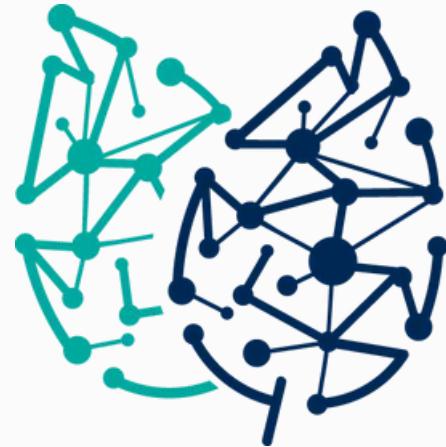


# CSS level one





# Training Outcomes



- 1 CSS Basics:** Write and apply basic styles for web pages, including colors, fonts, and layout.
- 2 Box Model:** Understand the CSS box model and how to manipulate margins, padding, borders, and content areas for effective layout design.
- 3 Layout Techniques:** Implement modern layout techniques, such as Flexbox, CSS Grid, and responsive media queries for building adaptive and fluid layouts.
- 4 Styling Components:** Apply styles to HTML elements using different selectors (class, ID, descendant, and pseudo-classes).
- 5 Animations and Transitions:** Create smooth animations and transitions to enhance user interaction and visual feedback.
- 6 Responsive Design:** Build websites that adapt to different screen sizes using flexible layouts, breakpoints, and media queries.

# KWL Strategy



k

What I know

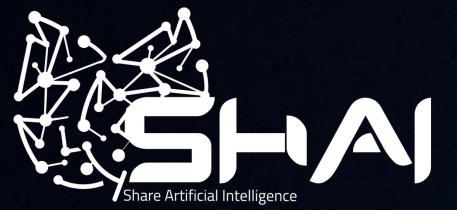
w

What I want to learn

l

What I learned





# Build the Base





Training



# JOURNE

Introduction

1

Basic  
Concepts of  
CSS

2

Types of CSS  
Selectors

3

Rules and ,  
Properties ,  
and Values

4

How to  
specify color

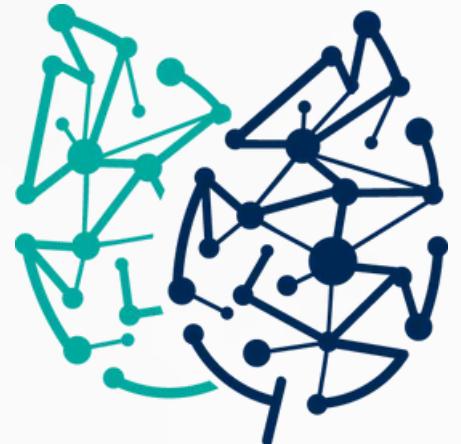




What is  
CSS?

## Introduction to CSS

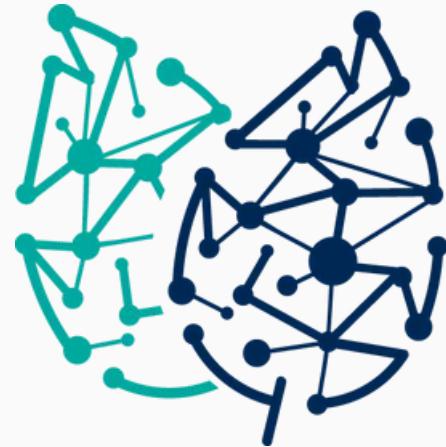




What is  
**CSS?**

**CSS Role in Web  
Development**

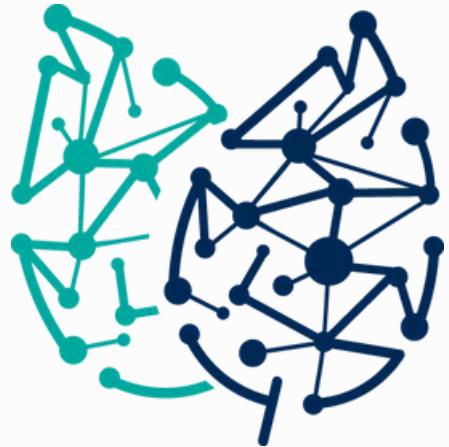




What is  
**CSS**?

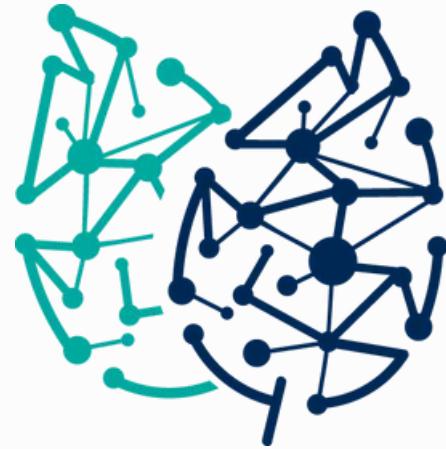
Introduction to CSS

CSS  
Cascading Style Sheets



# How CSS work?

HTML <b>(CONTENT)</b>	CSS (DESIGN)
<b>Structure and content of web pages</b>	<b>Style and appearance of web pages</b>
<b>Focuses on the layout, text, images, and links</b>	<b>Controls color, fonts, spacing, and layout</b>



# What is CSS?

## importance of CSS

### 1. Visual Appeal

**Creates aesthetically pleasing and consistent design across multiple pages**



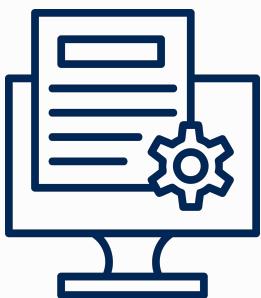
### 2. User Experience

**Allows for responsive design, making web pages accessible on various devices**



### 3. Content Management

**Enables easy updates and maintenance without altering the HTML structure**





Training



# JOURNE

Introduction



**Basic Concepts  
of CSS**



Types of CSS  
Selectors



Rules and ,  
Properties ,  
and Values



How to  
specify color





# What is Basic Concepts of CSS?



**Basic  
Concepts of  
CSS**

**Selectors**

**Properties and Values**

**Cascade**

**Inheritance**

**Specificity**



Training



# JOURNE

Introduction

1

Basic  
Concepts of  
CSS

Y

2

**Types of CSS  
Selectors**

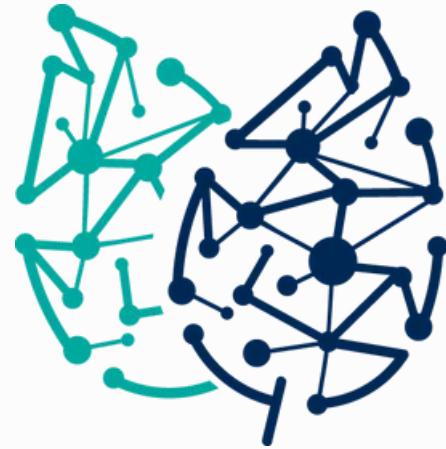
3

Rules and ,  
Properties ,  
and Values

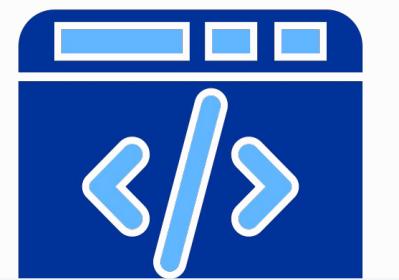
4

How to  
specify color





# Basic Syntax and **STRUCTURE**

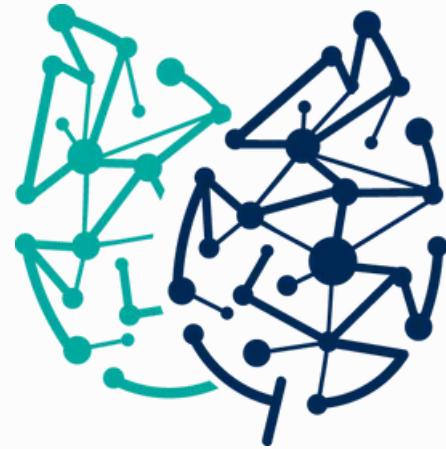


selector

```
p {  
  font-size: 16px;  
}
```

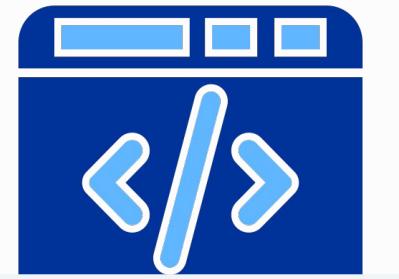
property

property value

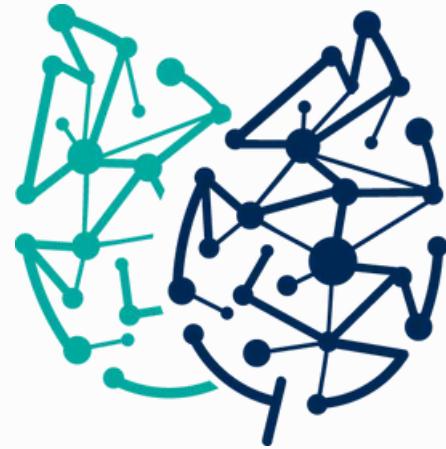


# Basic Syntax and **STRUCTURE**

**Universal Selector (\*)**

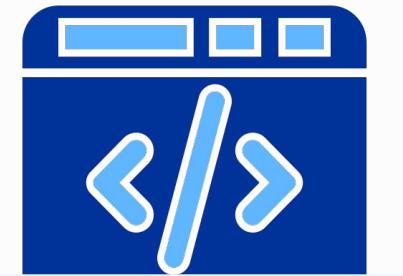


```
{  
color: red;  
}
```

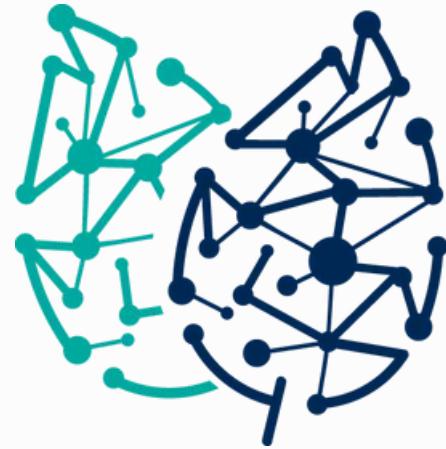


# Basic Syntax and STRUCTURE

Type (Element) Selector

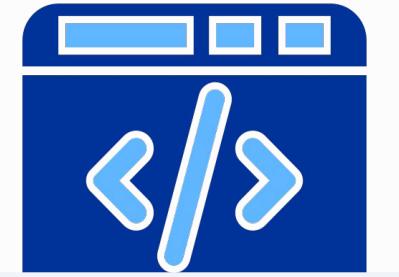


```
p {  
  font-size: 16px;  
}
```

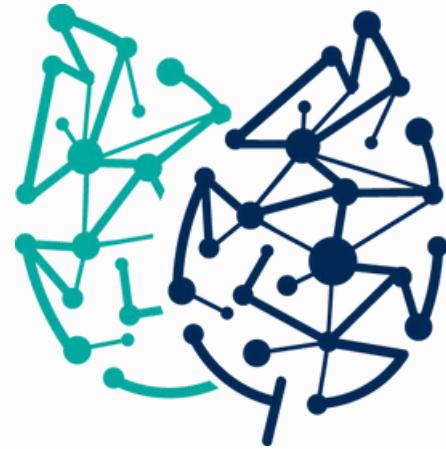


# Basic Syntax and **STRUCTURE**

Class Selector (.)

