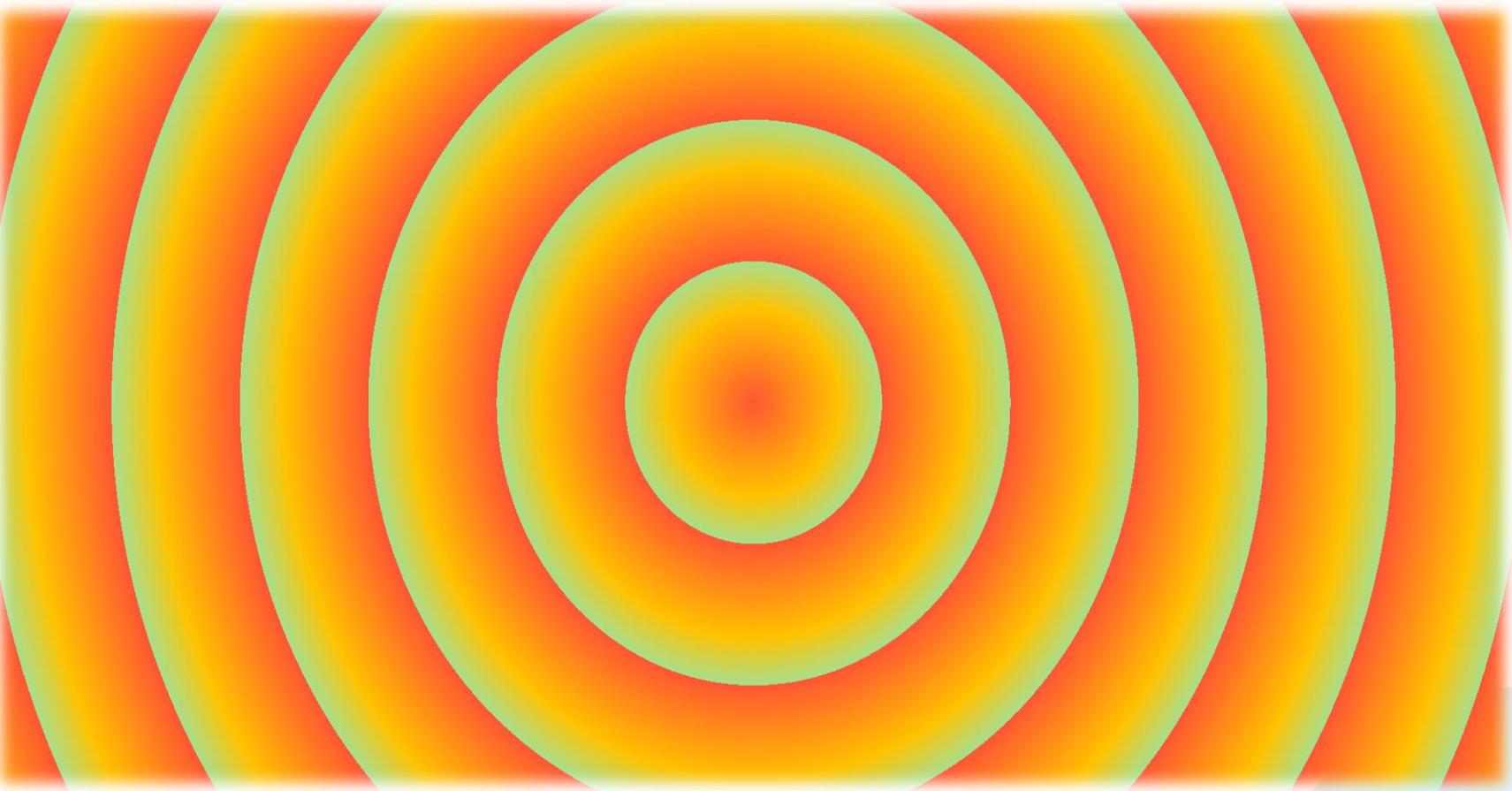
Gradient with Shape – CSS

Property

```
background-image: linear-gradient  
    (circle,  
     red, yellow);
```

Shape

Color-stops



15

BOX SHADOW IN CSS



BOX SHADOW – CSS

box-shadow: -2px 1px 2px 4px #61677a;

The diagram illustrates the components of a CSS box-shadow declaration. It features five rounded rectangular boxes arranged in a grid-like pattern. The first row contains 'Offset-Y' at the top left and 'Density/Spread' at the top right. The second row contains 'Offset-X' at the bottom left and 'Color' at the bottom right. The third row contains 'Blur-radius' at the center. Arrows point from each component to its corresponding value in the CSS code: 'Offset-Y' points to '-2px', 'Offset-X' points to '1px', 'Blur-radius' points to '2px', 'Color' points to '4px', and 'Density/Spread' points to '#61677a'.

16

DROP SHADOW IN CSS



DROP SHADOW – CSS

The diagram illustrates the components of a CSS drop-shadow property. It consists of five rounded rectangular boxes arranged in a circle, each containing a component name. Arrows point from each component name to its corresponding value in the CSS code below.

drop-shadow: -2px 1px 2px #61677a;

- Offset-Y: 1px
- Color: #61677a
- Blur-radius: 2px
- Offset-X: -2px



FILTERS IN CSS



FILTERS - CSS

01

grayscale()

03

blur()

05

brightness()

02

opacity()

04

contrast()

06

invert()



18

LISTS IN CSS



LIST PROPERTIES - CSS

01

List-style-type

02

List-style-image

03

List-style-position

04

List-style-property

19

ANCHOR IN CSS



ANCHOR STATES - CSS

-
- 01 Link (`a:link`)
 - 02 Visited (`a:visited`)
 - 03 Hover (`a:hover`)
 - 04 Active (`a:active`)

20

COMBINATORS IN CSS



COMBINATORS – CSS

01

Descendant Selector
(Space)

03

Adjacent Sibling
Selector (+)

05

Universal Selector (*)

02

Child Selector (>)

04

General Sibling Selector
(~)

21

DISPLAY IN CSS



DISPLAYS - CSS

01

Block-Level Elements

02

Inline Elements

03

Inline-Block Elements



22

POSITIONS IN CSS



POSITIONS - CSS



Relative



Static



Sticky



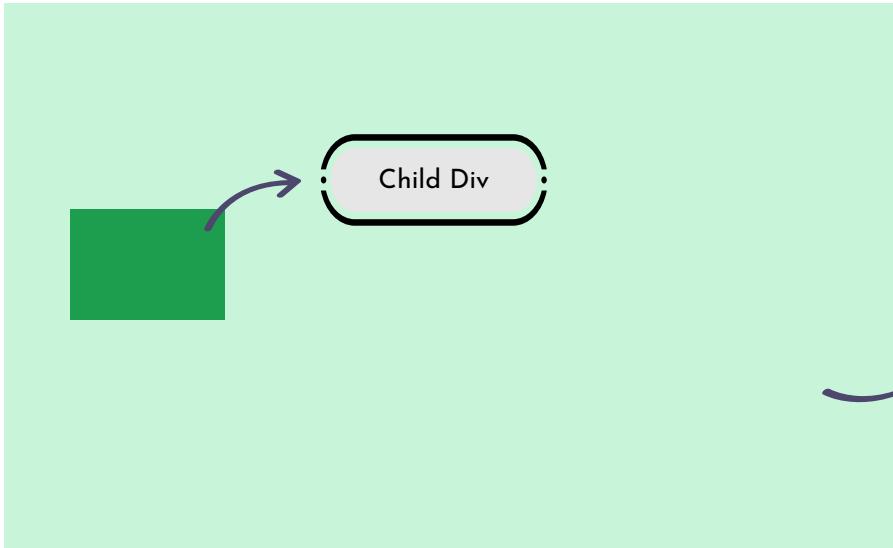
Absolute



Fixed

POSITIONS – Absolute

Top:0, left:0



Top:0, right:0

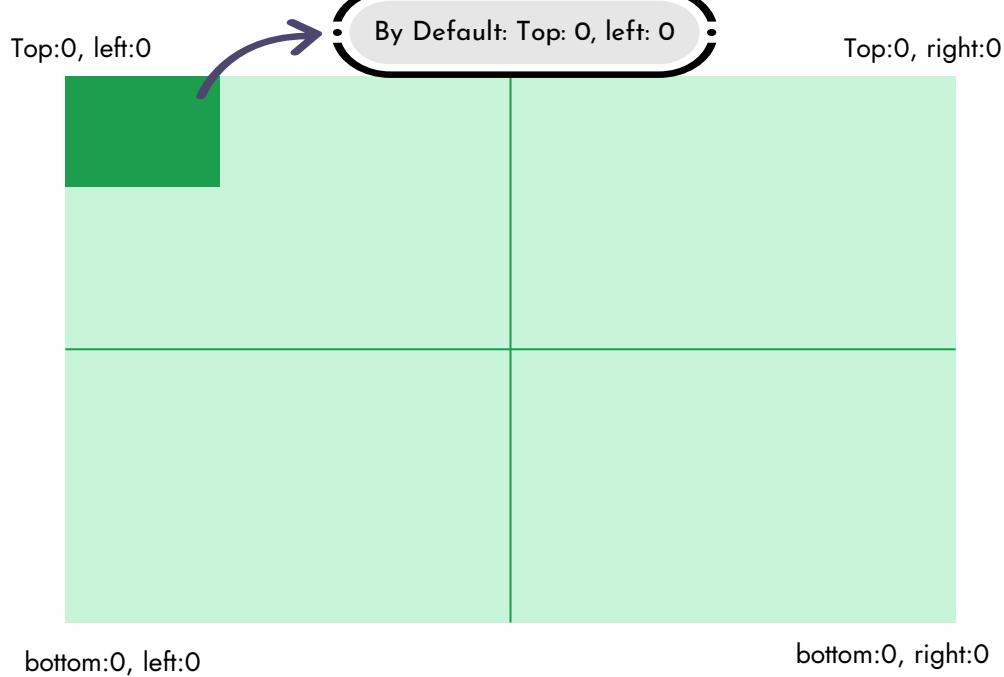
```
.child {  
  position: absolute;  
}
```

