



Pengembangan Web Praktik

HTML & CSS

Basics



Endang Anggiratih, S.T., M.Cs.

September, 2024

Materi perkuliahan

- | | | | |
|---|---|----|--|
| 1 | Perkenalan, Silabus, Pengenalan Konsep web dan server | 8 | Konsep microservices dan Rest API pada Flask |
| 2 | Dasar UI/UX Design, Instalasi Flask | 9 | Migration Model CRUD Rest API pada Flask |
| 3 | Integrasi, HTML & CSS Dasar | 10 | Penerapan API JWT pada Flask |
| 4 | Integrasi Flask View Template | 11 | Integrasi Flask dan Javascript |
| 5 | Integrasi Flask Bootstrap dan Mysql CRUD | 12 | JavaScript Custom Components |
| 6 | Penerapan Session pada Flask | 13 | Presentasi Projek Besar |
| 7 | Responsi | 14 | Presentasi Projek Besar |

HyperText Markup Language



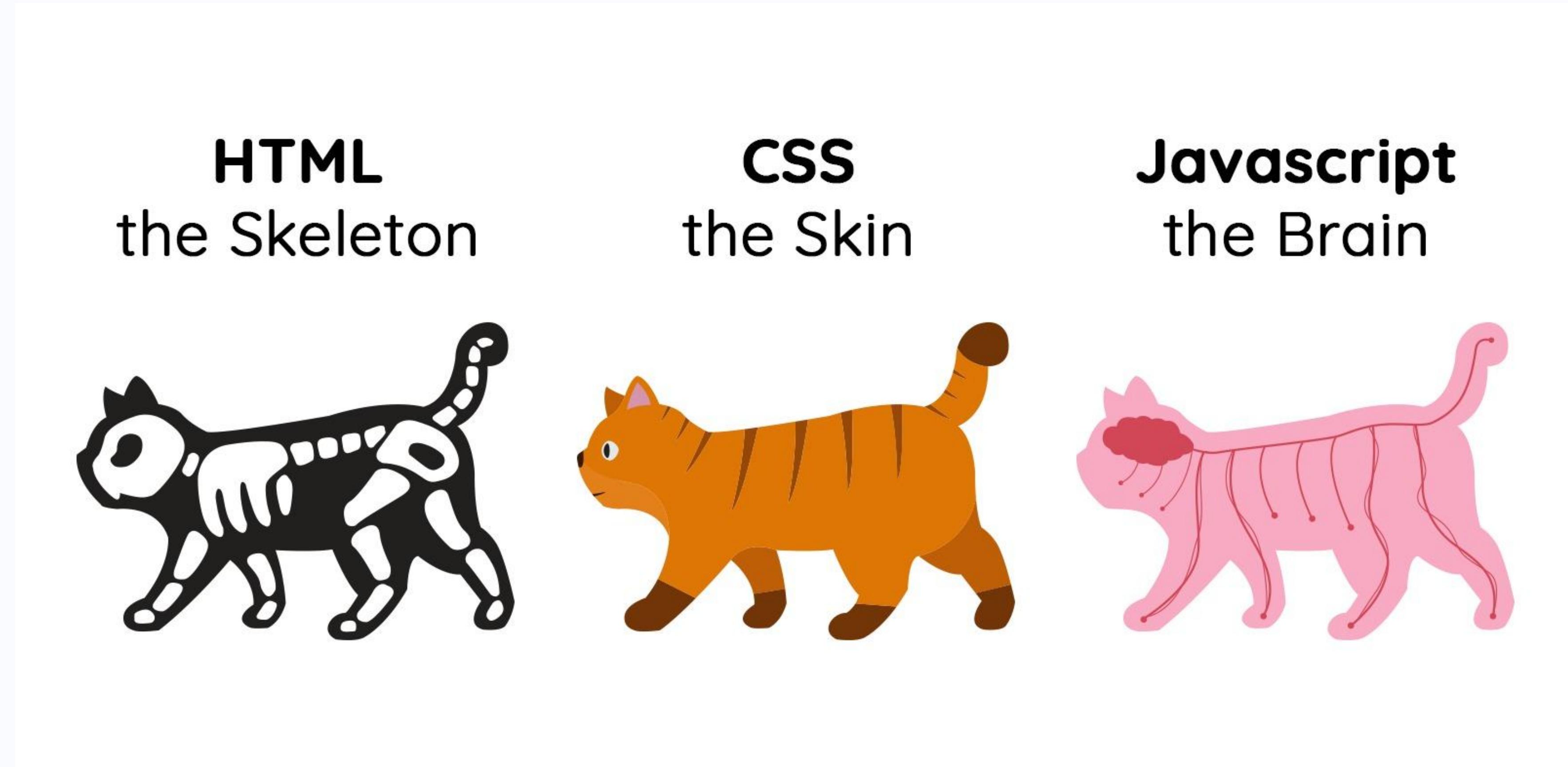
It is the standard language used to create and design documents on the World Wide Web, **defining the structure and layout** of web pages through the use of tags and attributes.