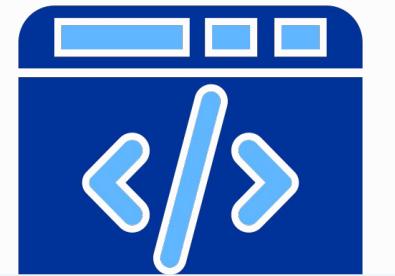


```
.example {  
background-color: yellow;  
}
```

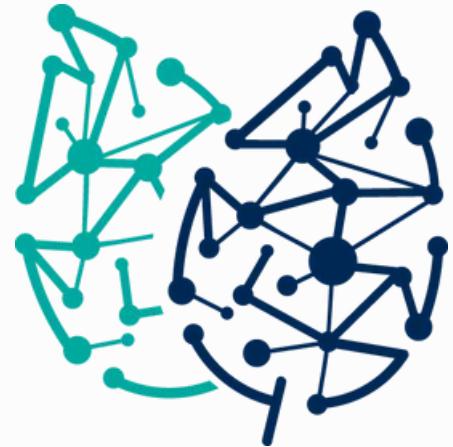


# Basic Syntax and **STRUCTURE**

**ID Selector (#)**

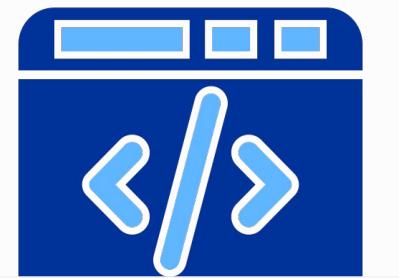


```
#header {  
  font-weight: bold;  
}
```

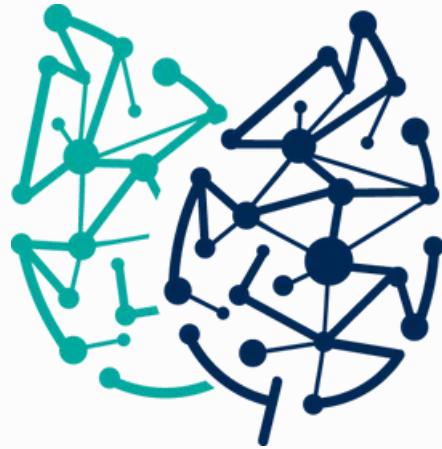


# Basic Syntax and **STRUCTURE**

Attribute Selector

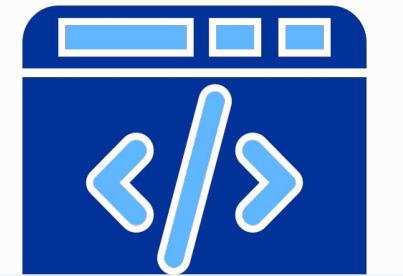


```
input[type="text"] {  
  border: 1px solid blue;  
}
```

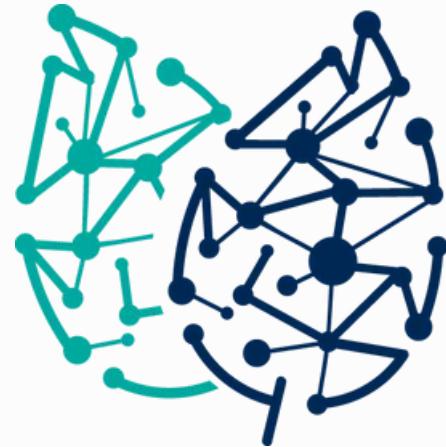


# Basic Syntax and STRUCTURE

Type (Element) Selector

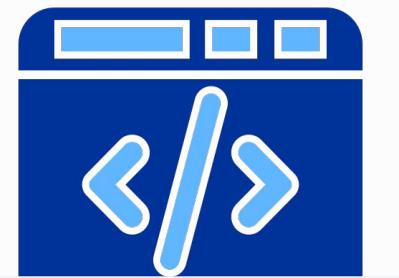


```
p {  
  font-size: 16px;  
}
```

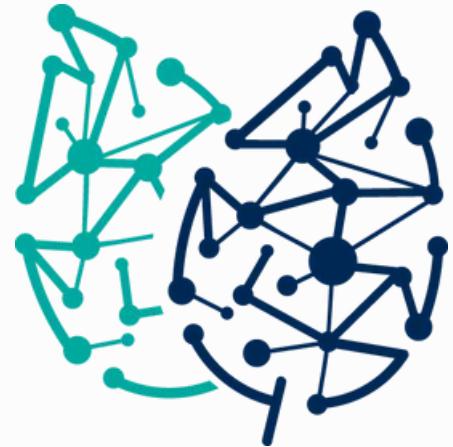


# Basic Syntax and **STRUCTURE**

Descendant Selector (Space)

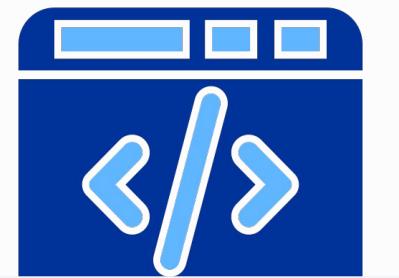


```
div p {  
  color: green;  
}
```

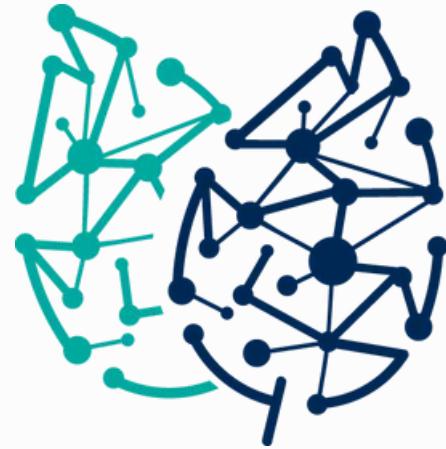


# Basic Syntax and **STRUCTURE**

Child Selector (>)



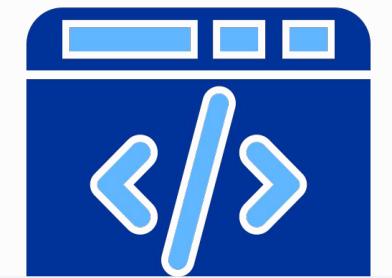
```
div > p {  
  margin-top: 10px;  
}
```



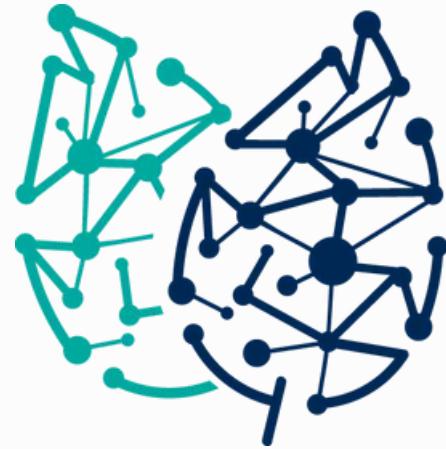
# Basic Syntax and **STRUCTURE**

Adjacent Sibling Selector

(4)

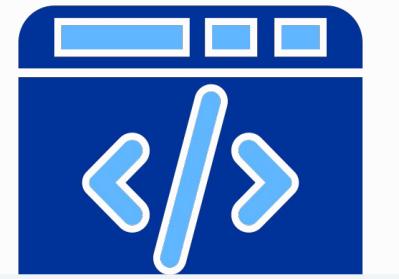


```
h1 + p {  
  font-size: 18px;  
}
```

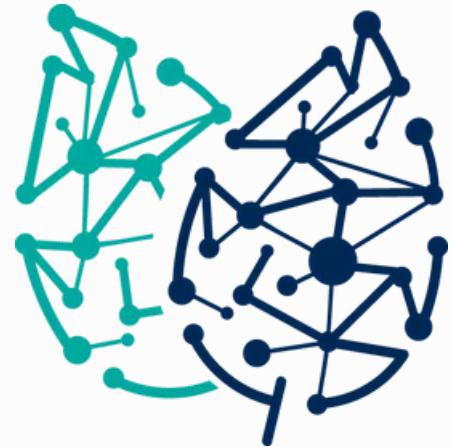


# Basic Syntax and **STRUCTURE**

General Sibling Selector (~)

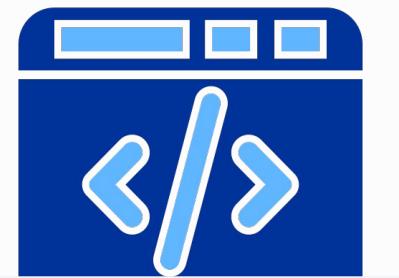


```
h1 ~ p {  
  color: blue;  
}
```

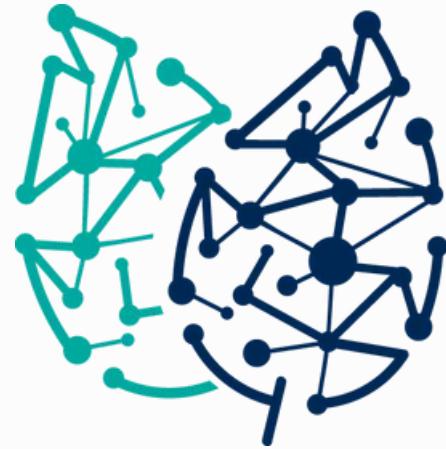


# Basic Syntax and **STRUCTURE**

Pseudo-classes

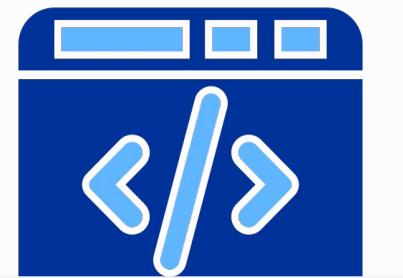


```
a:hover {  
    text-decoration: underline;  
}
```



# Basic Syntax and **STRUCTURE**

Pseudo-elements

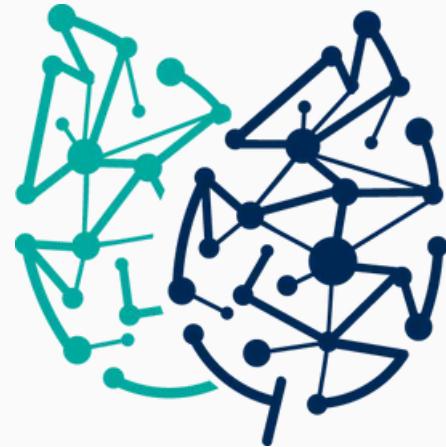


```
p::first-letter {  
    font-size: 2em;  
}
```



# Time to Reflect





Training



# JOURNE

Y

Introduction

1

Basic  
Concepts of  
CSS

2

Types of CSS  
Selectors

3

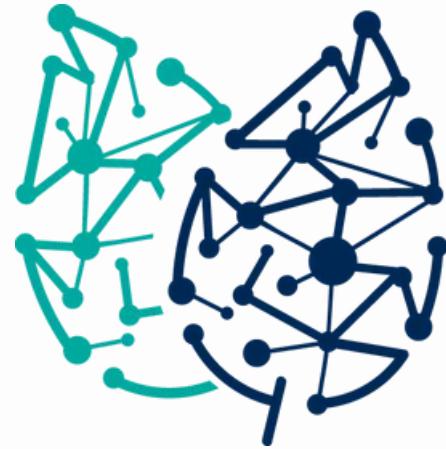
4

**Rules and ,  
Properties ,  
and Values**

5

How to  
specify color





# Basic Syntax and **STRUCTURE**

**Assigning Values to Properties  
in CSS**

**Properties**

**Attributes that define the style and layout of HTML elements.**

**Values**

**The specific settings or data assigned to properties**



# Color Properties

Example: **background-color: blue;**



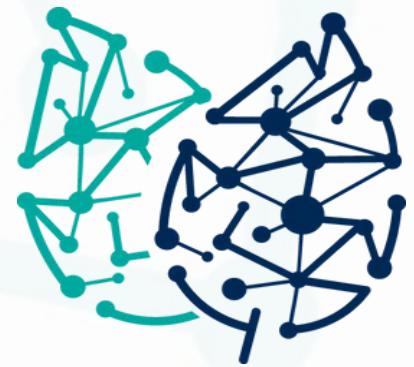
The screenshot shows a web browser window with the title "background-color property". The address bar indicates the file is located at "D:/CSS/programs/Ex...". The page content displays "Welcome to the javaTpoint.com" and four horizontal boxes with different background colors and their corresponding CSS code:

- background-color: lightgreen;
- background-color: pink;
- background-color: rgb(155, 55, 128);
- background-color: hsl(90, 22%, 50%);

color

background-color

border-color



# Font Properties

Example: `font-size: 16px;`

<u>FontWeight</u>	<u>Weight value</u>
<b>FontWeight ExtraBlack</b>	<b>950</b>
<b>FontWeight Black</b>	<b>900</b>
<b>FontWeight ExtraBold</b>	<b>800</b>
<b>FontWeight Bold</b>	<b>700</b>
FontWeight SemiBold	600
FontWeight Medium	500
FontWeight Normal	400
FontWeight SemiLight	350
FontWeight Light	300
FontWeight ExtraLight	200
FontWeight Thin	100

font color

font size

font weight

font family

font style

font spacing



# Text Properties

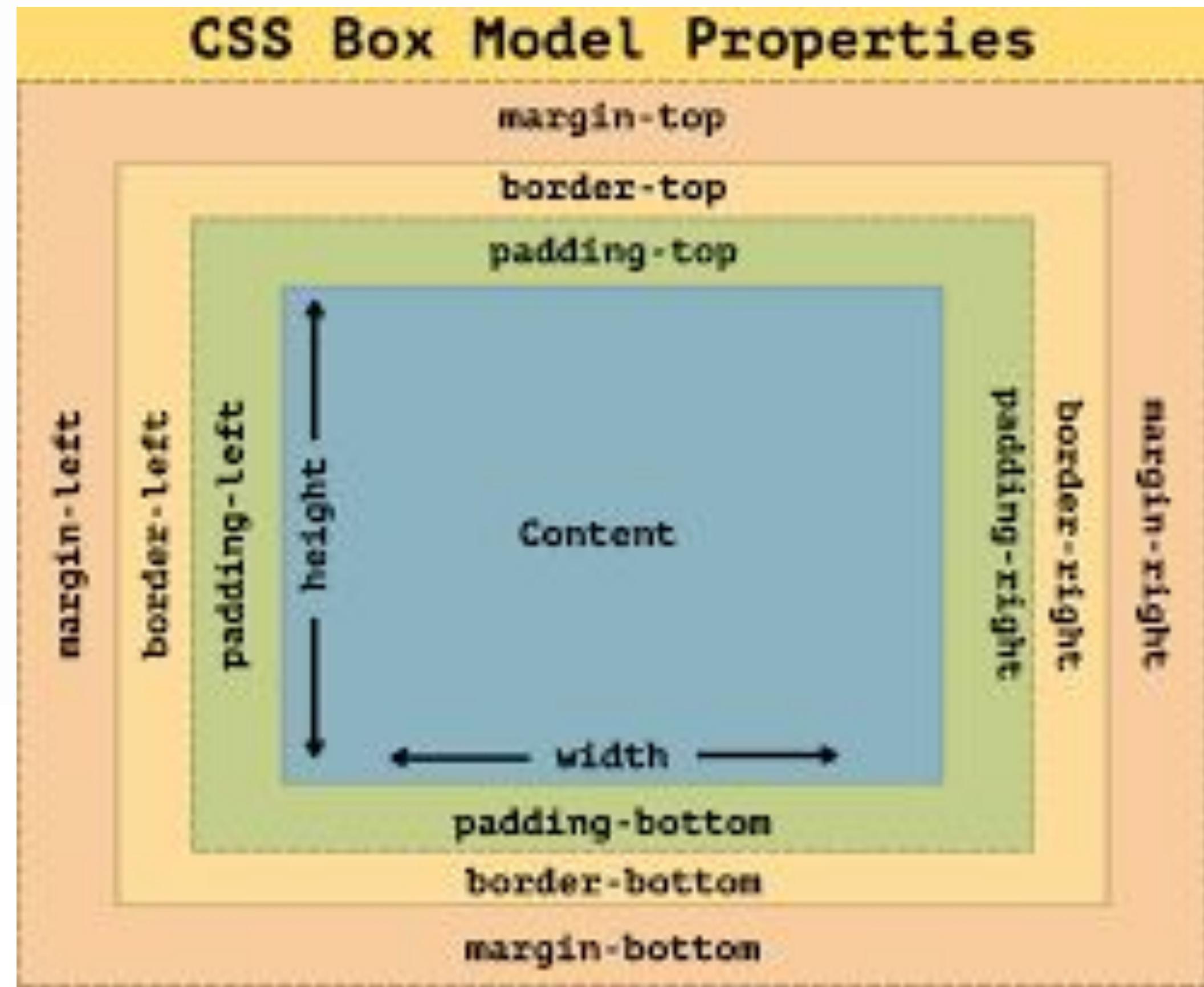
Example: `line-height: 3;`

LEFT	RIGHT
<p> Lorem ipsum dolor sit amet, vivamus consectetuer magna ipsum dignissim, a posuere volutpat mauris, magna pulvinar in vulputate ligula vel. At sem ante eu erat blandit. Blandit vestibulum dapibus libero mi quisque tortor, interdum tristique nulla vitae.</p>	<p> Lorem ipsum dolor sit amet, vivamus consectetuer magna ipsum dignissim, a posuere volutpat mauris, magna pulvinar in vulputate ligula vel. At sem ante eu erat blandit. Blandit vestibulum dapibus libero mi quisque tortor, interdum tristique nulla vitae.</p>
JUSTIFIED	CENTER
<p> Lorem ipsum dolor sit amet, vivamus consectetuer magna ipsum dignissim, a posuere volutpat mauris, magna pulvinar in vulputate ligula vel. At sem ante eu erat blandit. Blandit vestibulum dapibus libero mi quisque tortor, interdum tristique nulla vitae.</p>	<p> Lorem ipsum dolor sit amet, vivamus consectetuer magna ipsum dignissim, a posuere volutpat mauris, magna pulvinar in vulputate ligula vel. At sem ante eu erat blandit. Blandit vestibulum dapibus libero mi quisque tortor, interdum tristique nulla vitae.</p>