Parent Div

bottom:0, left:0

bottom:0, right:0

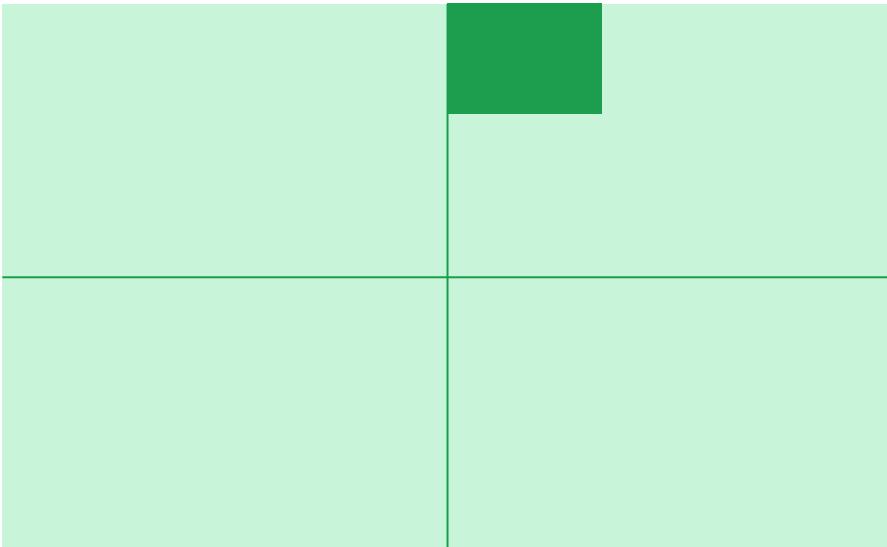
POSITIONS – Absolute



```
.child {  
  position: absolute;  
  left: 50%;  
}
```

POSITIONS – Absolute

Top:0, left:0



bottom:0, left:0

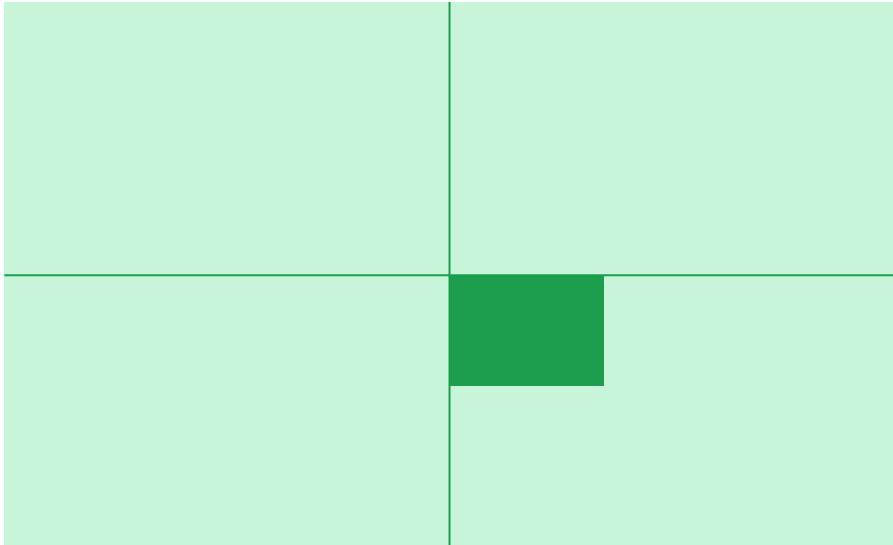
Top:0, right:0

bottom:0, right:0

```
.child {  
  position: absolute;  
  left: 50%;  
  top: 50%;  
}
```

POSITIONS – Absolute

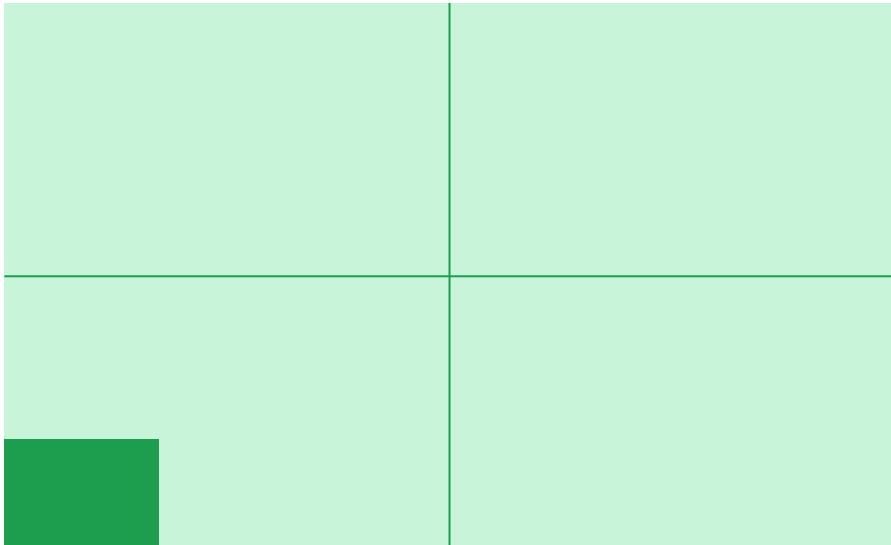
Top:0, left:0



```
.child {  
  position: absolute;  
  left: 0;  
  bottom:0;  
}
```

POSITIONS – Absolute

Top:0, left:0



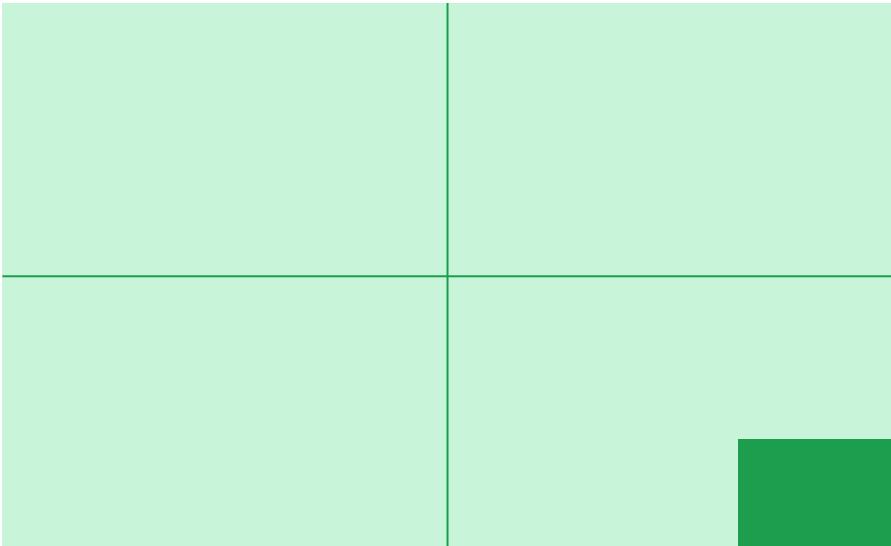
```
.child {  
  position: absolute;  
  right: 0;  
  bottom:0;  
}
```

bottom:0, left:0

bottom:0, right:0

POSITIONS – Absolute

Top:0, left:0



bottom:0, left:0

bottom:0, right:0

```
.child {  
  position: absolute;  
  top: 0;  
  left:0;  
}
```

POSITIONS – Absolute

Top:0, left:0



Top:0, right:0

You can use **px** and any **values**. Also, if you love the animation, Pls subscribe and our Target – 1 Million

bottom:0, left:0

bottom:0, right:0

Z-INDEX IN CSS



25

OVERFLOW IN CSS



OVERFLOW – CSS

01

Overflow: Visible
(Default)

02

Overflow: Hidden

03

Overflow: Scroll

04

Overflow: Auto

26

PSEUDO ELEMENTS IN CSS



Pseudo Elements - CSS

01

::before

03

::first-letter

05

::selection

02

::after

04

::first-line

06

::placeholder

27

PSEUDO CLASS IN CSS



Pseudo Class – CSS

01

:hover

03

:last-child

05

:first-of-type

02

:first-child

04

:nth-child(n)