Cascading Style Sheet

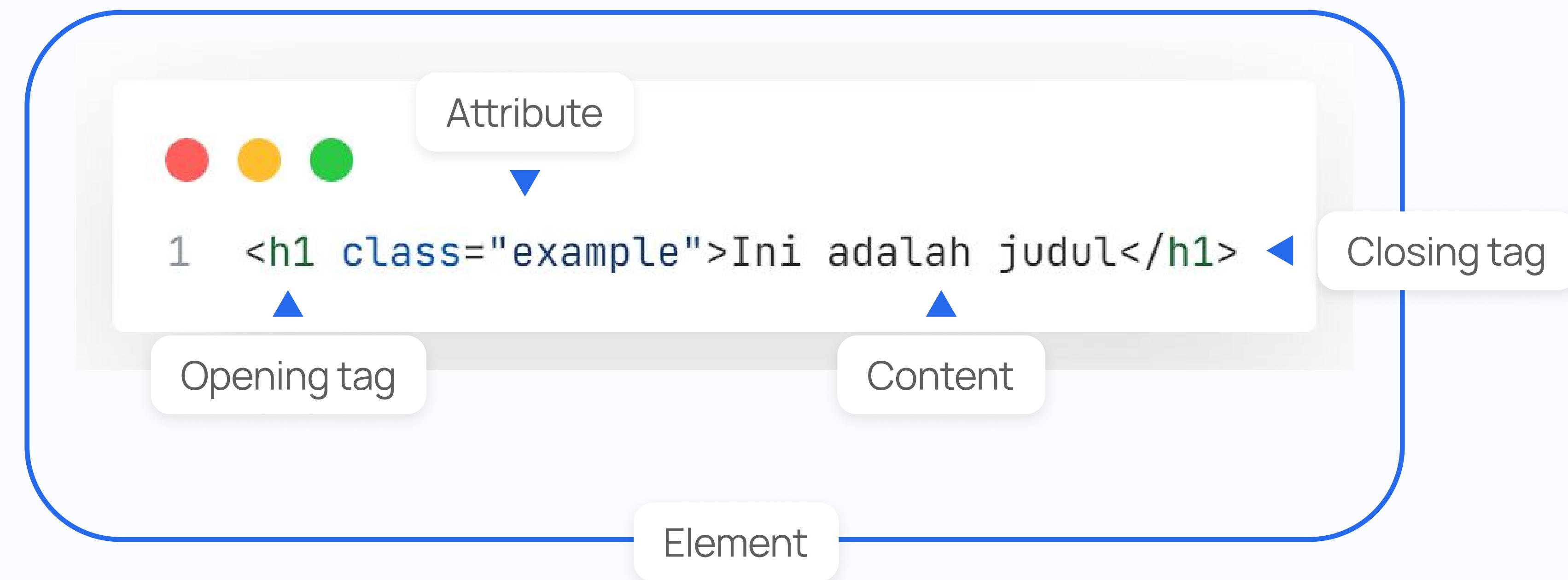


CSS is used to **define the visual appearance and formatting** of web pages. It allows developers to control layout, colors, fonts, and more across multiple pages of a website with consistency and efficiency.