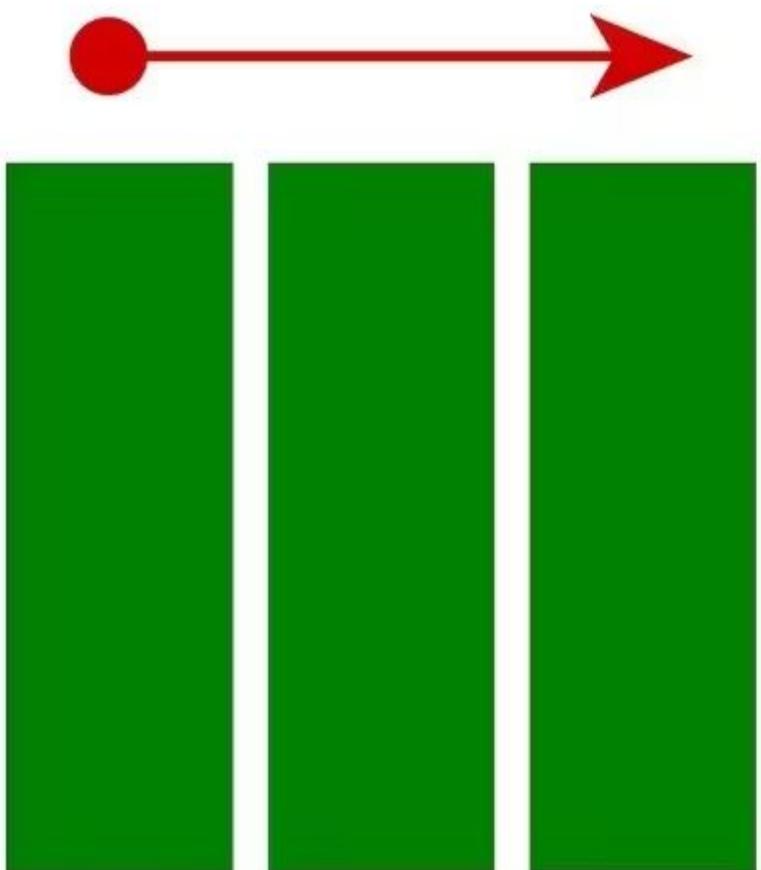


# Box Model Properties

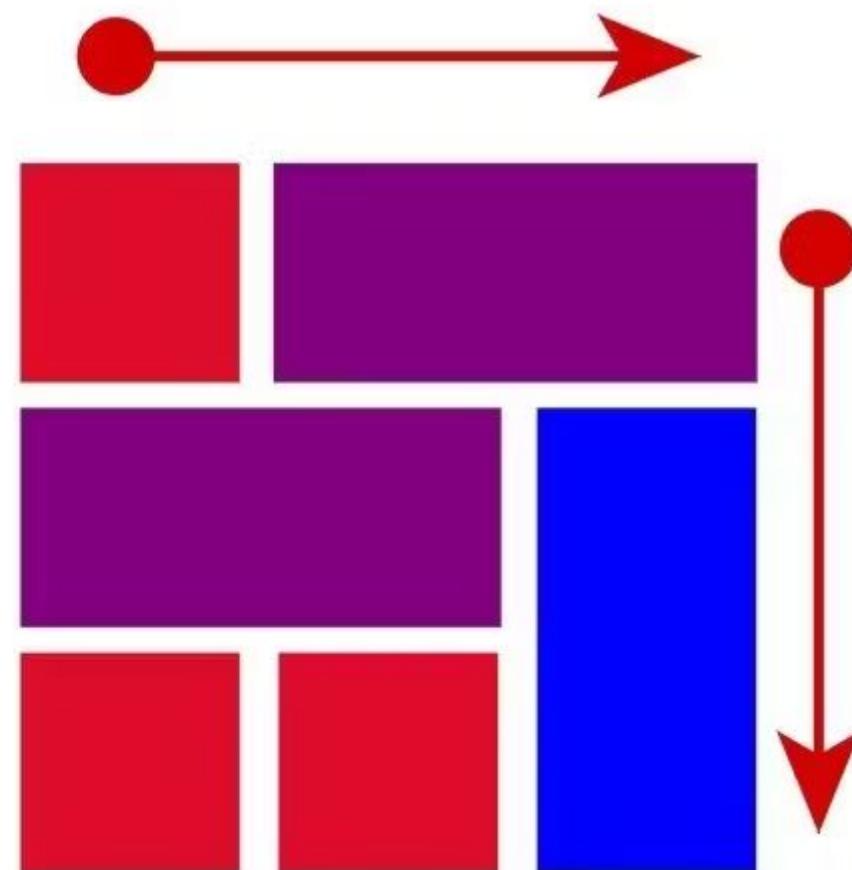




# Layout Properties



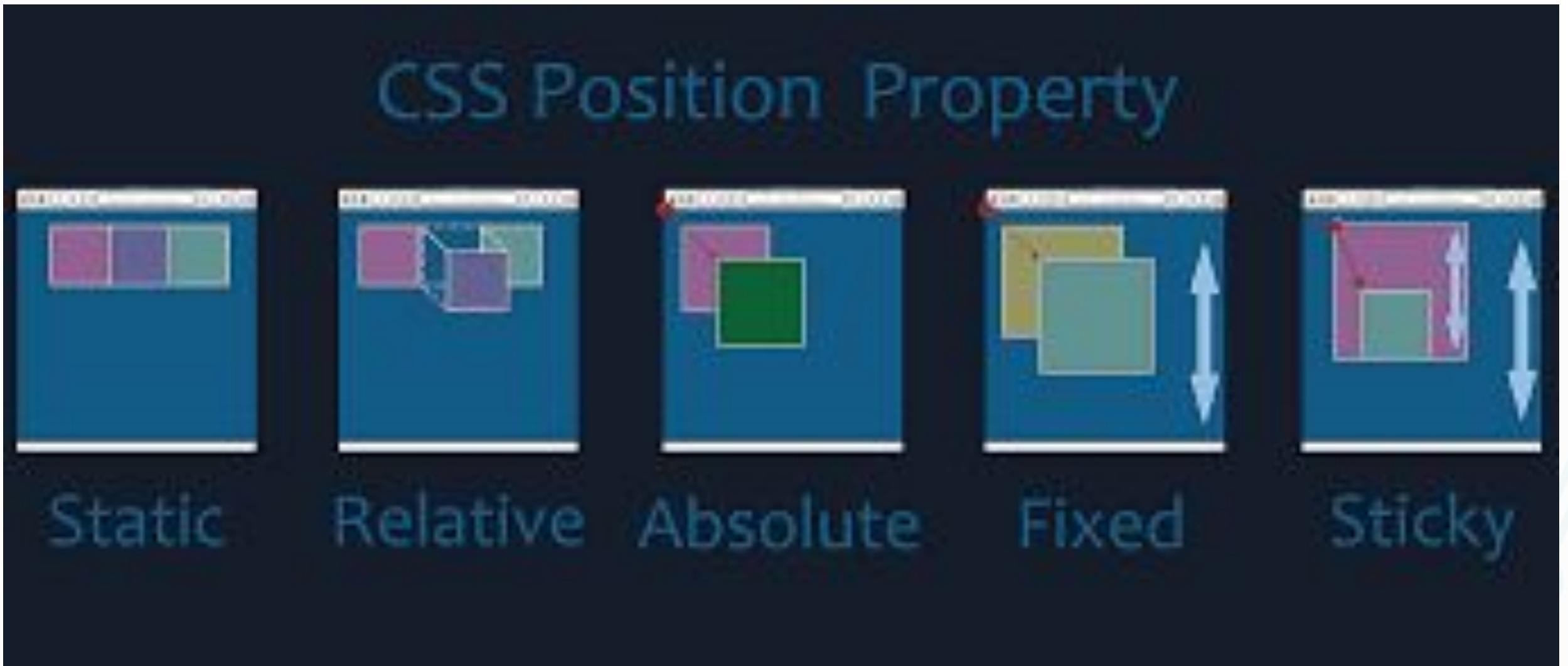
Flexbox  
One Dimensions



CSS Grids  
Two Dimensions

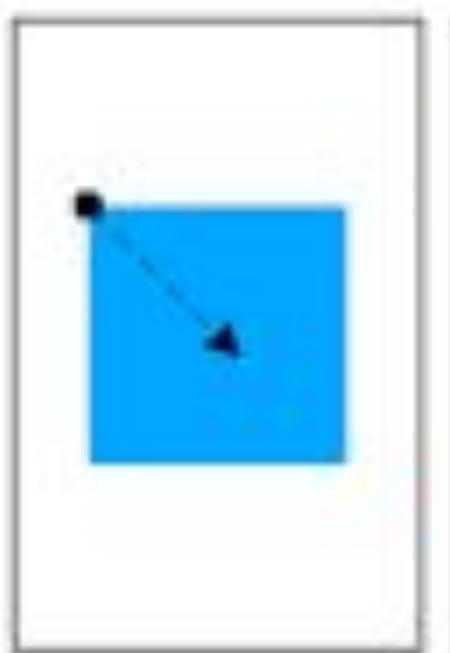


# Layout Properties

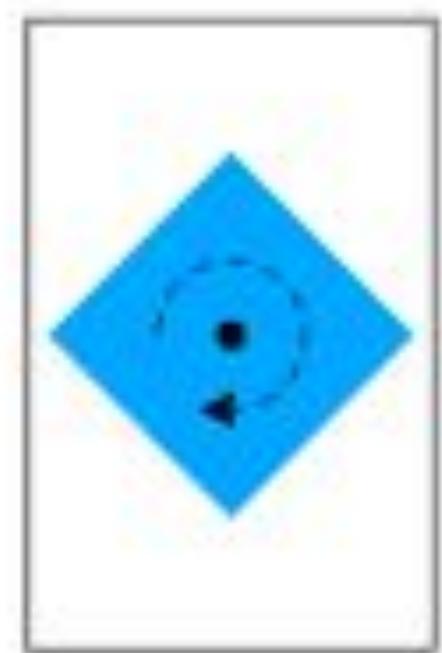
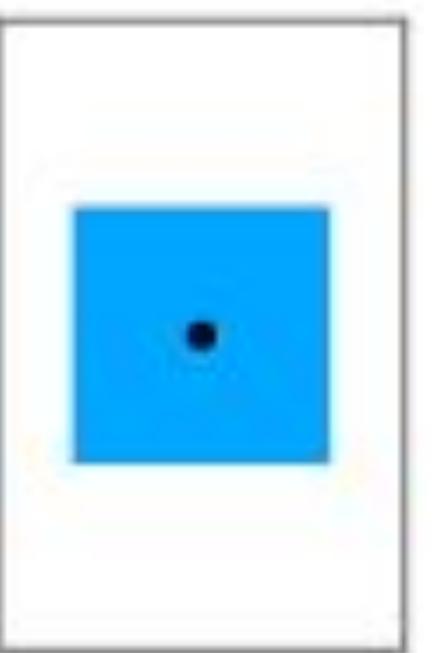




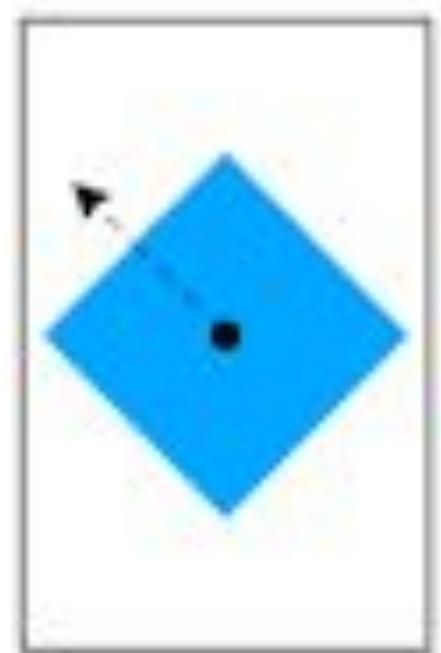
# Layout Properties



Apply transform-origin

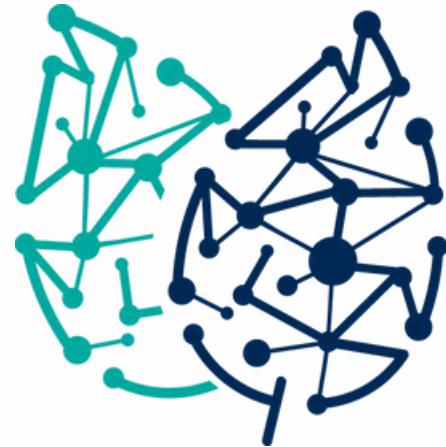


Apply transform  
functions



Unapply transform-origin





# Basic Syntax and STRUCTURE

## CSS Properties and Values

### CSS VALUES

*Enter a CSS Property*

help view all

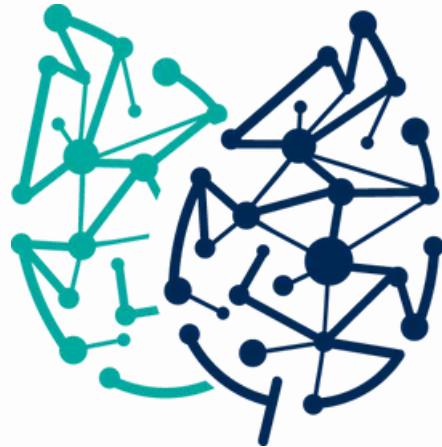
css2 css3+ flexbox animations 3d transforms paged media grid

scroll snap logical props tables masks positioning border-image

columns typography motion path compositing text-decoration

transforms scrollbars

aspects of an element's appearance or behavior, and values specify the settings for these properties.



# Basic Syntax and STRUCTURE

## Types of Values

**Predefined terms used as values.**

Example: 'color: red;'

**Can be specified by name, hex code, RGB, or other color models.**

Example: 'background-color: #ff5733;'

**Used for dimensions and spacing.**

Example: 'width: 50%; or margin: 10px;'

**Keyword**

s

**Numeric  
Values**

**Color**

s

**URL**

s

**Percentage  
and  
Length Units**

**Numbers often used with units.**

Example: 'font-size: 16px;'

**Used for properties like  
background-image.**

Example: 'background-image:  
url('image.jpg');'



# Time To Reflect





Training



# JOURNE

Introduction

1

Basic  
Concepts of  
CSS

2

Types of CSS  
Selectors

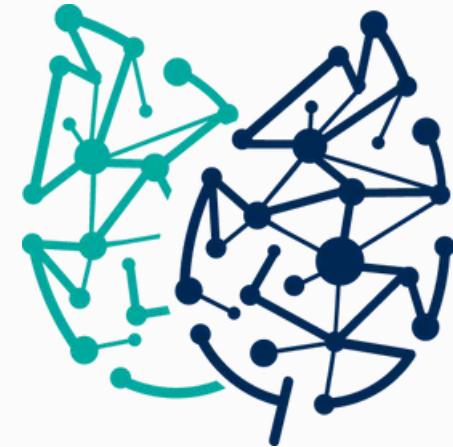
3

Rules and ,  
Properties ,  
and Values

4

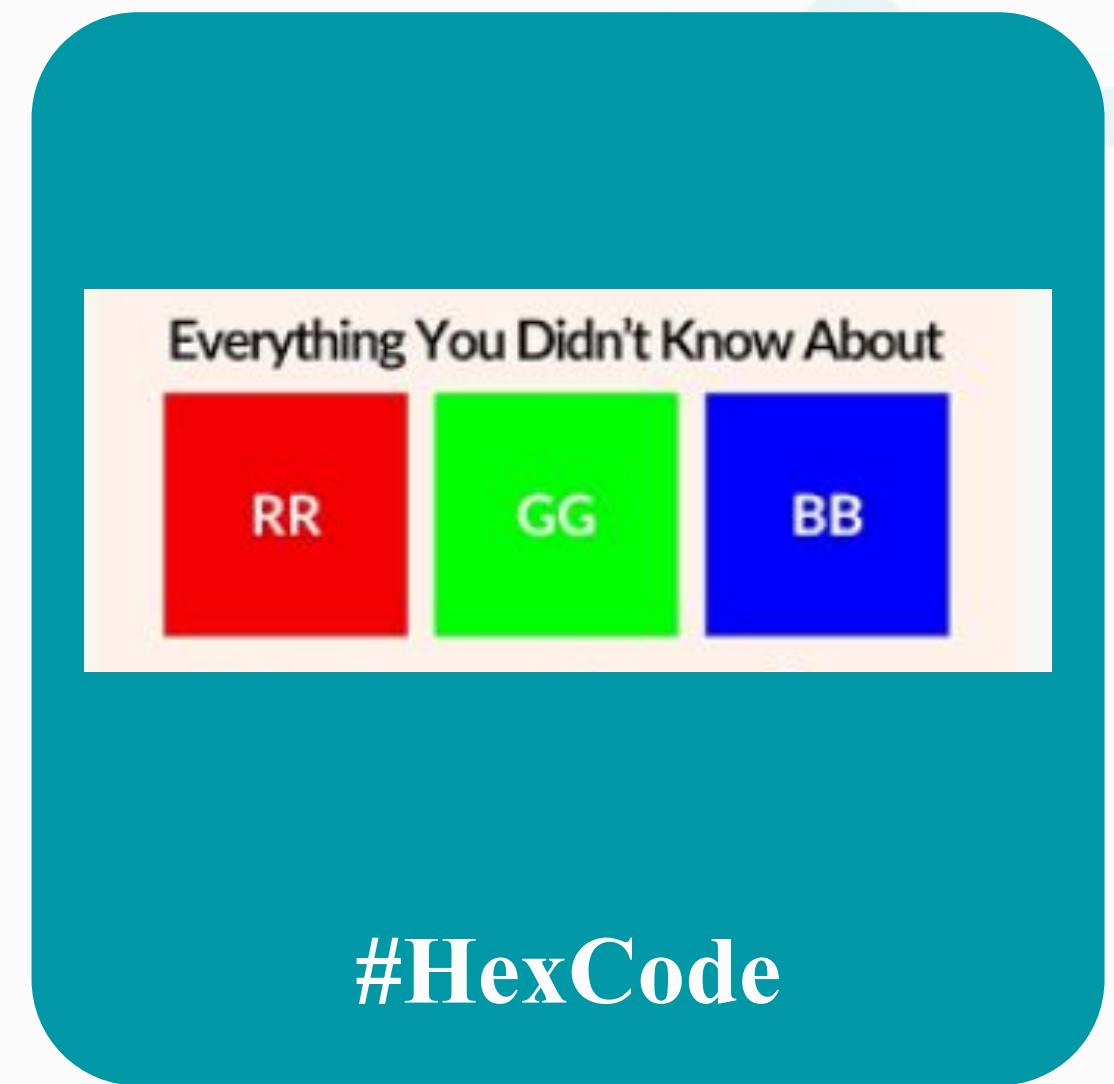
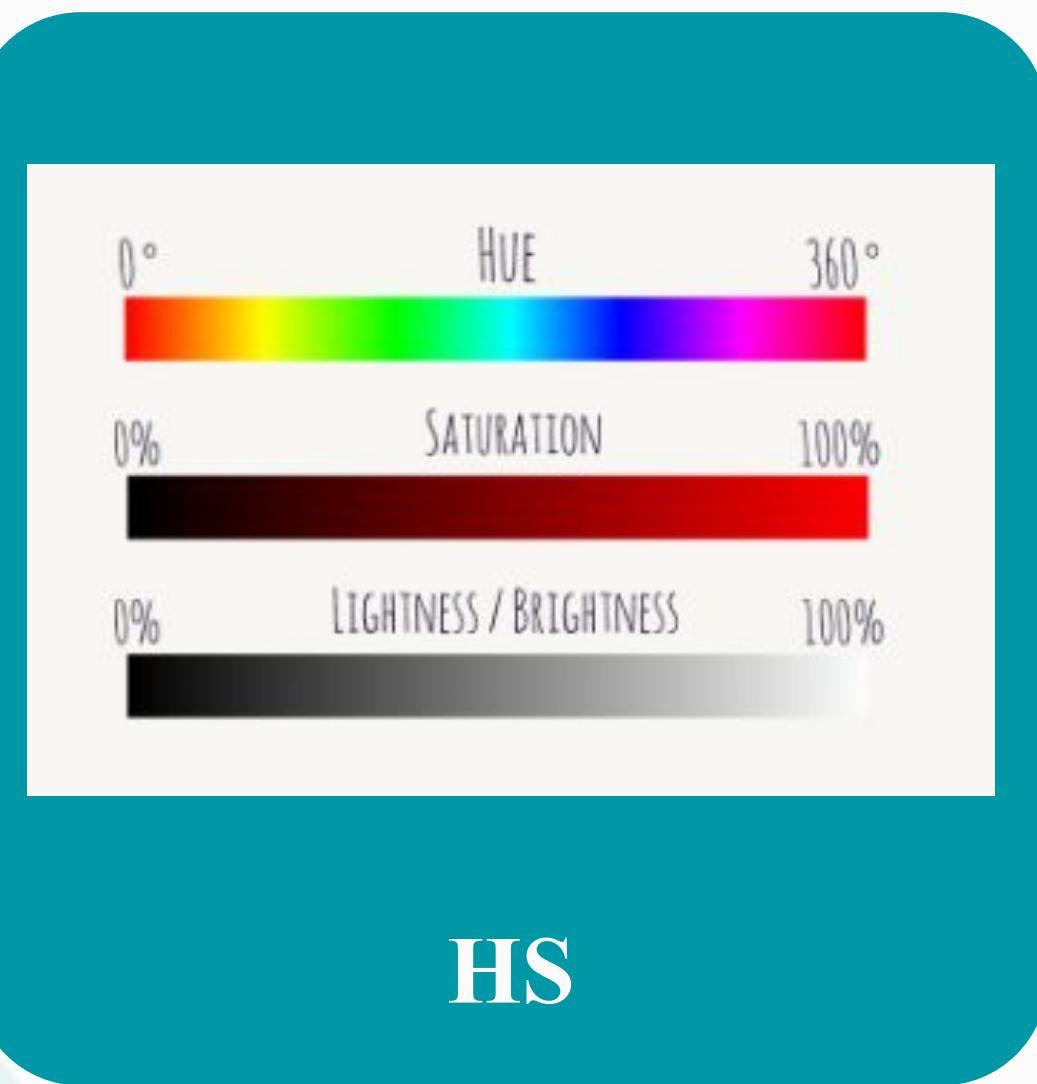
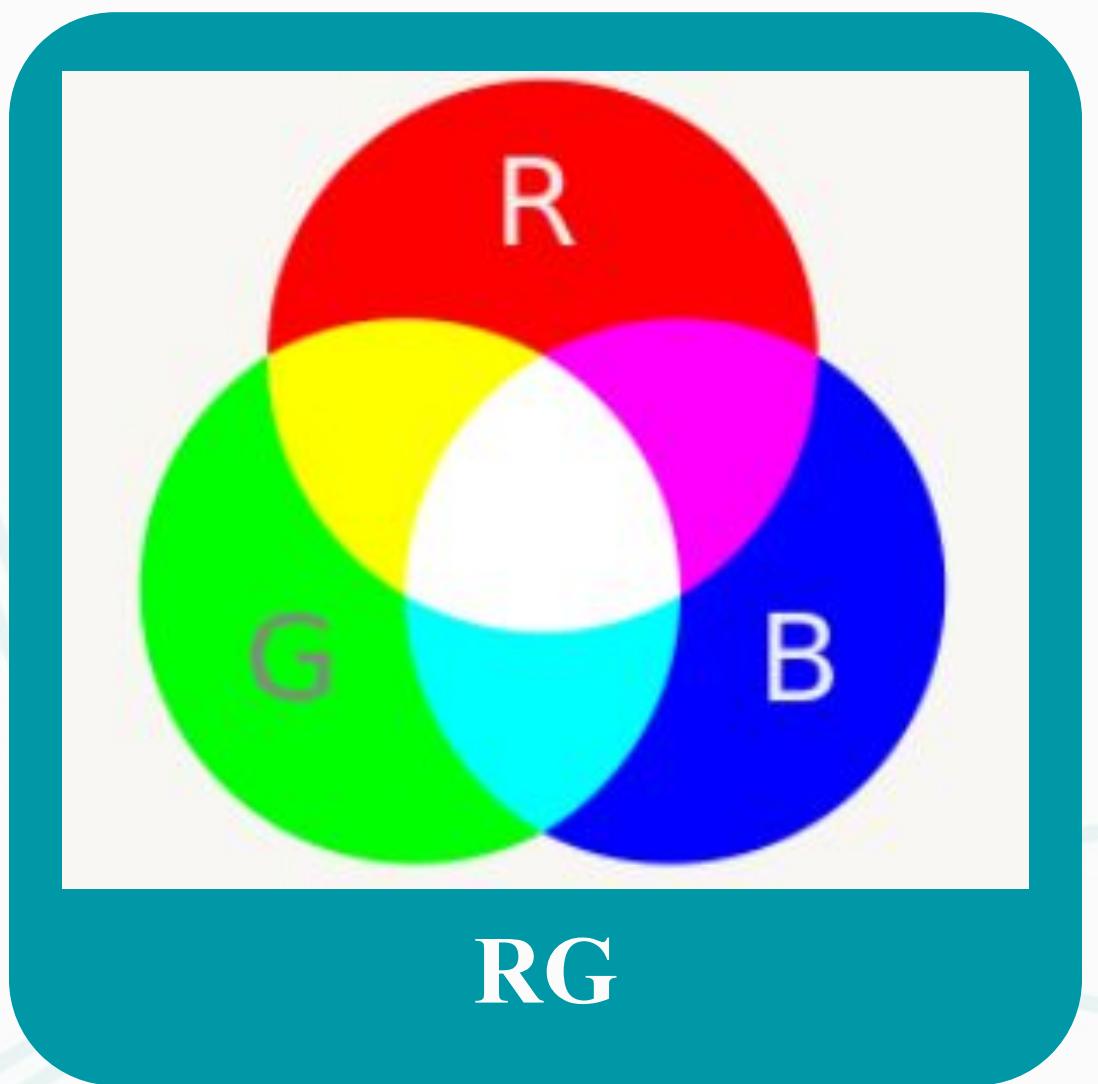
**How to specify  
color**