06

:last-of-type

28

COLUMN LAYOUT IN CSS



COLUMN LAYOUT - CSS

01

Column-count

03

Column-rule

02

Column-gap

04

Column-fill

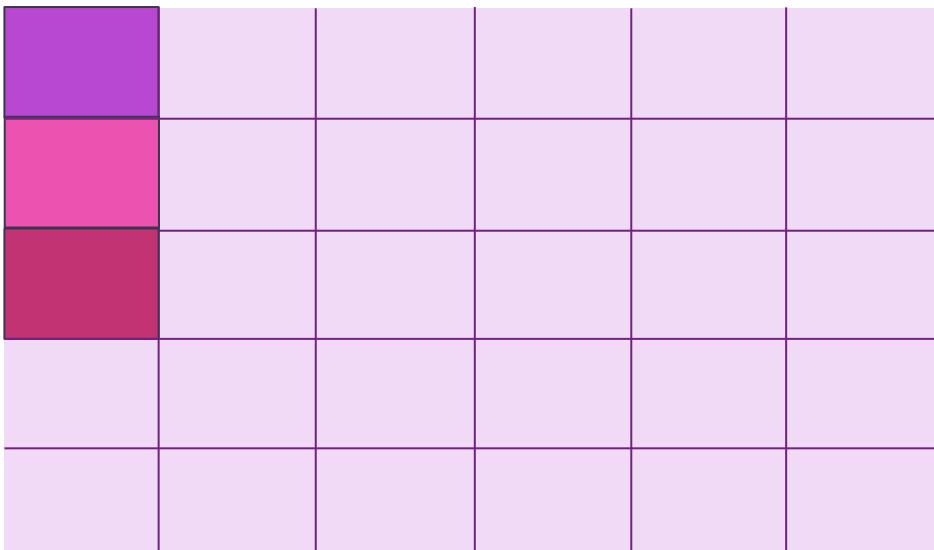


28

FLEXBOX IN CSS

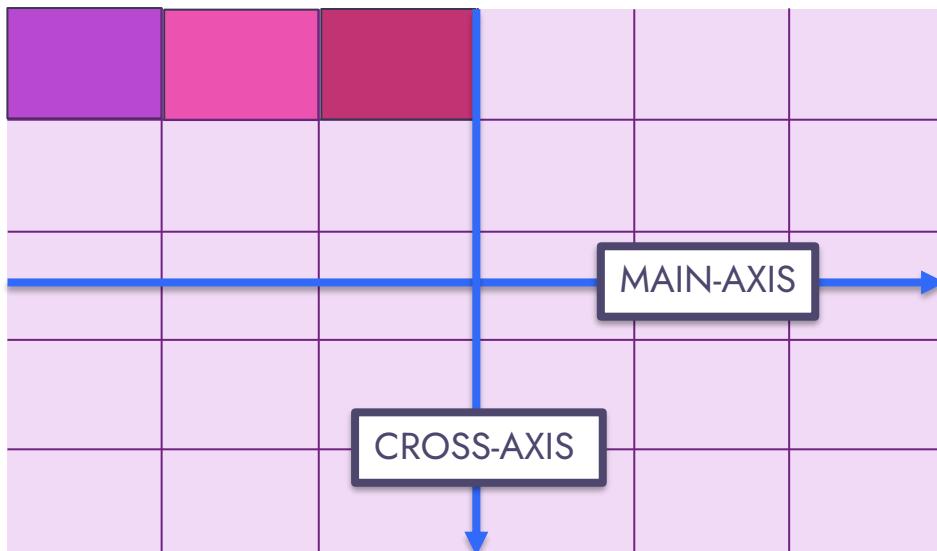


FLEXBOX – CONTAINER



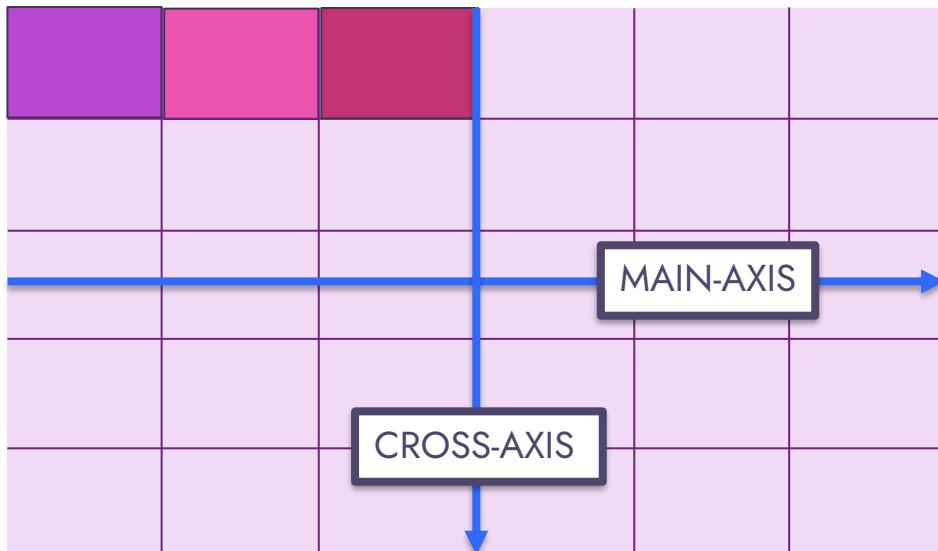
```
.flex-container {  
    display: flex;  
}
```

FLEXBOX – ROW



```
.flex-container {  
  display: flex;  
  flex-direction: row;  
}
```

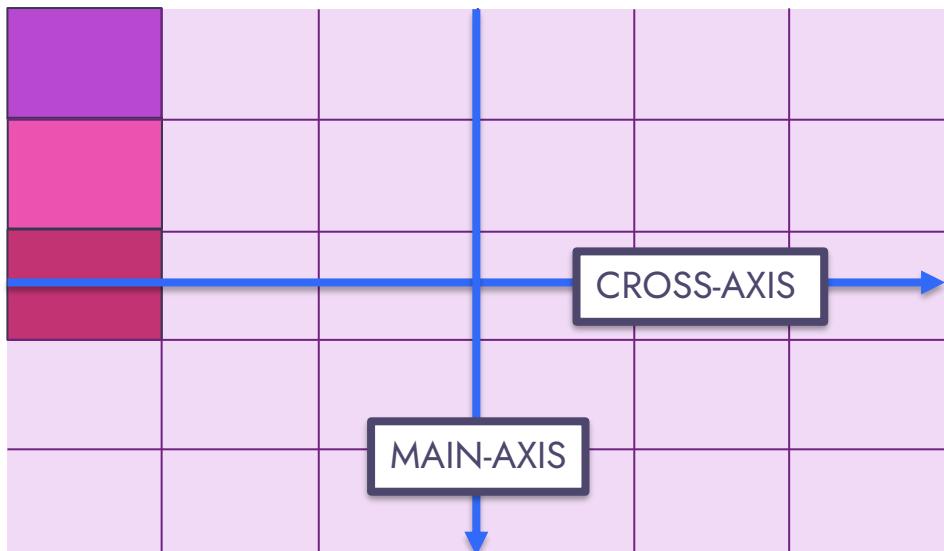
FLEXBOX – COLUMN



```
.flex-container {  
    display: flex;  
    flex-direction: row;  
    flex-direction: Column;  
}
```

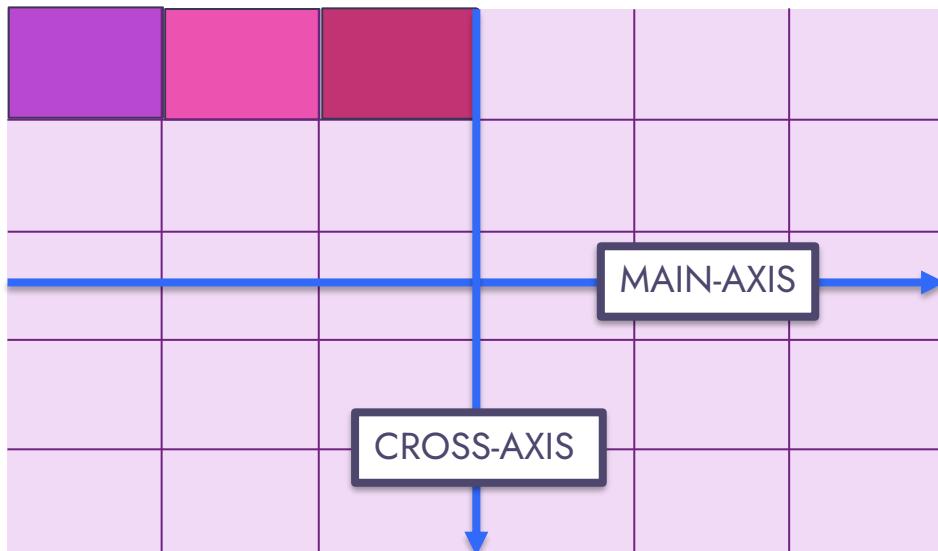


FLEXBOX – CONTAINER



```
.flex-container {  
    display: flex;  
    flex-direction: Column;  
    flex-direction: row;  
    justify-content: start  
}
```

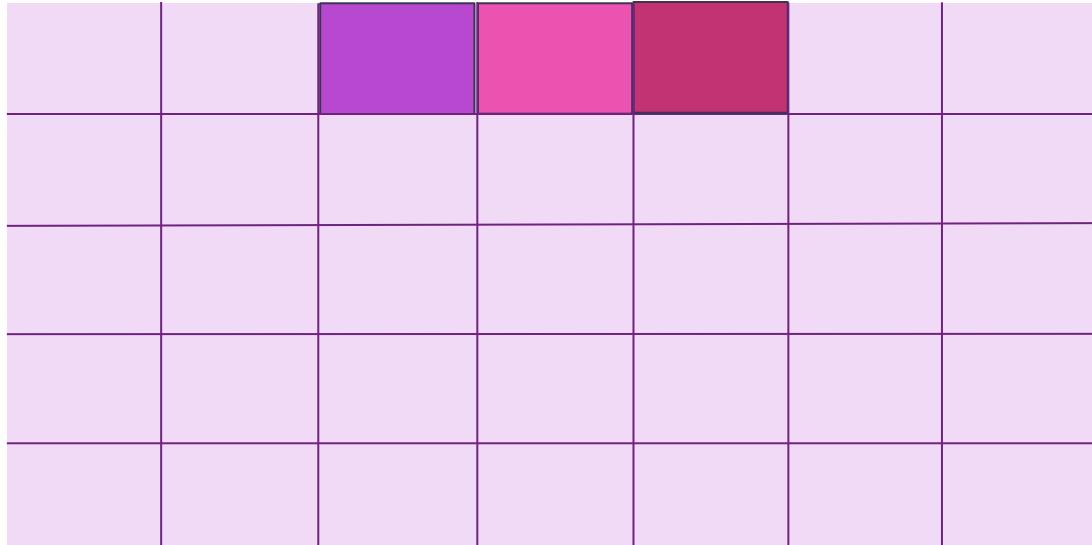
FLEXBOX - CSS



```
.flex-container {  
    display: flex;  
    flex-direction: row;  
    justify-content:center  
}
```