Analogi pondasi website



HTML element structure



HTML document anatomy

a minimal basic HTML5 standard template

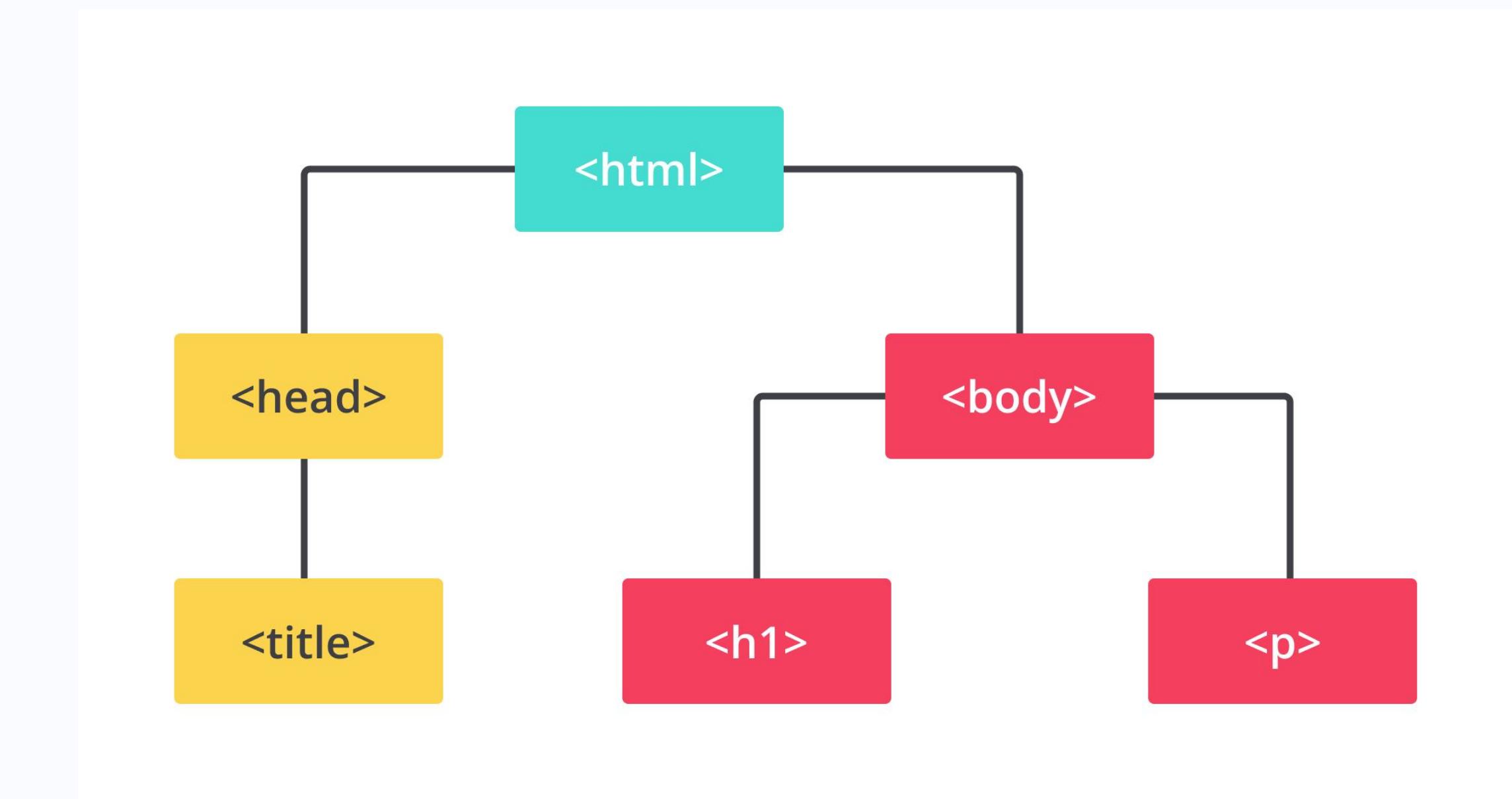
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Your Page Title</title>
6 </head>
7 <style>
8
9 </style>
10 <body>
11
12 </body>
13 </html>
```

DOM Tree

a hierarchical representation of each node in the tree corresponds to a part of the document

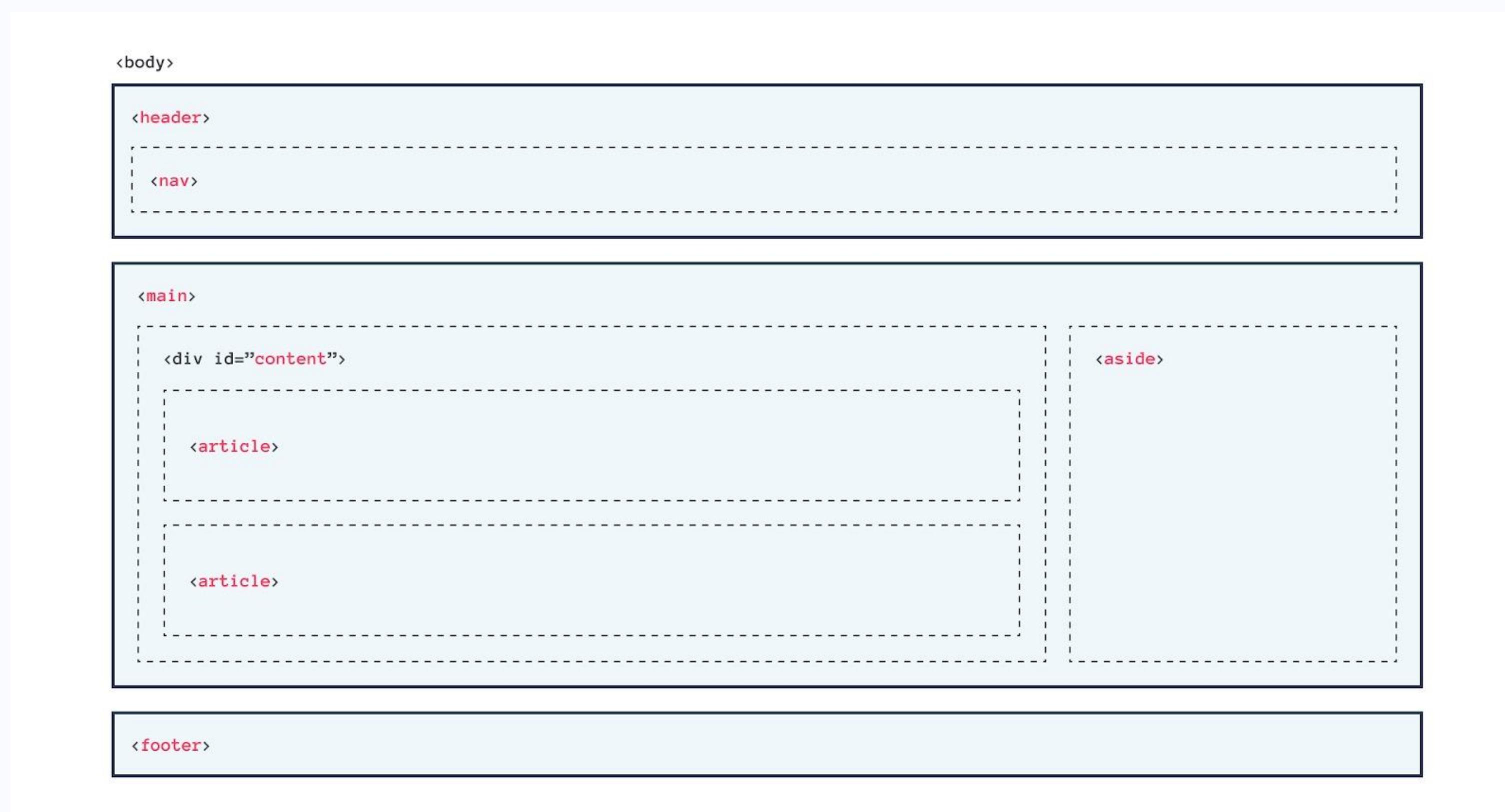
- ● ●

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Your Page Title</title>
5 </head>
6 <body>
7   <h1>Title</h1>
8   <p>This is a paragraph</p>
9 </body>
10 </html>
```