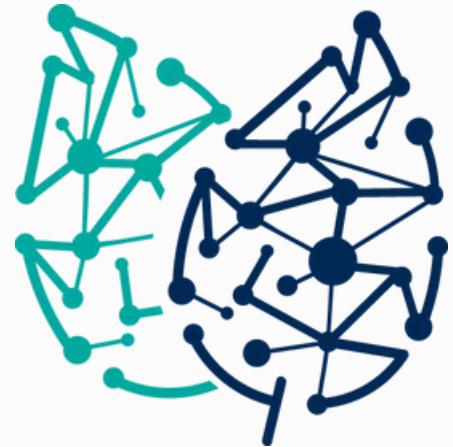




# Basic Styling

## Methods for Specifying Colors





# Basic Styling

## Methods for Specifying Colors

### Color Names

- Simple, predefined color names.
- Example: ‘color: green;’

### Hexadecimal Code

- Six-digit codes representing colors.
- Example: ‘color: #ff5733;’

### RGB

- Numeric values ranging from 0 to 255.
- Example: ‘color: rgb(255, 0, 0);’

### HSL

- Values define the hue, saturation, and lightness.
- Example: ‘color: hsl(120, 100%, 50%);’

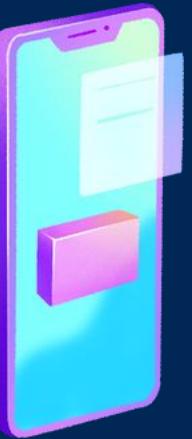


# Time to Reflect!



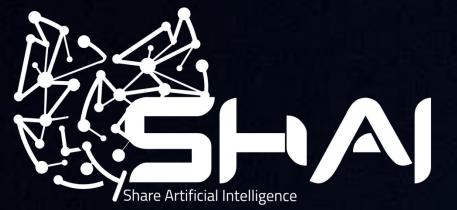


Share Artificial Intelligence



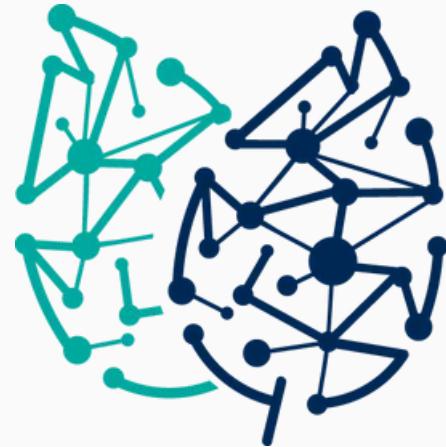
Gamify  
Your  
Training





# Elevate the Elements





Training



# JOURNEY

1

Y

2

3

4

5

6

**Text Styling in  
CSS**

Box Model

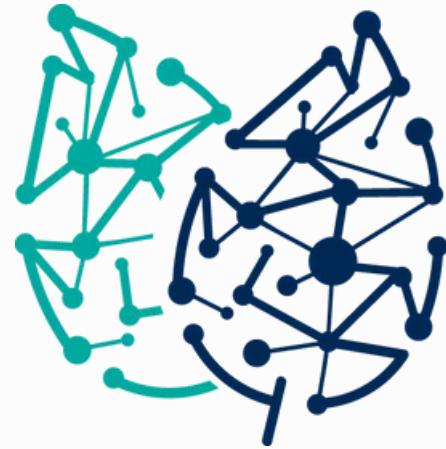
List Styling  
in CSS

Table  
Styling in  
CSS

Form  
Styling in  
CSS

Positioning  
in CSS





# Text Styling

## Font Size

**px**

Pixels, a fixed unit.

**%**

Percentage of the parent element's font size.

**em**

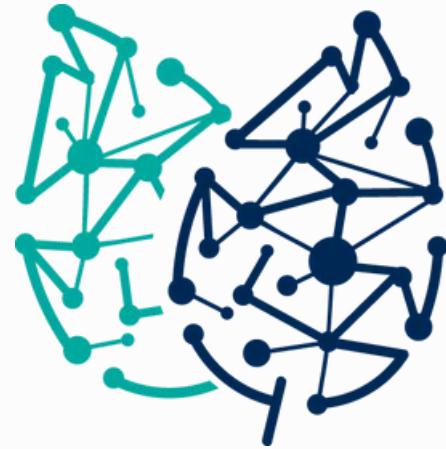
Relative to the parent element's font size.

**vw/vh**

Viewport width/height (can scale text according to screen size).

**rem**

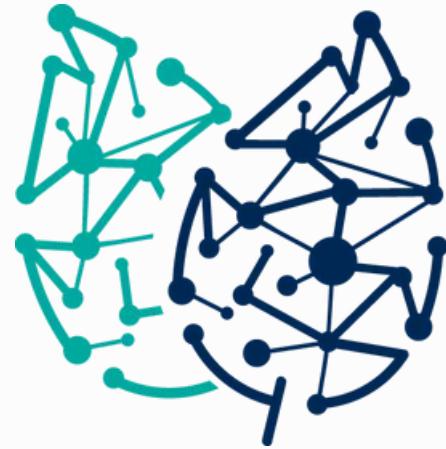
Relative to the root element (<html>), often used for consistent scaling.



# Basic Styling

## Bold Text (font-weight)

```
font-weight: 100; Ultra Light  
font-weight: 200; Thin  
font-weight: 300; Light  
font-weight: 400; Regular  
font-weight: 500; Medium  
font-weight: 600; Semi bold  
font-weight: 700; Bold  
font-weight: 800; Heavy  
font-weight: 900; Black
```



# Basic Styling

```
p.normal {  
    font-style: normal;  
}  
  
p.italic {  
    font-style: italic;  
}  
  
p.oblique {  
    font-style: oblique;  
}
```

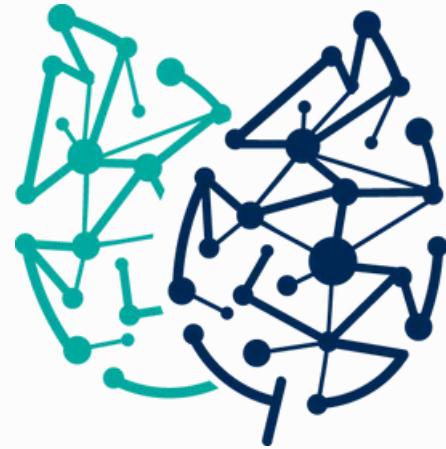
## Italic Text (font-style)

### The font-style property

This is a paragraph in normal style.

*This is a paragraph in italic style.*

*This is a paragraph in oblique style.*



# Basic Styling

```
div.a {  
  line-height: normal;  
}  
  
div.b {  
  line-height: 1.6;  
}  
  
div.c {  
  line-height: 80%;  
}  
  
div.d {  
  line-height: 200%;  
}
```

## Spacing Between Lines (line-height)

### **line-height: normal (default):**

This is a paragraph with a standard line-height.  
The standard line height in most browsers is about 110% to 120%.

### **line-height: 1.6 (recommended):**

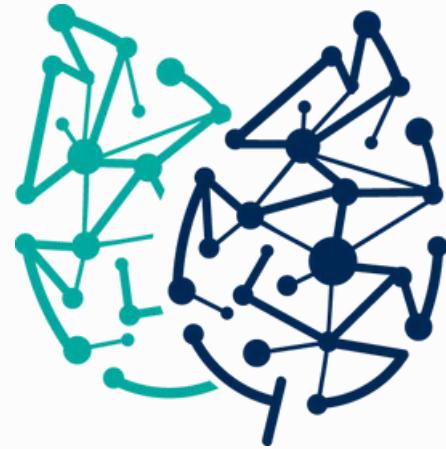
This is a paragraph with the recommended line-height.  
The line height is here set to 1.6. This is a unitless value;  
meaning that the line height will be relative to the font size.

### **line-height: 80%:**

This is a paragraph with a smaller line-height.  
The line height is here set to 80%.

### **line-height: 200%:**

This is a paragraph with a bigger line-height.  
The line height is here set to 200%.



# Basic Styling

```
p.a {  
  word-spacing: normal;  
}  
  
p.b {  
  word-spacing: 30px;  
}  
  
p.c {  
  word-spacing: 1cm;  
}
```

## Spacing Between Words (word-spacing)

### **word-spacing: normal:**

This is some text. This is some text.