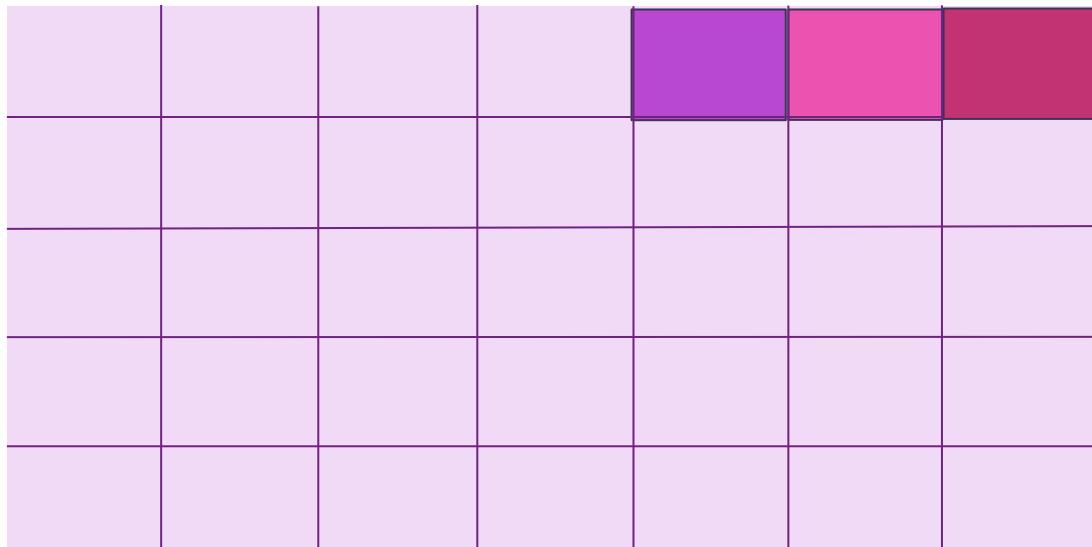


FLEXBOX – CSS



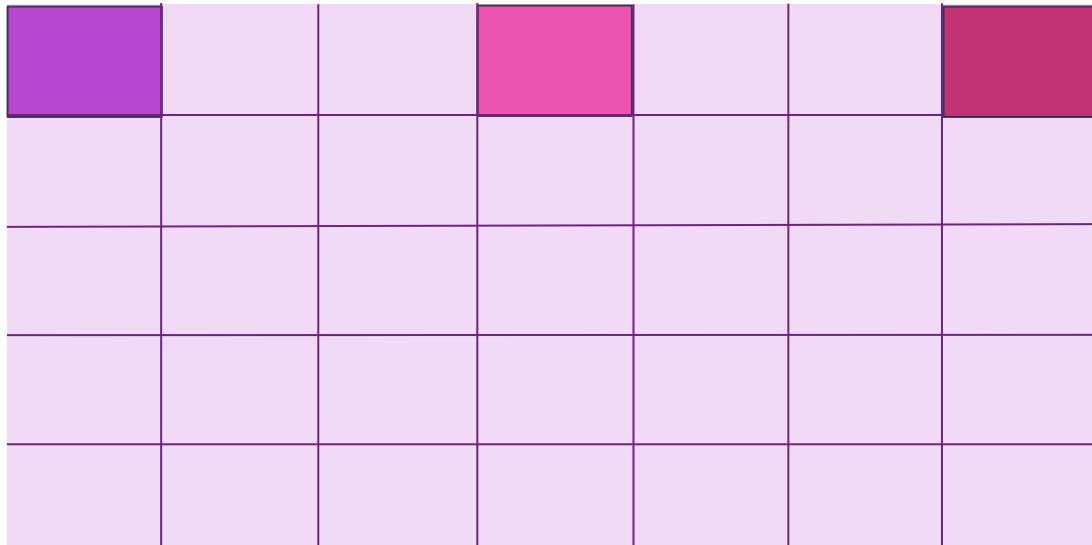
```
.flex-container {  
    display: flex;  
    flex-direction: row;  
    justify-content:flex-end  
}
```

FLEXBOX – CSS



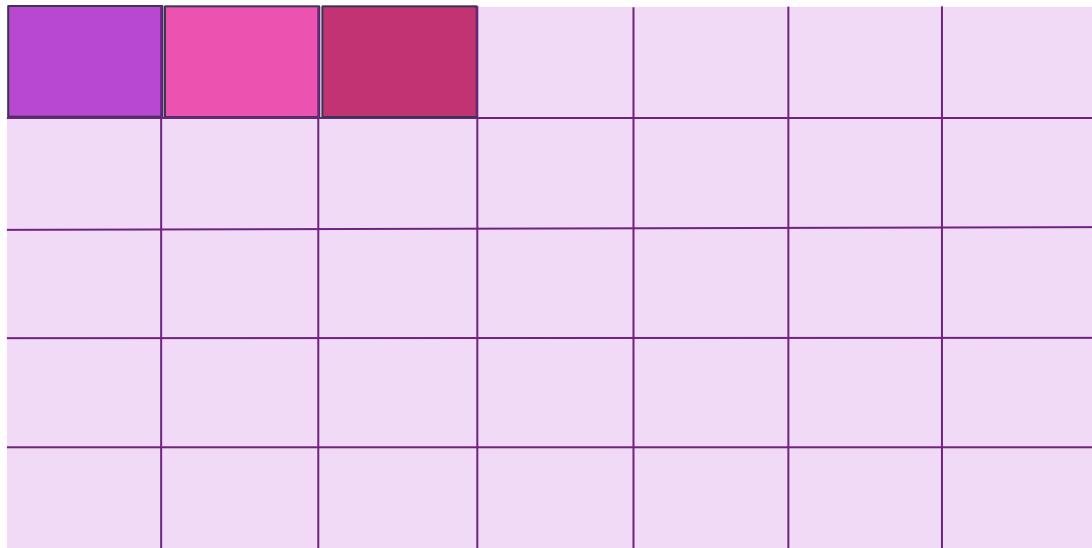
```
.flex-container {  
    display: flex;  
    flex-direction: row;  
    justify-content: space-between  
}
```

FLEXBOX – CSS



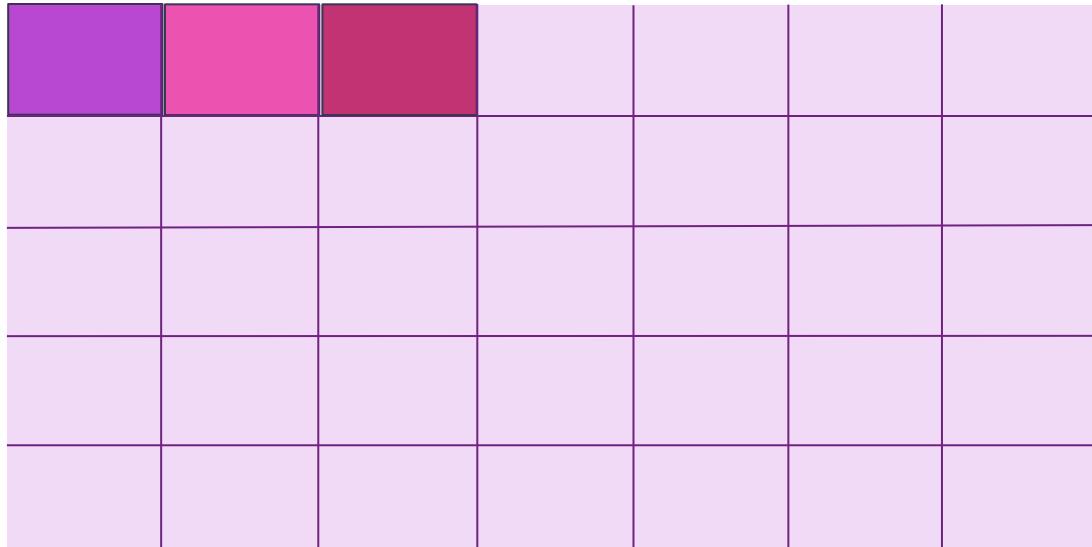
```
.flex-container {  
    display: flex;  
    flex-direction: row;  
    justify-content: start;  
}
```

FLEXBOX – CSS



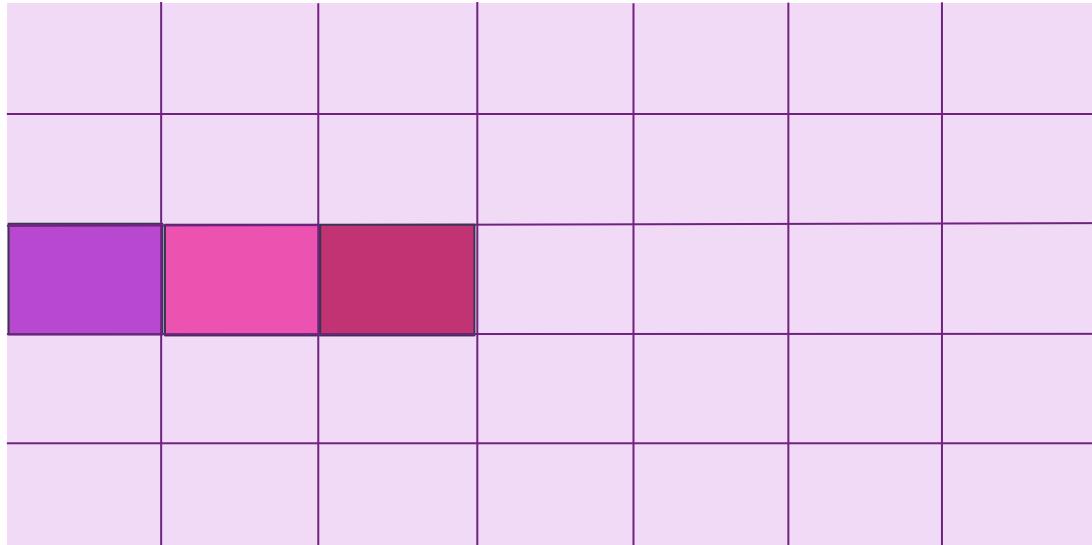
```
.flex-container {  
    display: flex;  
    flex-direction: row;  
    justify-content: start;  
    align-items: start;  
}
```

FLEXBOX – CSS



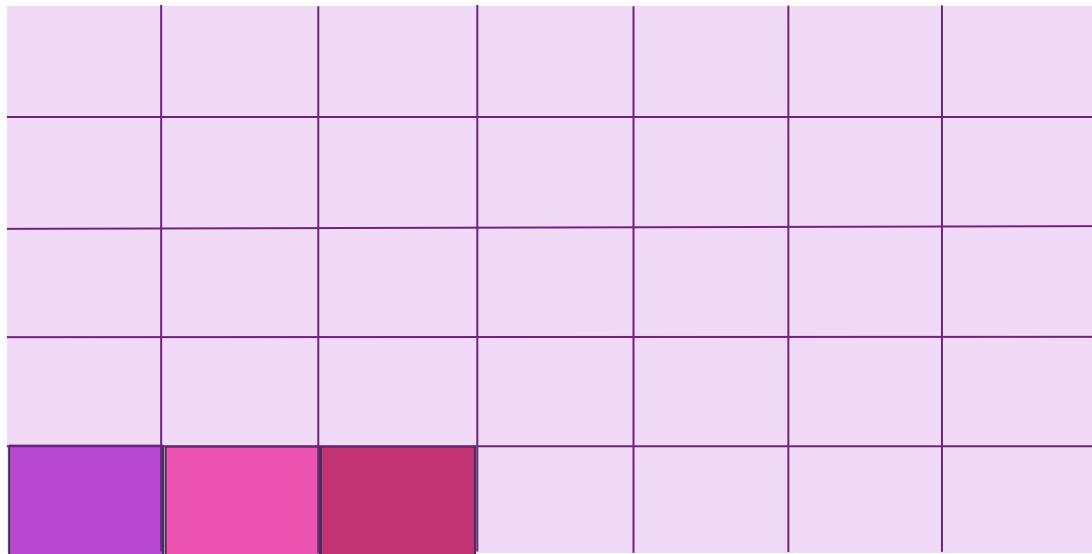
```
.flex-container {  
    display: flex;  
    flex-direction: row;  
    justify-content: start;  
    align-items: center;  
}
```