Semantic HTML

uses specific tags that convey the meaning and structure of the web content, improving accessibility and SEO



CSS Structure



Apply styling in HTML

External Style Sheet

Embedded Style Sheet

Inline Style

CSS Conceptions

Inheritance

Group selector

Rule order

Specificity

Child selector

Exercise

My Simple Blog
Welcome to my blog about web development and more!

First Post: Introduction to HTML
Posted on January 1, 2024

Understanding the basics of HTML.

This is the first post on my blog where I will talk about the basics of HTML and why it's crucial for creating websites.

Topics Covered:

- What is HTML?
- Basic HTML structure
- Important HTML tags

Copyright © 2024 by My Simple Blog.

Blog Creation Challenge: Basic HTML Website

Objective

Create a simple, static blog website using HTML5. The site should have a clean, readable layout and use semantic HTML tags appropriately.



Reference:
<https://share.aliepr.my.id/pweb3-lat>

Exercise

Requirements

HTML File:

- Create a single HTML file named blog.html.

Page Structure:

- Use the HTML5 doctype.
- Your page should include the following semantic elements: <header>, <footer>, <main>, and <article> tags.

Header:

- The <header> should contain the title of your blog and a brief introduction or tagline.

Main Content Area:

- Use the <main> tag to enclose the blog content.
- Inside <main>, include at least two <article> elements representing individual blog posts.
- Each <article> should have:
 - A headline wrapped in an <h2> tag.
 - A publication date in a <p> tag.
 - A main content paragraph.
 - A figure with an image and a <figcaption>. Ensure images have appropriate alt text.

List:

- Include at least one unordered () or ordered () list in one of the articles, summarizing key points or topics covered in the blog post.

Footer:

- The <footer> should contain copyright information.

Style (Optional):