### **word-spacing: 30px:**

This is some text. This is some text.

### **word-spacing: 1cm:**

This is some text. This is some text.



# Time to Reflect





Training

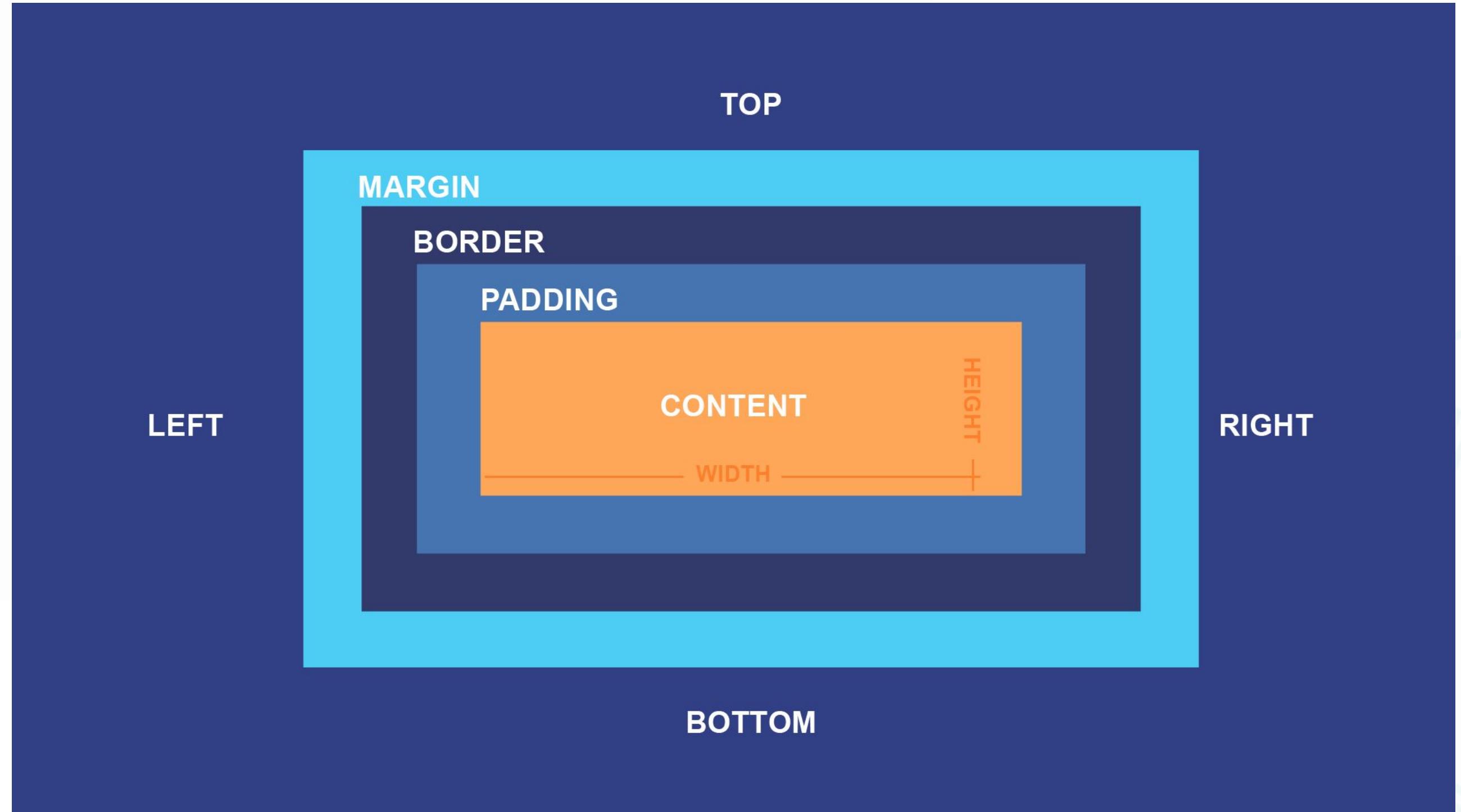


# JOURNEY





# Box Model Properties





# Time to Reflect



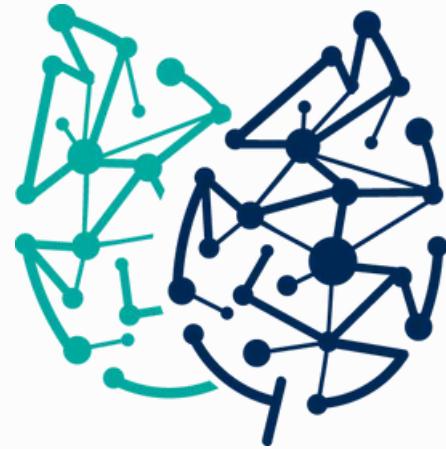


Training



# JOURNEY





# Basic Styling

```
ol.c {list-style-type: upper-roman;}  
ol.d {list-style-type: lower-alpha;}
```

- I. Coffee
  - II. Tea
  - III. Coca Cola
- 
- a. Coffee
  - b. Tea
  - c. Coca Cola

## List Styling in CSS

```
ul.a {list-style-type: circle;}  
ul.b {list-style-type: square;}
```

- Coffee
  - Tea
  - Coca Cola
- 
- Coffee
  - Tea
  - Coca Cola



# Time to Reflect



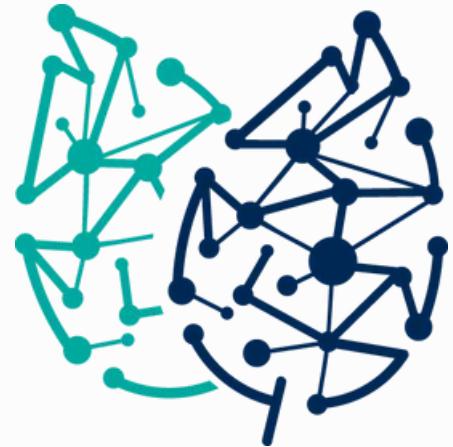


Training



# JOURNEY





# Basic Styling

## Table Styling in CSS

```
#table2 {  
    border-collapse: collapse;  
    border-spacing: 10px;  
}
```

Firstname	Lastname
Peter	Griffin
Lois	Griffin

```
#table1 {  
    border-collapse: separate;  
    border-spacing: 10px;  
}
```