FLEXBOX – COLUMN



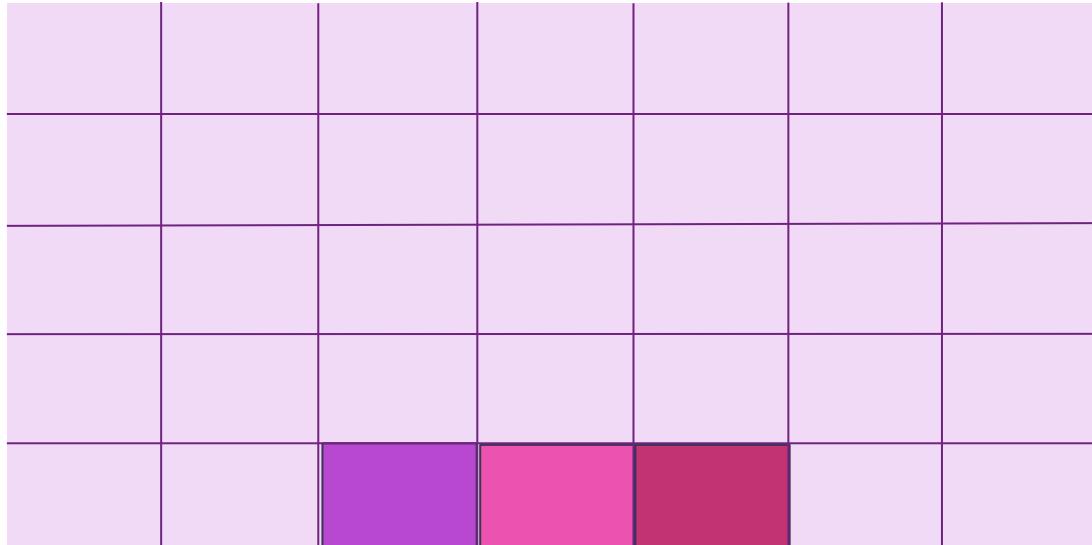
```
.flex-container {  
    display: flex;  
    flex-direction: row;  
    justify-content: start;  
    align-items: end  
}
```

FLEXBOX – COLUMN



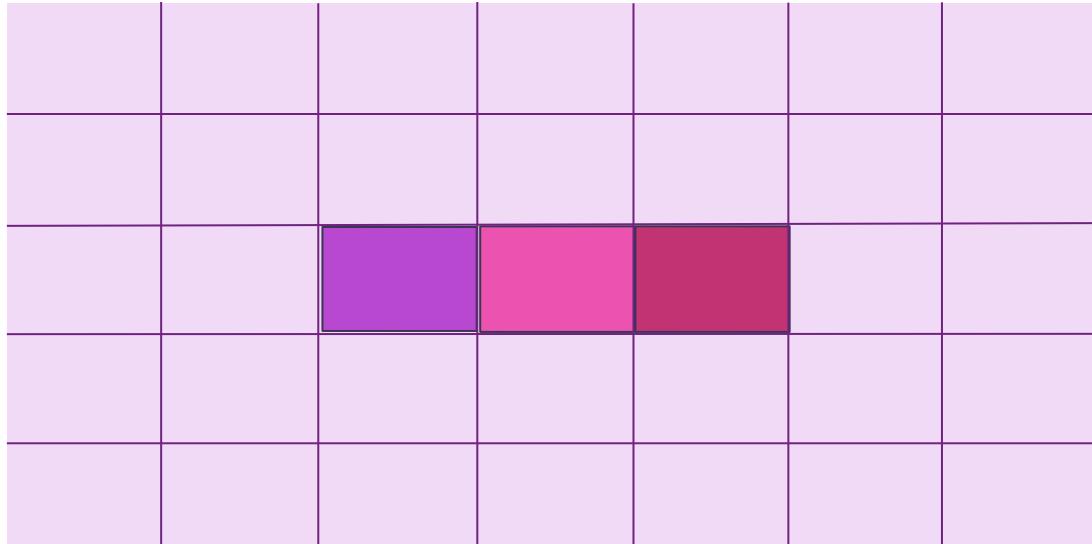
```
.flex-container {  
    display: flex;  
    flex-direction: row;  
    justify-content:center  
    align-items:end;  
}
```

FLEXBOX – COLUMN



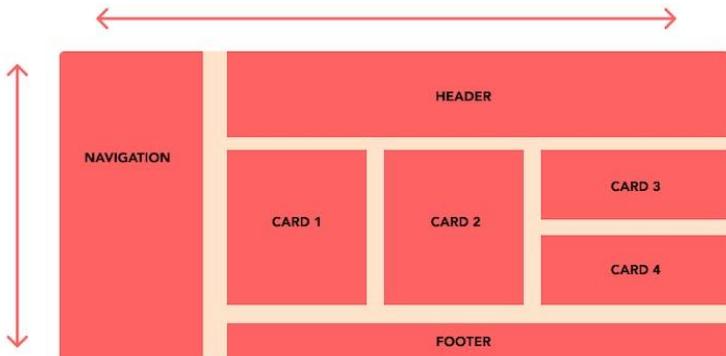
```
.flex-container {  
    display: flex;  
    flex-direction: row;  
    justify-content:center  
    align-items:center  
}
```

FLEXBOX – COLUMN



```
.flex-container {  
    display: flex;  
    flex-direction: row;  
    justify-content:center  
    align-items:center  
}
```

GRID IN CSS



29

TRANSITIONS IN CSS



TRANSITIONS – CSS

01

transition-property

02

transition-duration

03

transition-timing-function

04

transition-delay

30

TRANSFORM IN CSS



TRANSFORM - CSS

01

transform: translate()

03

transform: rotate()

02

transform: scale()

04

transform: skew()



ANIMATION IN CSS



ANIMATION – CSS

01

animation-name

03

animation-timing-function

05

animation-iteration-count

02

animation-duration

04

animation-delay

06

animation-direction

32

CSS VARIABLES IN CSS



33

SPECIFICITY IN CSS



SELECTORS IN CSS

01

* (universal selector)

03

Class or pseudo-class

05

Inline style

02

Element / pseudo-element

04

id

06

!important

The Level of **SPECIFICITY** – CSS

0.0.0.0.0



The universal selector has **0 specificity** (No priority)

The Level of SPECIFICITY - Element

0.0.0.0.1

Element / pseudo-elements

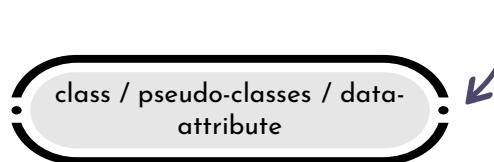
Includes only element selectors, such as **h1, img, p::before**. For each ID in a matching selector, add **0-0-0-0-1** to the weight value.



The Level of SPECIFICITY - Class

0.0.0.1.0

class / pseudo-classes / data-attribute

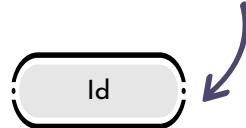


Includes class selectors, such as `.myClass`, `[type="radio"]`, `:hover`. For each class, attribute selector, or pseudo-class in a matching selector, add **0-0-0-1-0** to the weight value.



The Level of SPECIFICITY – ID

0.0.1.0.0



Includes only ID selectors, such as **#btn**. For each ID in a matching selector, add **0-0-1-0-0** to the weight value.



The Level of SPECIFICITY - Inline

0.1.0.0.0



For each inline style ex

```
<div class="container" style="background: #000;">  
add 0-1-0-0-0 to the weight value.
```



The Level of SPECIFICITY - !Important

1.0.0.0.0

!important



For each important keyword ex `h1 {color: red !important}` add **1-0-0-0-0** to the weight value.



34



NEW CSS FEATURES



New Features - CSS



01

:is, :has, :not, :where

03

Container queries

05

Aspect ratio

02

Media query range
syntax

04

Accent-color

06

Scroll snap





New Features - CSS



Individual Transform Properties



Gap Property for Flexbox



CSS Writing Mode



CSS Nesting



CSS Logical Properties
(inline and block)



:focus-visible



35

BONUS IN CSS





BONUS - CSS

01

FLOATS

03

CSS AOS PLUGIN

05

SCROLL BAR STYLING

02

CSS CLIP PATH

04

CSS SHAPE DIVIDER

06

WEBSITE REFERENCES



CLIP PATH – CSS



```
img{  
  clip-path: polygon(50% 0%, 0% 100%, 100% 100%);  
}
```