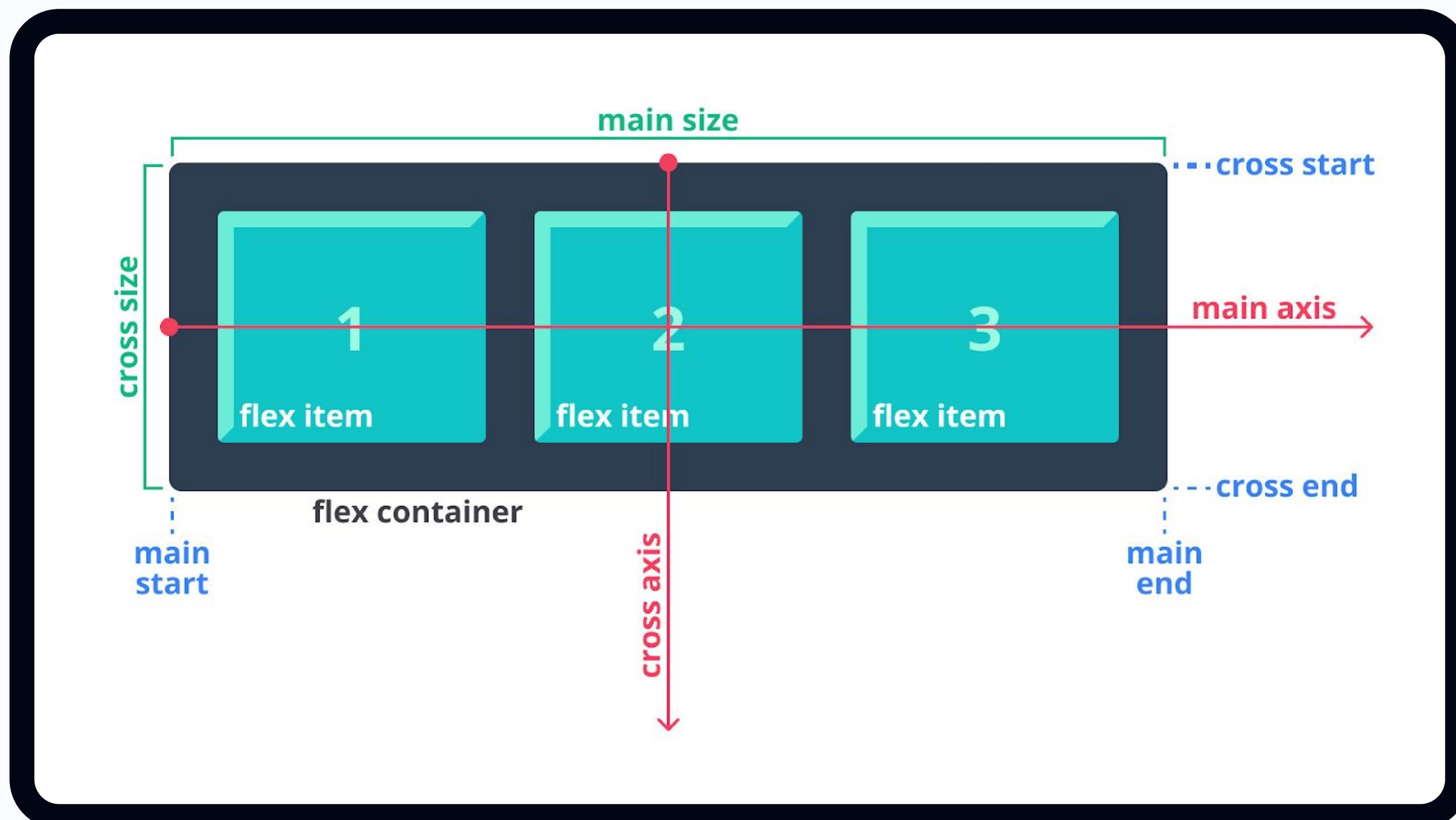
- While CSS styling is not required, you may add a simple internal or external stylesheet to improve the visual layout.

Images:

- Source at least two images relevant to your blog content. These should be used within your <figure> elements.

FlexBox Concept

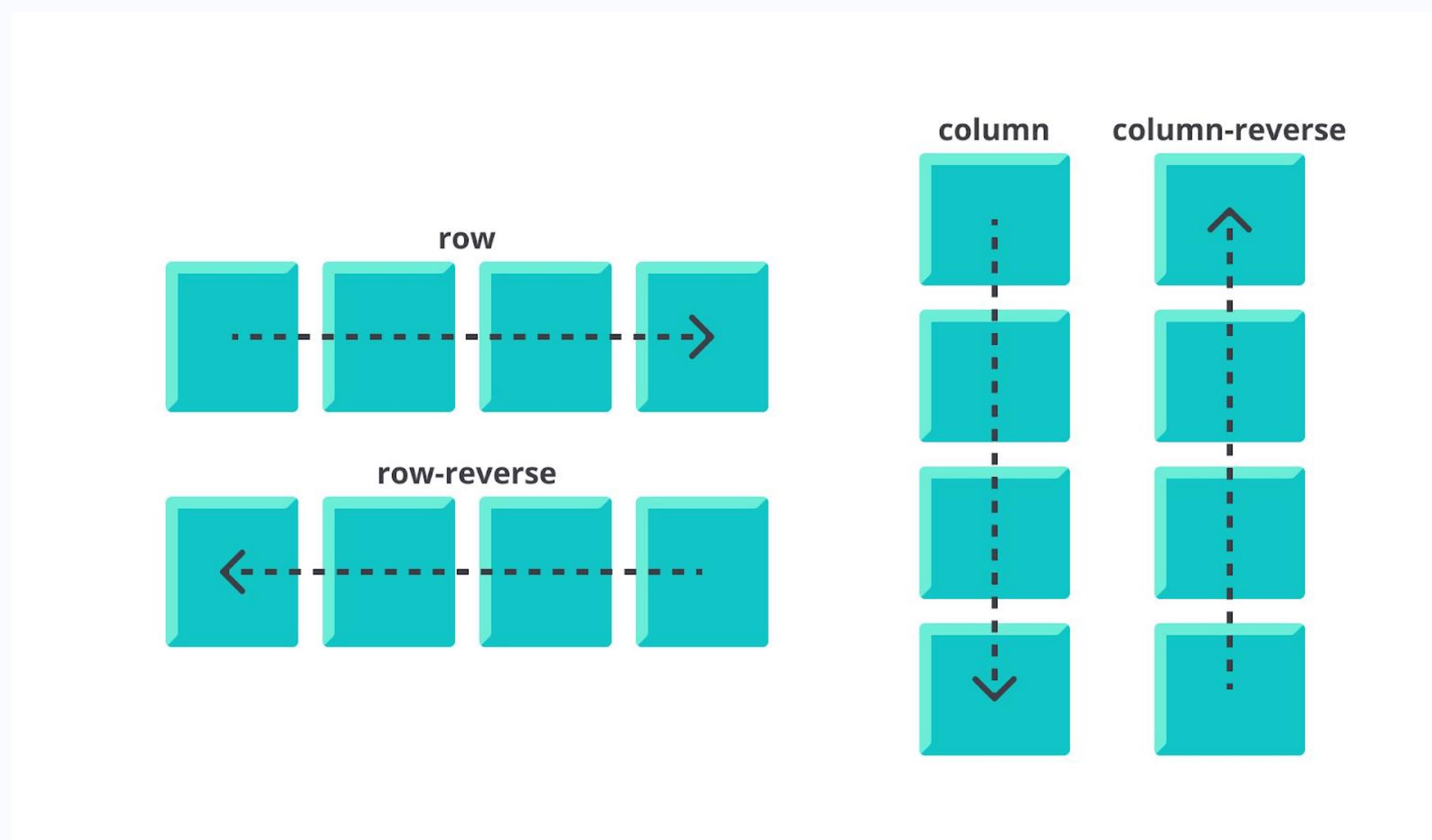
Control over the alignment, direction, and spacing of items within a container



```
1  div {  
2    display: flex;  
3  }  
4
```