Firstname	Lastname
Peter	Griffin
Lois	Griffin



# Time to Reflect



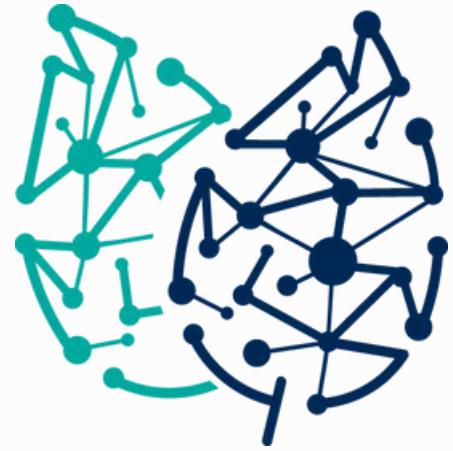


Training



# JOURNEY





# Basic Styling

## Form Styling in CSS

**width**

**padding**

**border**

**background-color**

**text-align**

First Name

Last Name

Country

Submit



# Time to Reflect

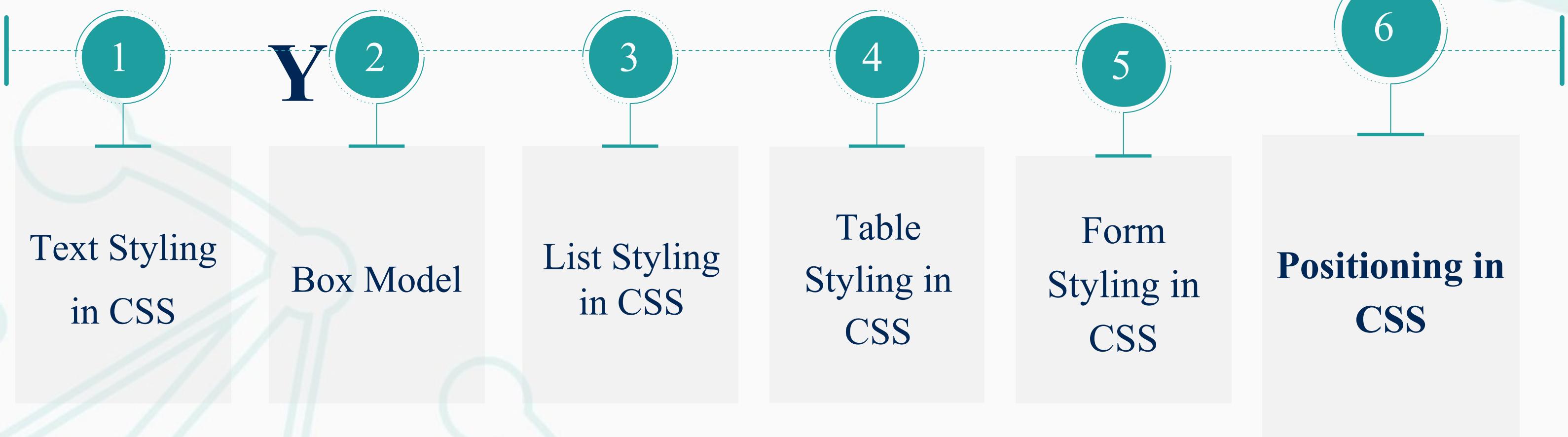


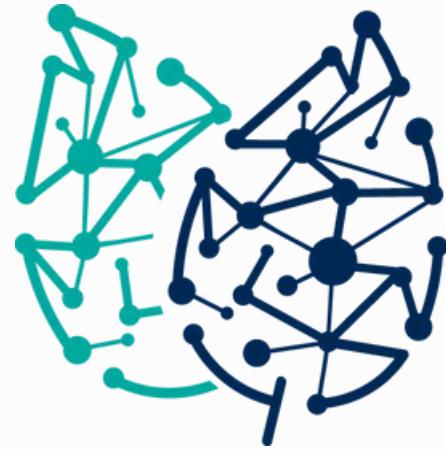


Training



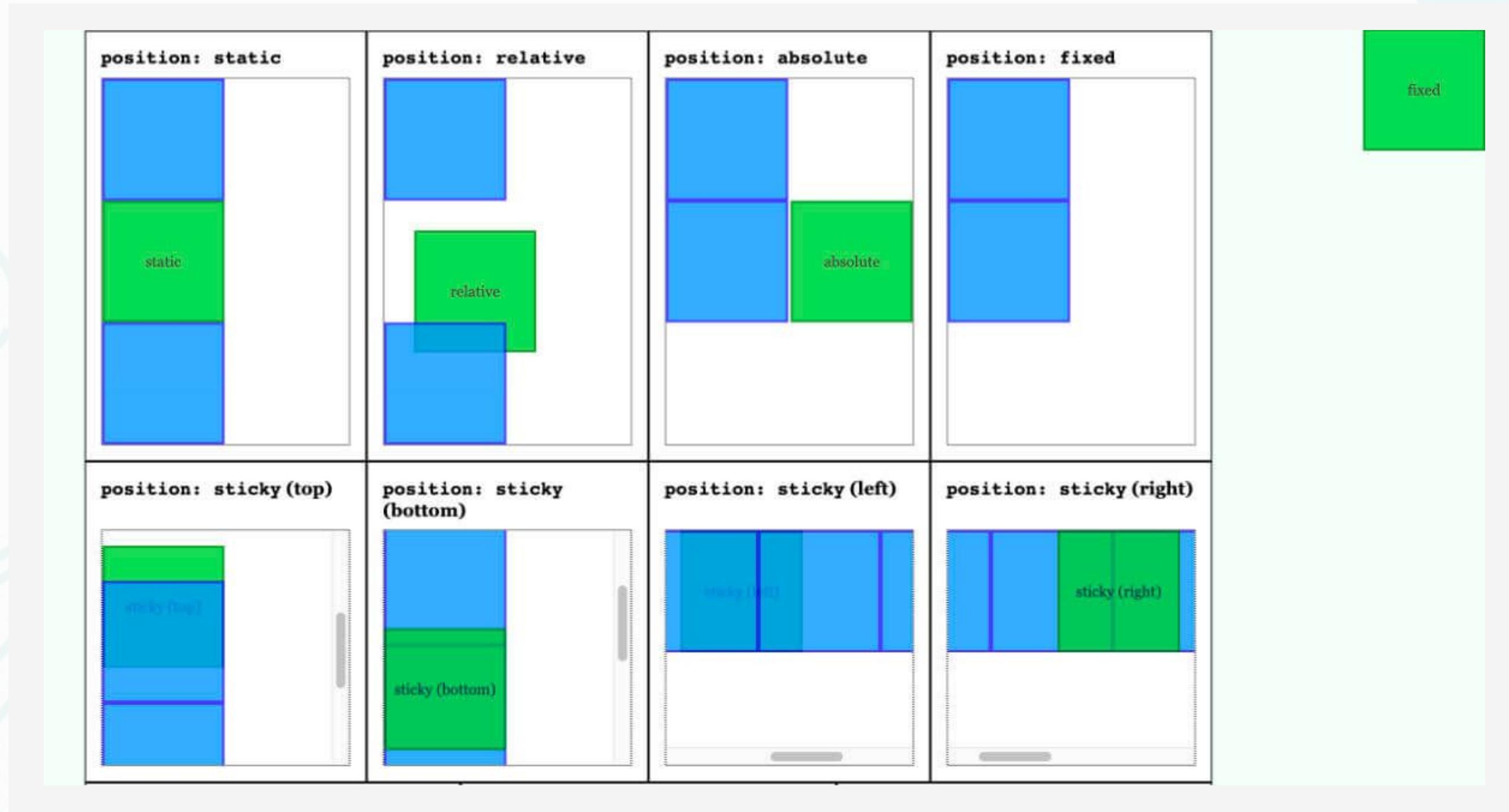
# JOURNEY





# Basic Styling

## Positioning in CSS







# Code Like Pro





Training

# JOURNE



**Basic Concepts  
of CSS Grid**



Basic  
Concepts of  
CSS Flex

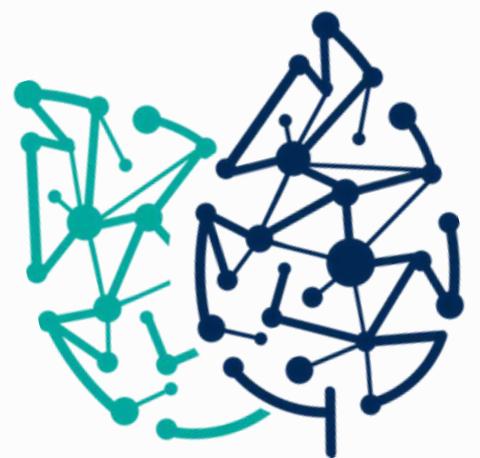


Responsive

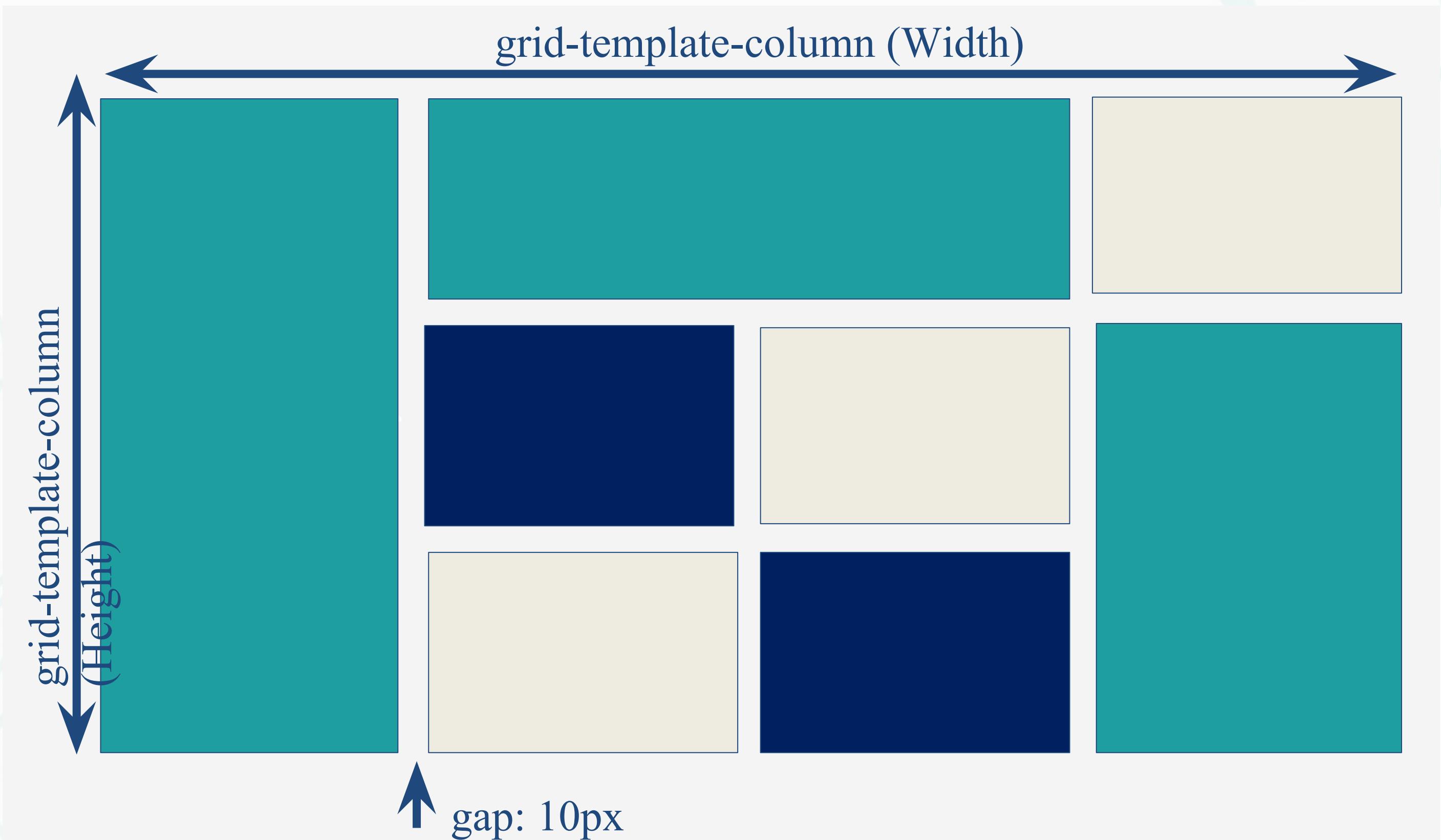


Ways to  
name selector





# Basic Concepts of CSS Grid





Training

# JOURNE



Basic  
Concepts of  
CSS Grid



**Basic Concepts of  
CSS Flex**



Responsive

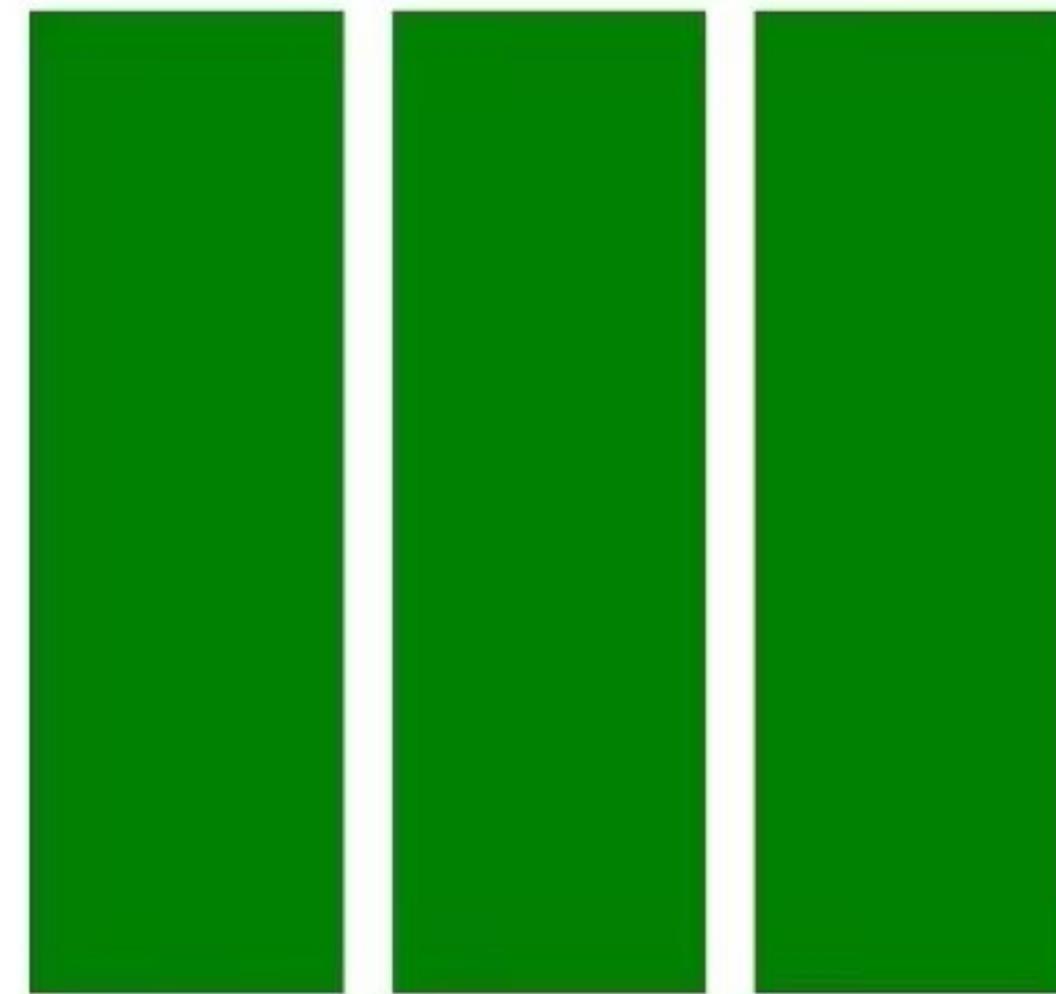


Ways to  
name selector





# Basic Concepts of CSS Flex



**Flexbox**  
One Dimensions



# flex-direction

row



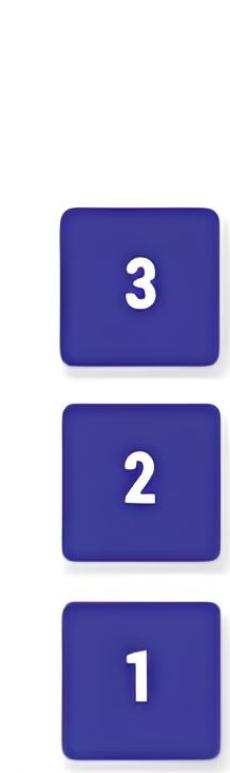
row-reverse

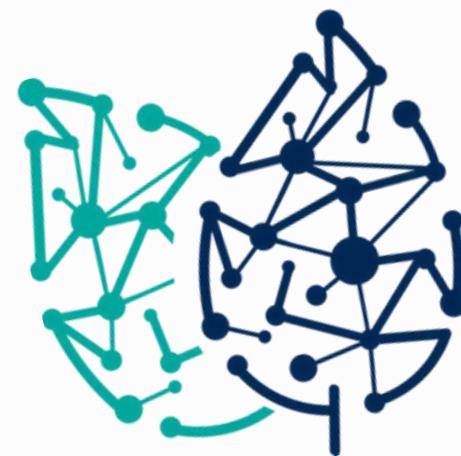


column



column-reverse





# justify-content

flex-start

- 1
- 2
- 3

center

- 1
- 2
- 3

flex-end

- 3
- 2
- 1

space-between

- 1
- 2
- 3

space-around

- 1
- 2
- 3

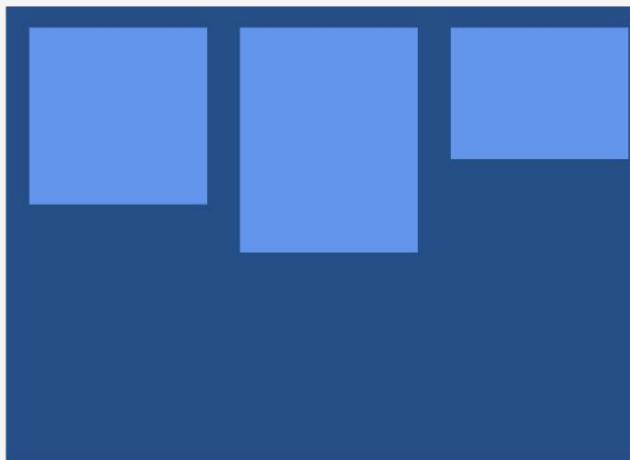


# align-items

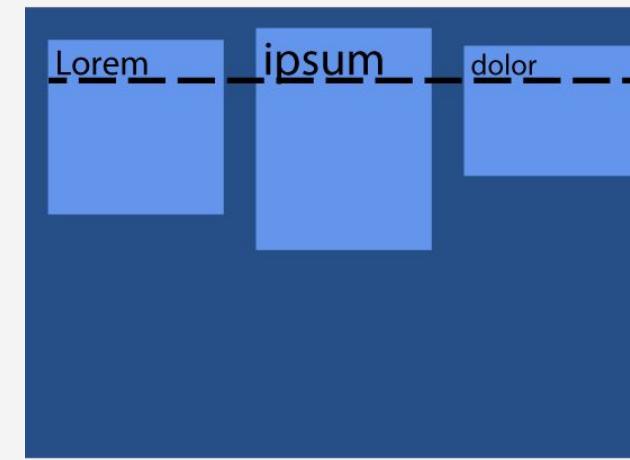
stretch



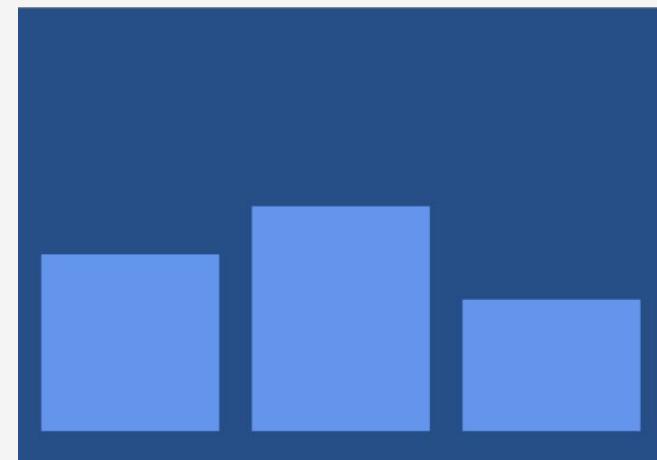
flex-start



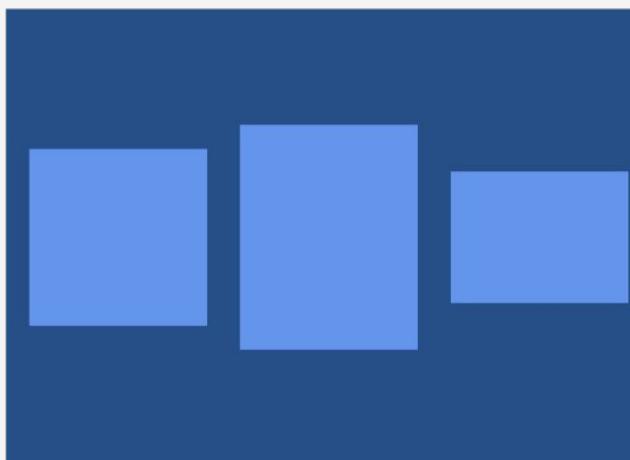
baseline



flex-end