Left: 0%
Top: 100%

Left: 50%
Top: 0%

Left: 100%
Top: 100%

36

MINI PROJECTS IN CSS



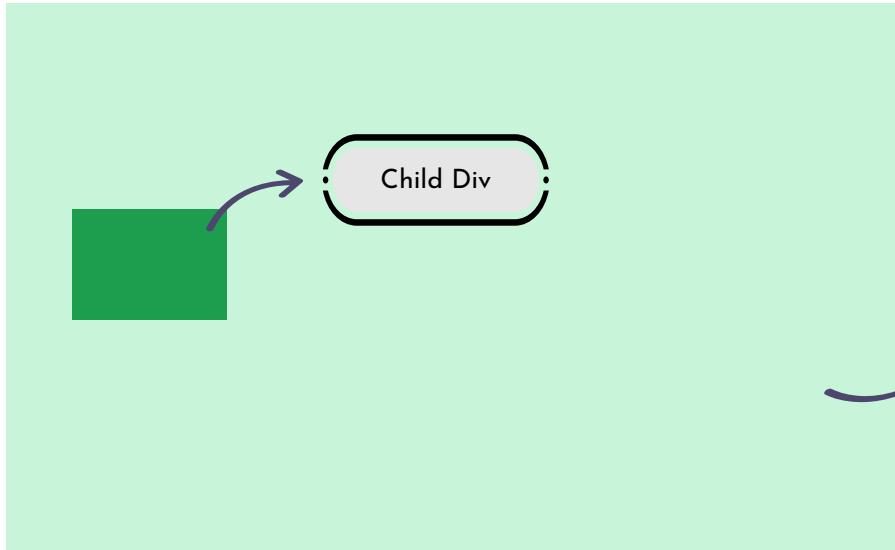
HOW TO CENTER A DIV INSIDE A DIV



HOW TO CENTER A DIV INSIDE DIV IN CSS

POSITIONS – Absolute

Top:0, left:0



Top:0, right:0

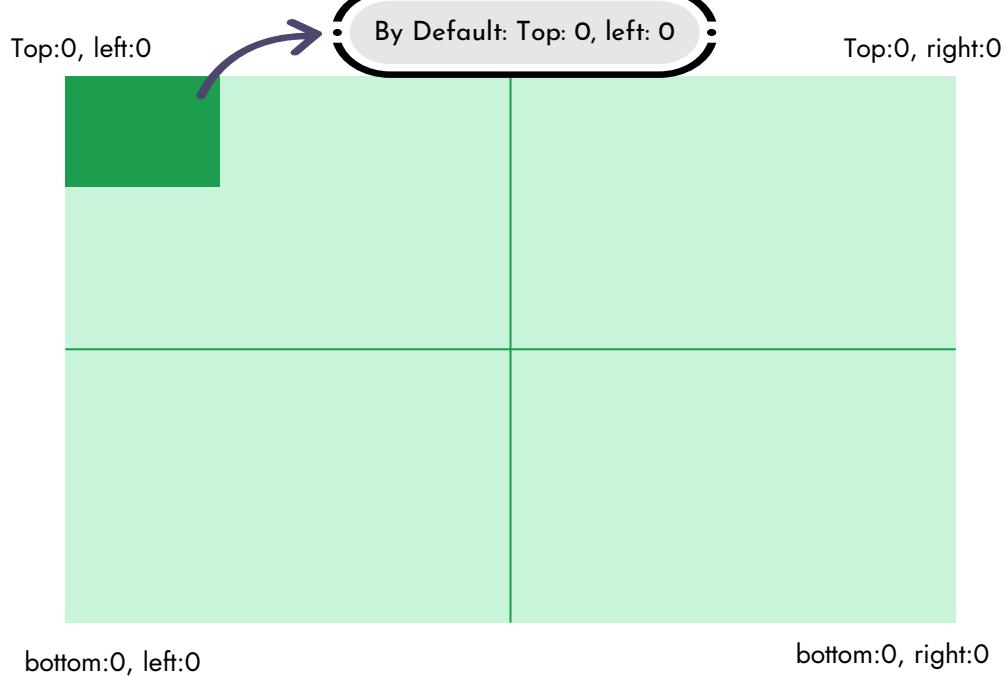
```
.child {  
  position: absolute;  
}
```

Parent Div

bottom:0, left:0

bottom:0, right:0

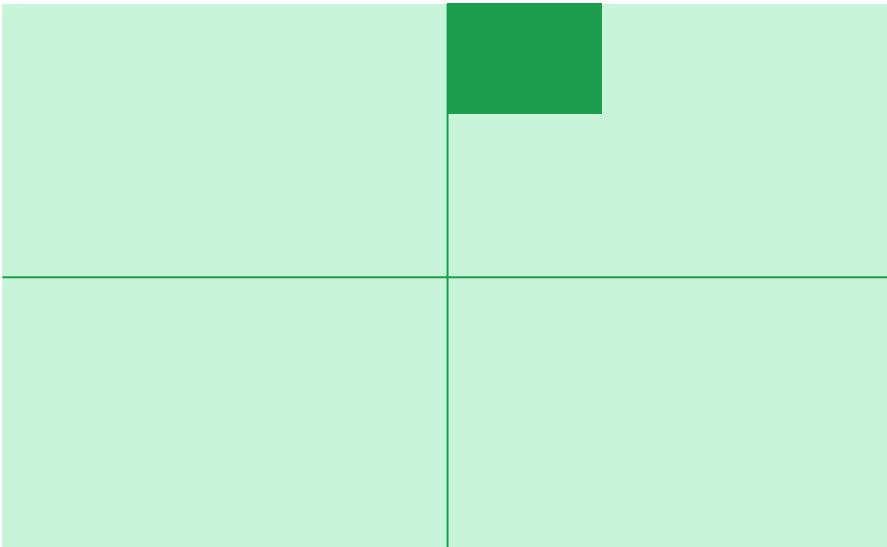
POSITIONS – Absolute



```
.child {  
  position: absolute;  
  left: 50%;  
}
```

POSITIONS – Absolute

Top:0, left:0



bottom:0, left:0

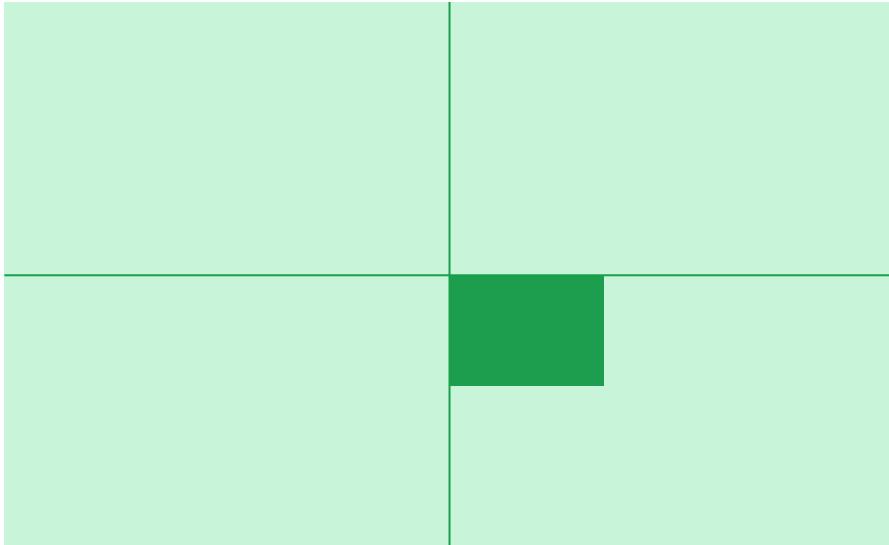
Top:0, right:0

bottom:0, right:0

```
.child {  
  position: absolute;  
  left: 50%;  
  top: 50%;  
}
```

POSITIONS – Absolute

Top:0, left:0



bottom:0, left:0

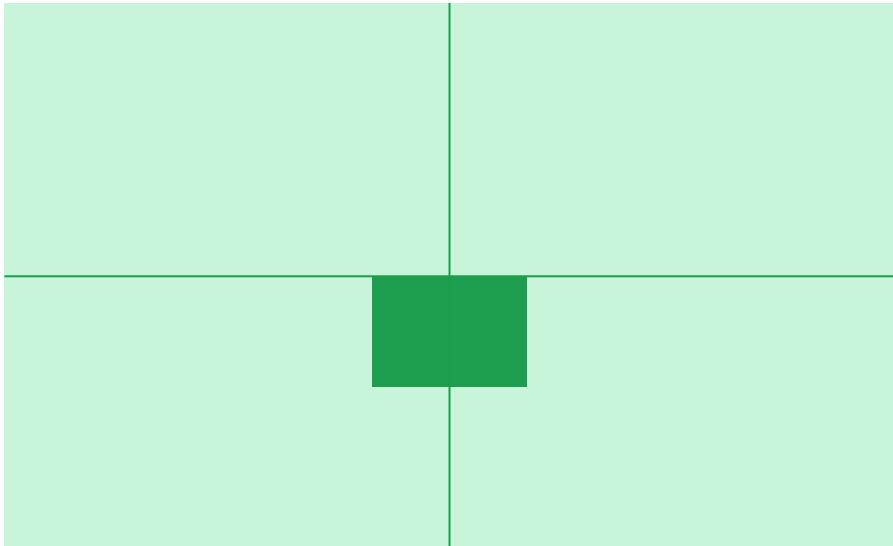
Top:0, right:0

```
.child {  
  position: absolute;  
  left: 50%;  
  top: 50%;  
  transform: translate(-50%)  
}
```

bottom:0, right:0

POSITIONS – Absolute

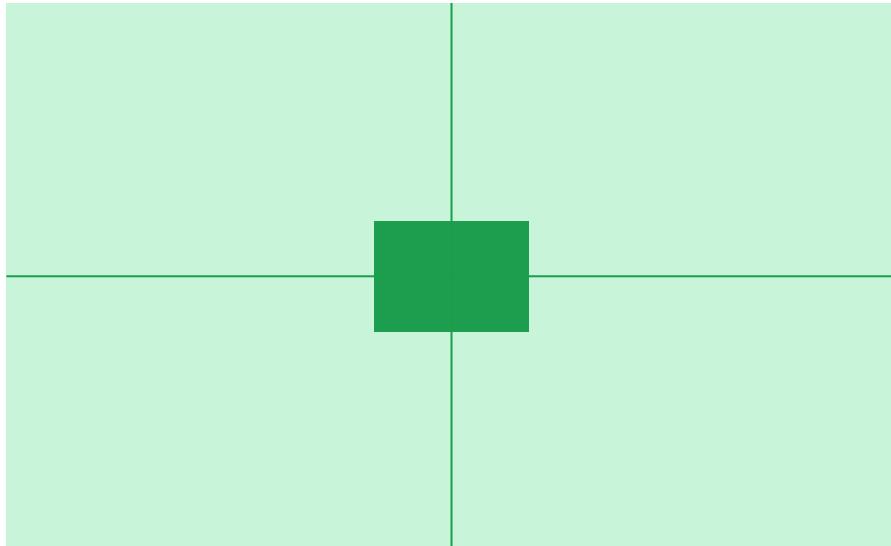
Top:0, left:0



```
.child {  
  position: absolute;  
  left: 50%;  
  top:50%;  
  transform:translate(-50%,-50%)  
}
```