Flex direction

specifies the direction in which the container's items are laid out, either as rows or columns



row* | row-reverse | column | column-reverse

Flex wrap

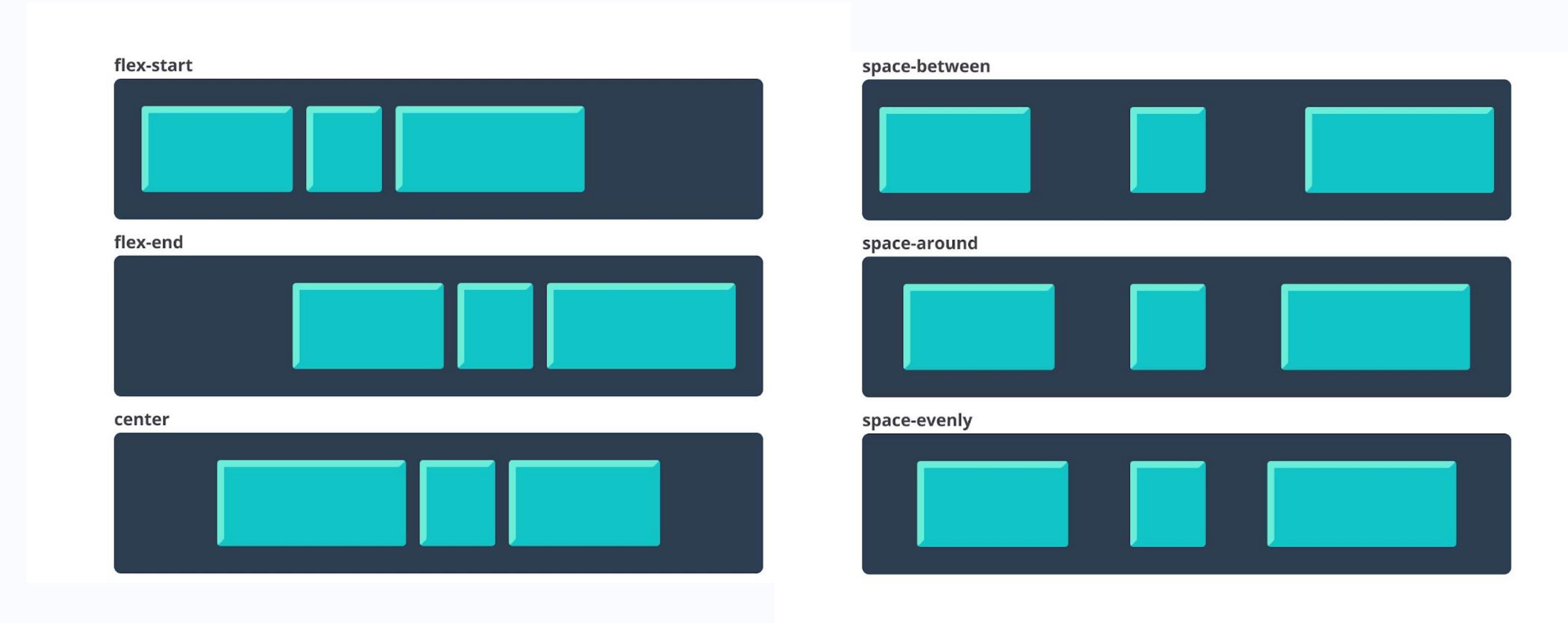
determines whether items within a flex container are forced into a single line or can wrap