center





# align-self

align-self-start

align-self-end

align-self-center

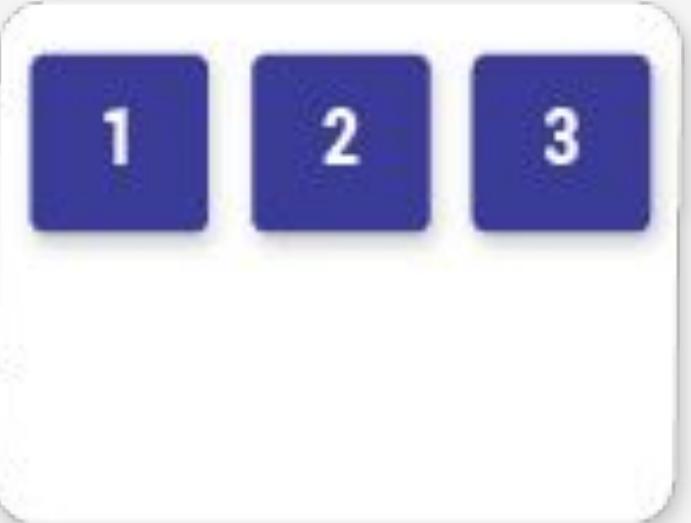
align-self-baseline

align-self-stretch



# flex-wrap

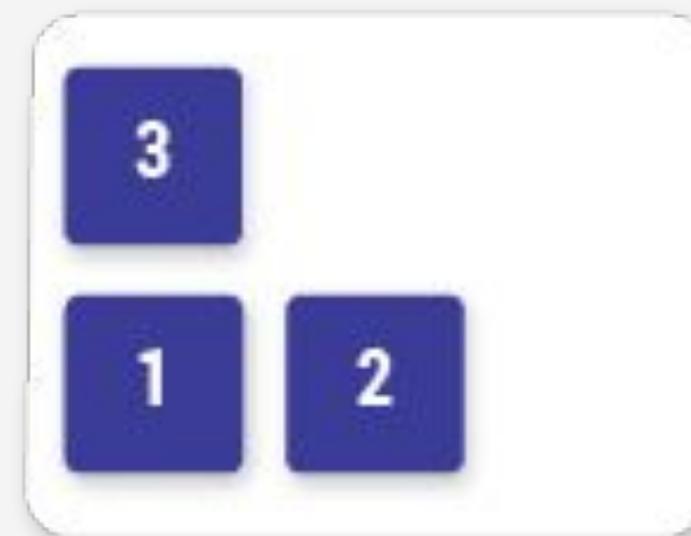
nowrap



wrap



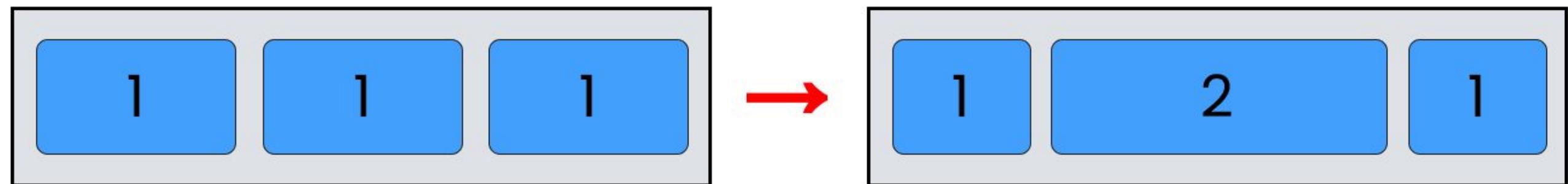
wrap-reverse



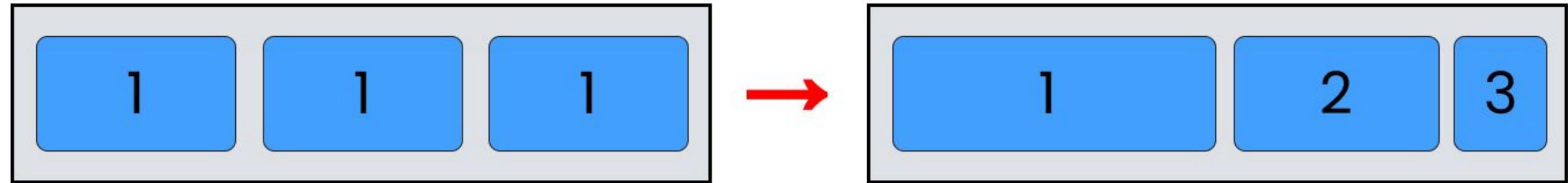


flex

flex-grow



flex-shrink





Training

# JOURNE



Basic  
Concepts of  
CSS Grid



Basic  
Concepts of  
CSS Flex



**Responsive**



Ways to  
name selector





# Responsive





Training

# JOURNE



Basic  
Concepts of  
CSS Grid



Basic  
Concepts of  
CSS Flex



Responsive



**Ways to name  
selector**





# Camel Case

First word is lowercase, subsequent words start with uppercase letters

.myButton  
.mainContainer  
.activeUser





# Pascal Case

Pascal case is similar to camel case, but with the first letter of each word capitalized

.MyButton  
.MainContainer  
.ActiveUser

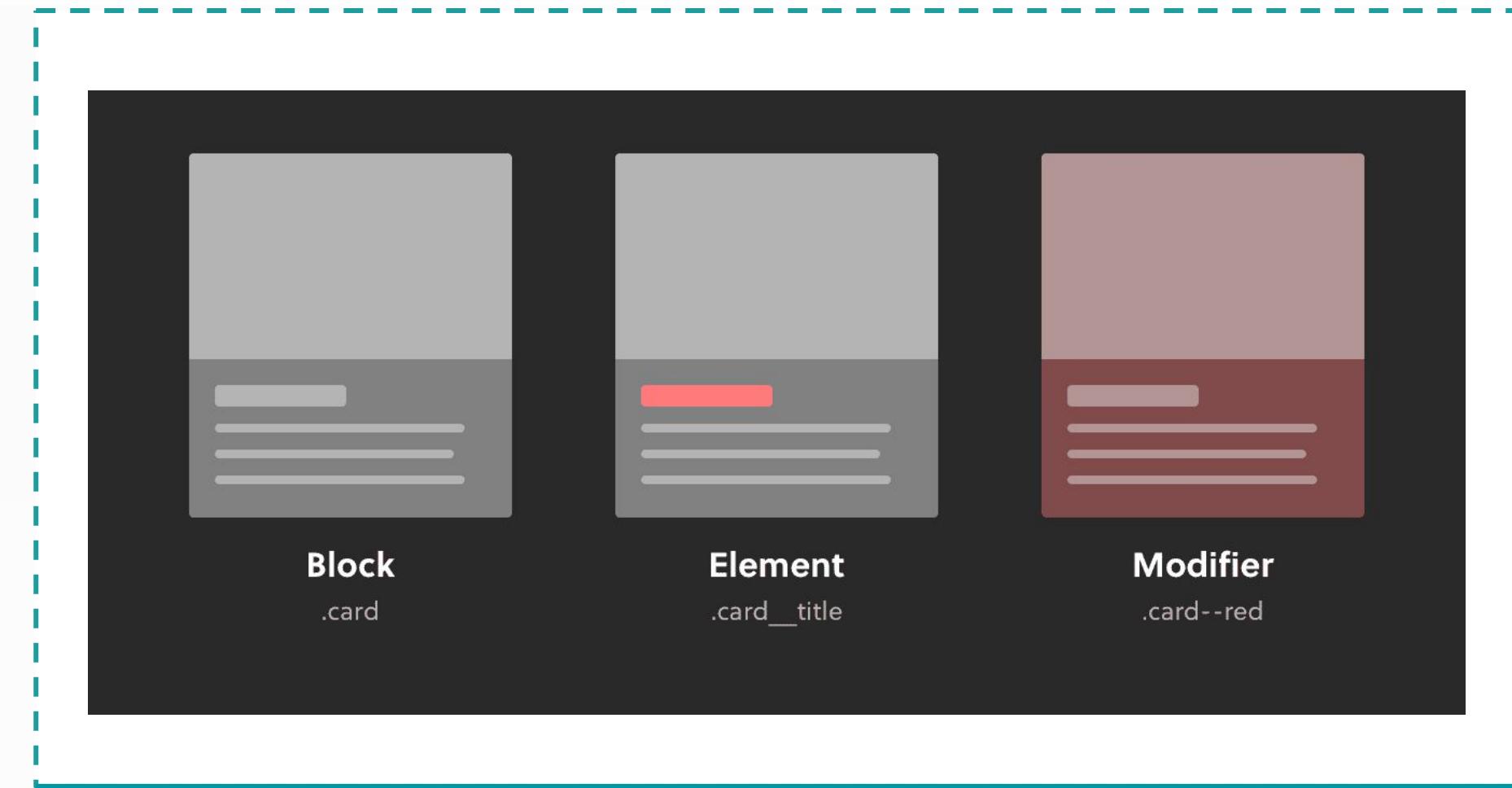




# BEM (Block, Element, Modifier)

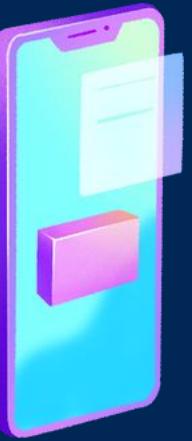
It stands for Block, Element, and Modifier. It's designed to create modular and reusable components.

.menu  
.menu\_\_item  
.menu\_\_item--active





Share Artificial Intelligence



Gamify  
Your  
Training



# KWL Strategy



k

What I know

w

What I want to learn

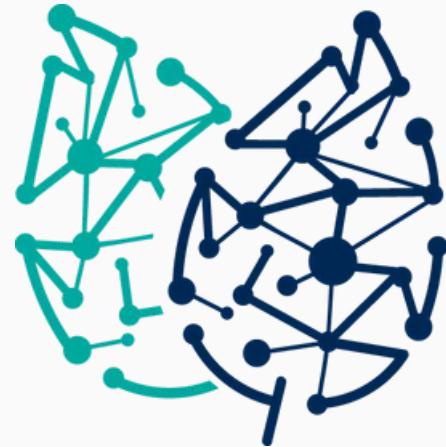
l

What I learned



# Time to Reflect



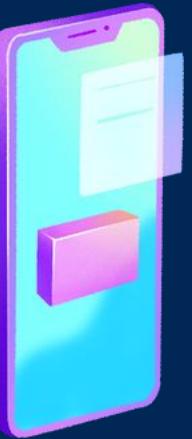


Q&A





Share Artificial Intelligence



Gamify  
Your  
Training





# Assignment