POSITIONS – Absolute

Top:0, left:0



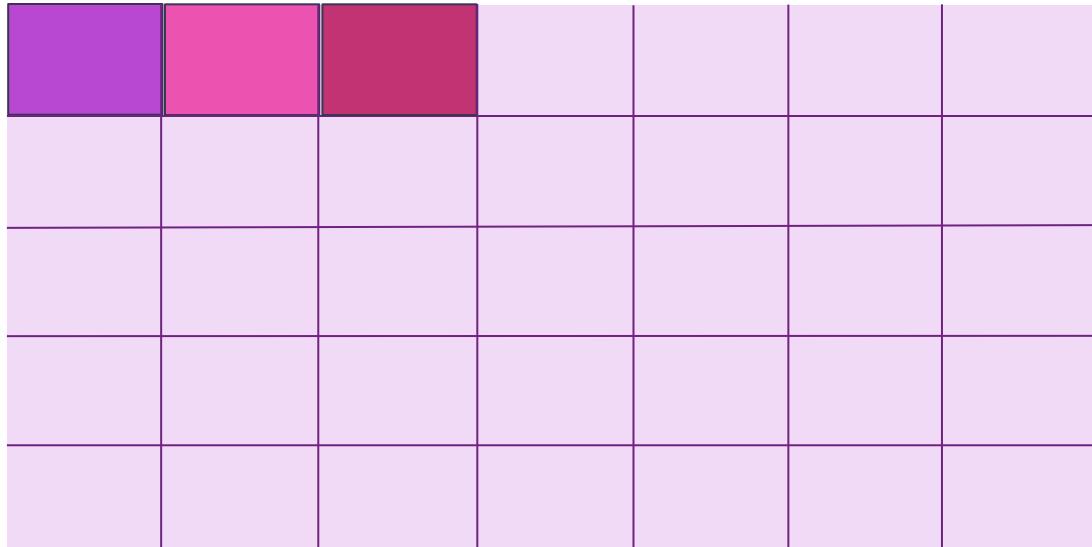
bottom:0, left:0

Top:0, right:0

bottom:0, right:0

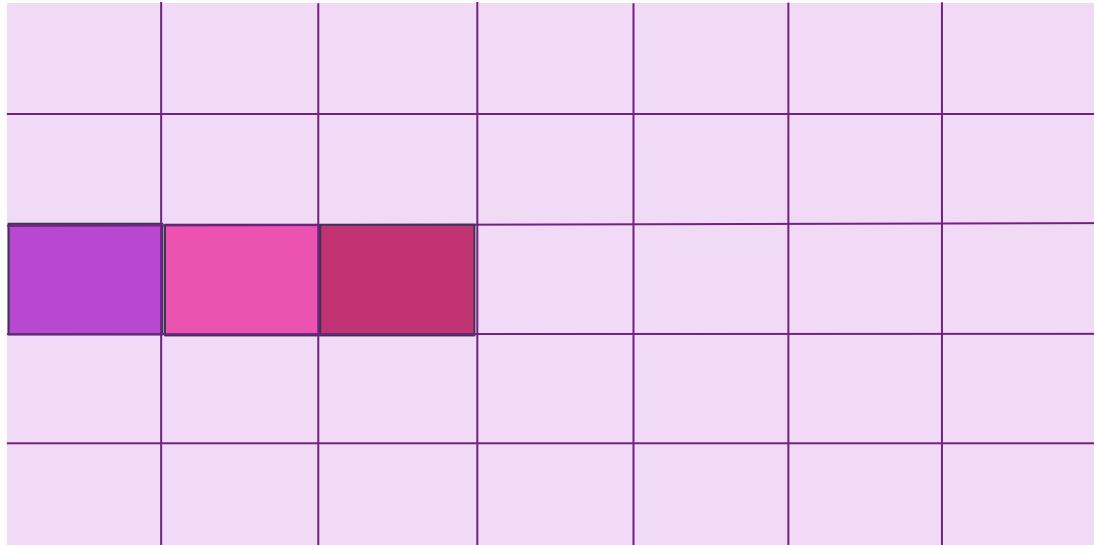
```
.child {  
  position: absolute;  
  left: 50%;  
  bottom:50%;  
  transform:translate(-50%,-50%)  
}
```

FLEXBOX – CSS



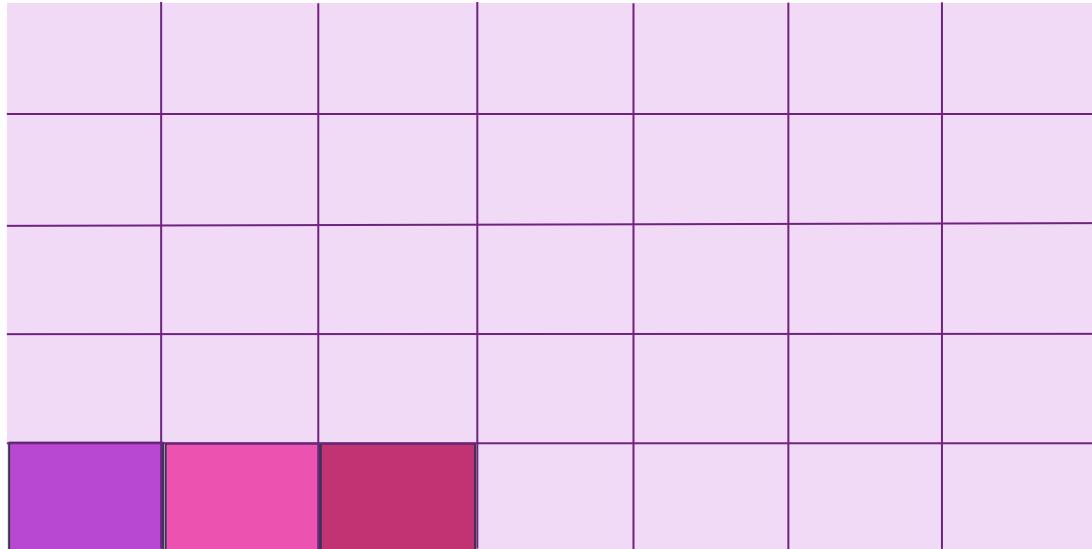
```
.flex-container {  
    display: flex;  
    flex-direction: row;  
    justify-content: start;  
    align-items: center;  
}
```

FLEXBOX – COLUMN



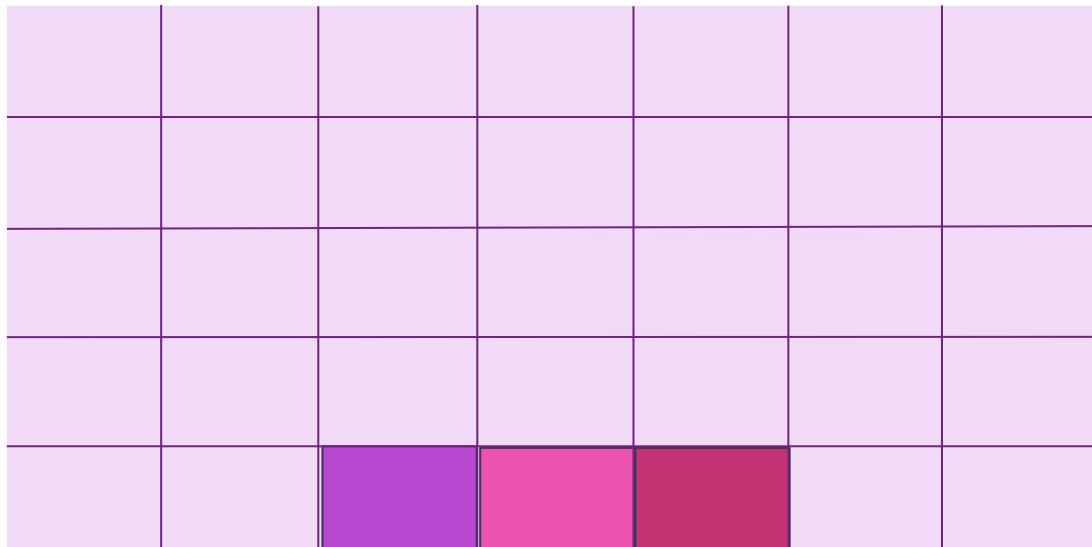
```
.flex-container {  
    display: flex;  
    flex-direction: row;  
    justify-content: start;  
    align-items: end  
}
```

FLEXBOX – COLUMN



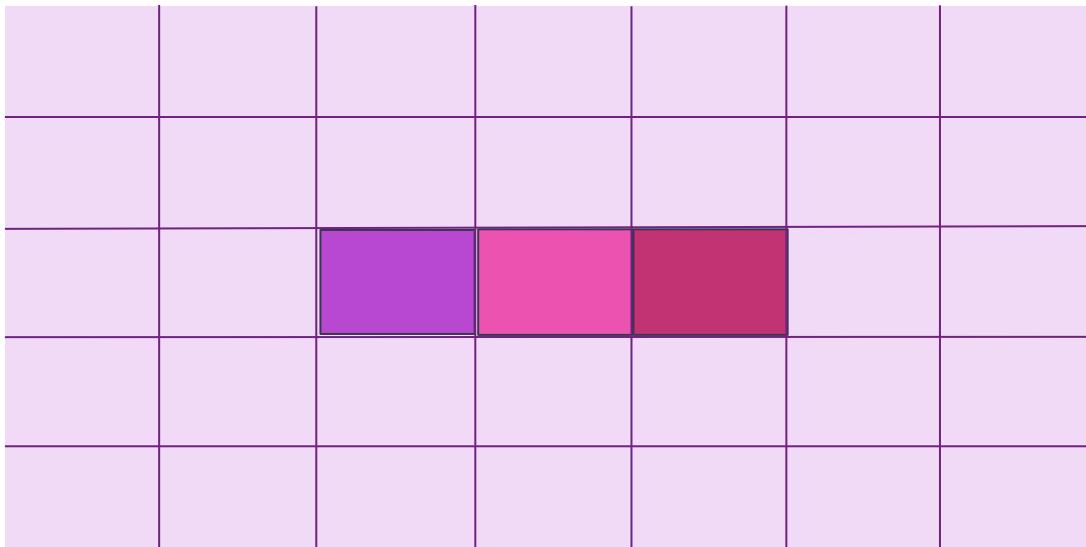
```
.flex-container {  
    display: flex;  
    flex-direction: row;  
    justify-content:center  
    align-items:end;  
}
```

FLEXBOX – COLUMN



```
.flex-container {  
    display: flex;  
    flex-direction: row;  
    justify-content:center  
    align-items:center  
}
```

FLEXBOX - COLUMN



```
.flex-container {  
    display: flex;  
    flex-direction: row;  
    justify-content:center;  
    align-items:center;  
}
```

Please Like & Share the
World's Best CSS Course
Ever ❤

Container Queries



CSS Architecture & Best Practices



VENDOR PREFIXES

Browser Compatibility in CSS



VENDOR PREFIXES - CSS

01

The `-webkit-` prefixed property will work in WebKit-based browsers such as [Chrome](#) and [Safari](#)

02

The `-moz-` prefixed property will work in [Firefox](#)

03

The `-ms-` prefixed property will work in [Internet Explorer](#)

04

The `-o-` prefixed property will work in [Opera](#)



Browser Compatibility in CSS



Vendor prefixes

Returning to the gradient example we saw earlier, you may have noticed that something wasn't quite right there. To get the gradient rendering across all modern browsers, you need to include multiple versions of the declaration, like this:

```
nav {  
    background-color: red;  
    background-image: url(gradient-slice.png);  
    background-image: -webkit-linear-gradient(top right, #A60000, #FFFFFF);  
    background-image: -moz-linear-gradient(top right, #A60000, #FFFFFF);  
    background-image: -ms-linear-gradient(top right, #A60000, #FFFFFF);  
    background-image: -o-linear-gradient(top right, #A60000, #FFFFFF);  
    background-image: linear-gradient(top right, #A60000, #FFFFFF);  
}
```

This is because the [CSS Image Values and Replaced Content Module Level 3](#) — the spec that CSS gradients are specified in, is not finalised. Unfinished CSS features are implemented in browsers in an experimental capacity, with a vendor-specific prefix so that different browser's implementations can be tested without impacting on other browser's implementations. In the above example:

- The `-webkit-` prefixed property will work in WebKit-based browsers such as Chrome and Safari
- The `-moz-` prefixed property will work in Firefox
- The `-ms-` prefixed property will work in Internet Explorer
- The `-o-` prefixed property will work in Opera

Official W3C policy states that you shouldn't really use experimental properties in production code, but people do, as they want to make sites look cool and keep on the cutting edge. I don't see a problem with this, but if you want to use vendor prefixed properties in your code, you really should use include all the different prefixes so that as many browsers as possible can use all the features you are implementing.

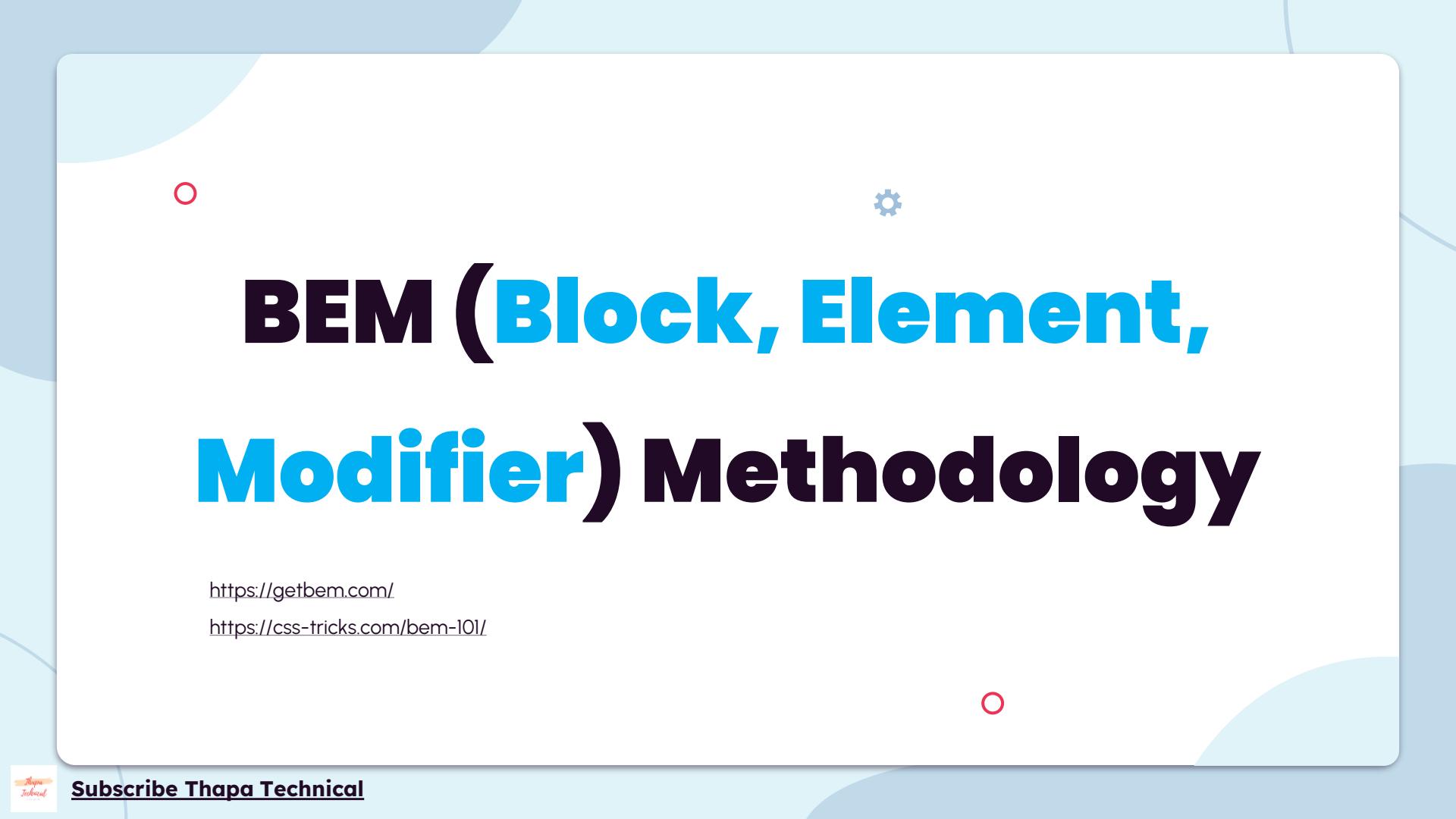
Notice that an unprefixed version of the property has also been included, at the bottom of the list: this is so that when the CSS feature is finalised, and the prefixes are removed, the code will still work — this is good future proofing practice.

https://www.w3.org/community/webed/wiki/Optimizing_content_for_different_browsers:_the_RIGHT_way

BEM



[Subscribe Thapa Technical](#)



BEM (Block, Element, Modifier) Methodology

<https://getbem.com/>

<https://css-tricks.com/bem-101/>

SMACSS



[Subscribe Thapa Technical](#)



SMACSS

(Scalable and Modular Architecture for CSS)

[Home - Scalable and Modular Architecture for CSS \(smacss.com\)](http://smacss.com)

40

FINAL PROJECTS IN CSS



SEO IN CSS



SEO - CSS

01

Include a descriptive and keyword-rich <title> tag within the <head> section of your HTML document.

02

Meta Description Tag

03

Proper Heading Tags

04

Image Optimization



TESTING IN CSS



CSS Performance and Optimization

01

Minification and Compression

02

Critical CSS: "Above-the-fold" content

03

Reducing Render Blocking



Website Testing - CSS

01

Lighthouse testing

02

<https://pagespeed.web.dev/>

03

<https://gtmetrix.com/>



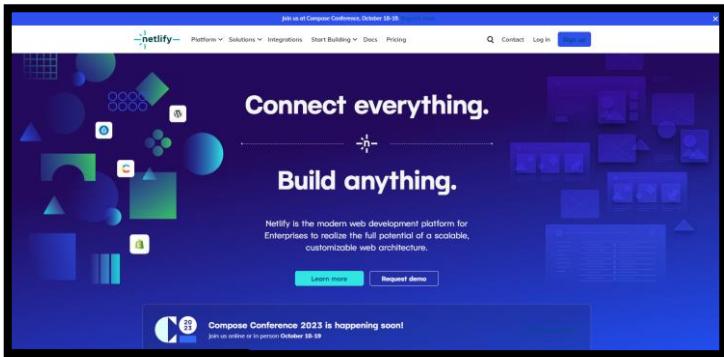
HOSTING WEBSITE LIVE



HOSTING WEBSITE LIVE

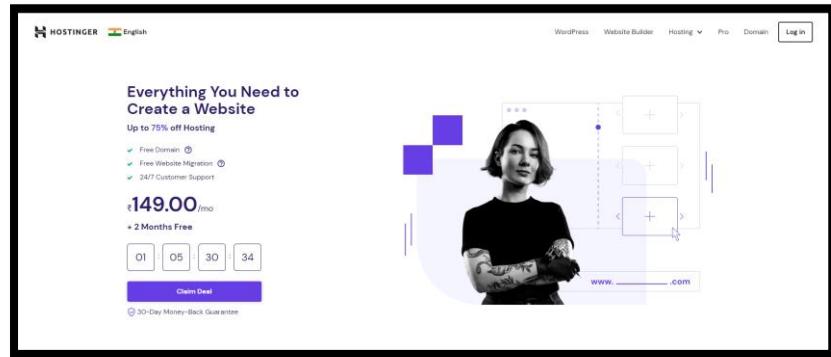


NETLIFY



<https://www.netlify.com/>

HOSTINGER



<http://www.hostinger.com/>

Thanks!

**Subscribe Thapa Technical if you liked
the presentation.**