nowrap* | wrap | wrap-reverse

Justify content

aligns items horizontally within a container

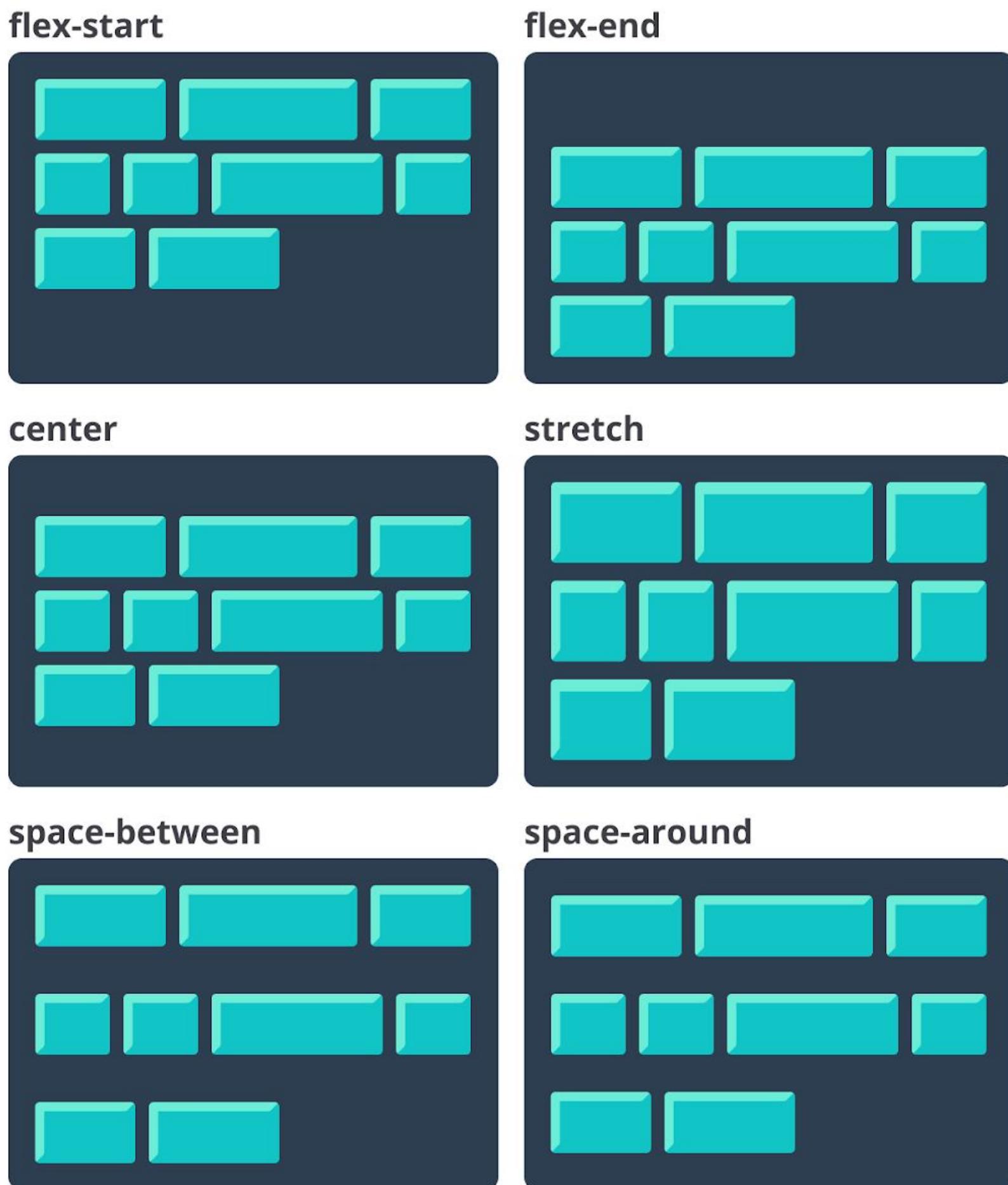


flex-start* | flex-end | center | space-between |
space-around | space-evenly

Align items

aligns items vertically within a container

flex-start* | flex-end | center | stretch |
space-between | space-around



Gap

set the gutters between rows and columns

<all-gap> | <row-gap> <col-gap>

gap: 10px



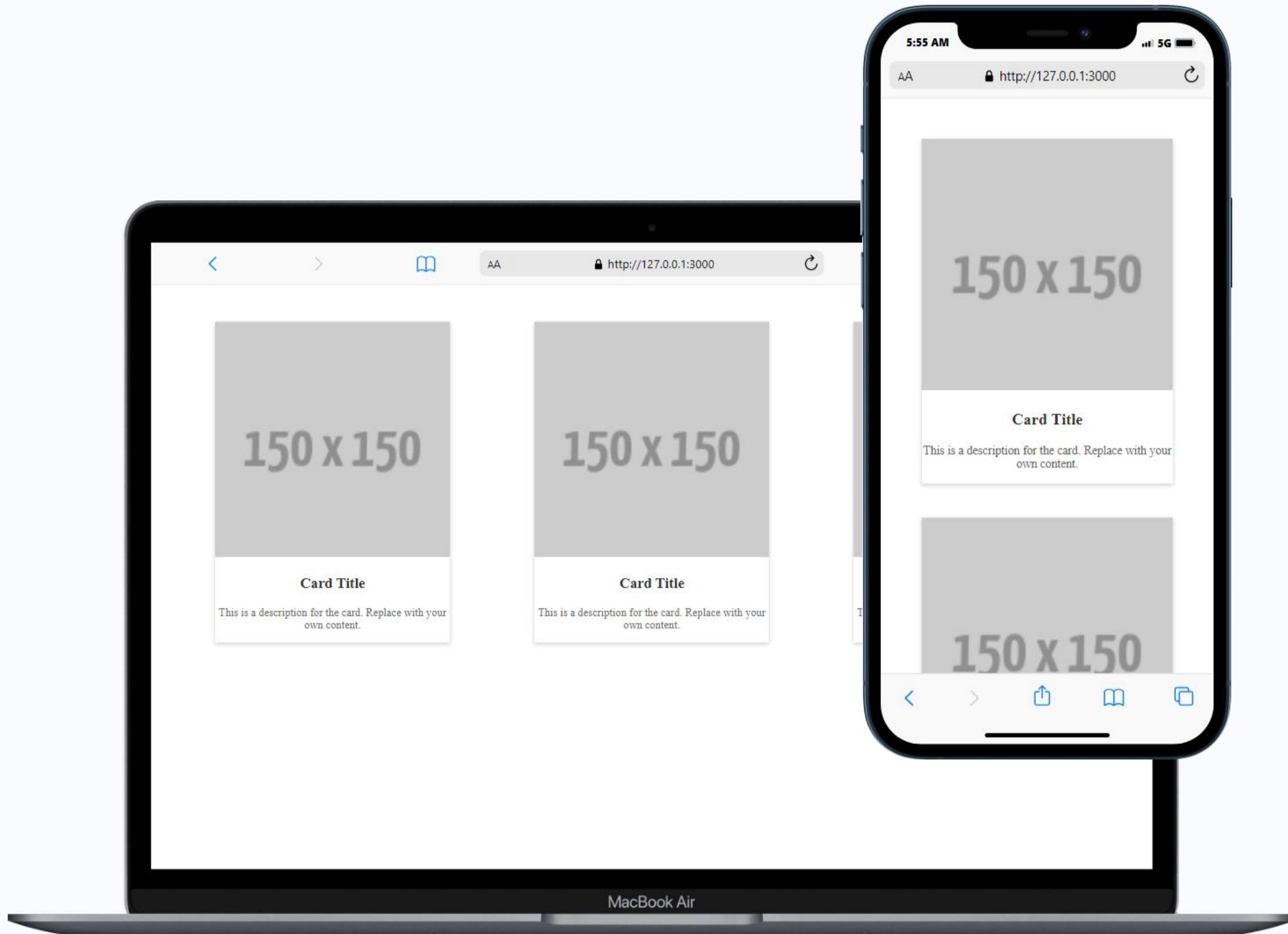
gap: 30px



gap: 10px 30px



Exercise



Flexbox Card Layout Challenge

Objective

Develop a responsive card layout using HTML and CSS Flexbox. Each card should include an image, a title, and a description



Reference:

<https://share.aliepr.my.id/pweb3-flexbox>

Exercise

Requirements

HTML and CSS Files:

- Create a single HTML file and a separate CSS file or include the CSS in a `<style>` tag within the HTML file.

Flexbox Container:

- Use a `<div>` element as a container for your cards and apply Flexbox styles to manage the layout.

Card Design:

- Each card should be a `<div>` element with class `card`.
- Include an `` element for the image, an `<h3>` for the title, and a `<p>` for the description.
- Ensure the image sources are valid or use placeholder images.

Responsive Layout:

- The card container should use `display: flex;`.
- Ensure the layout is responsive using `flex-wrap` to allow cards to wrap on smaller screens.
- Use `justify-content` to manage spacing between and around cards.

Styling:

- Cards should have a consistent width and margin.
- Apply a box-shadow to elevate the cards visually.
- Ensure text alignment and padding are aesthetically pleasing.

Accessibility:

- Ensure all images have appropriate alt attributes.
- Follow semantic HTML practices where possible.

Additional Features (Optional):

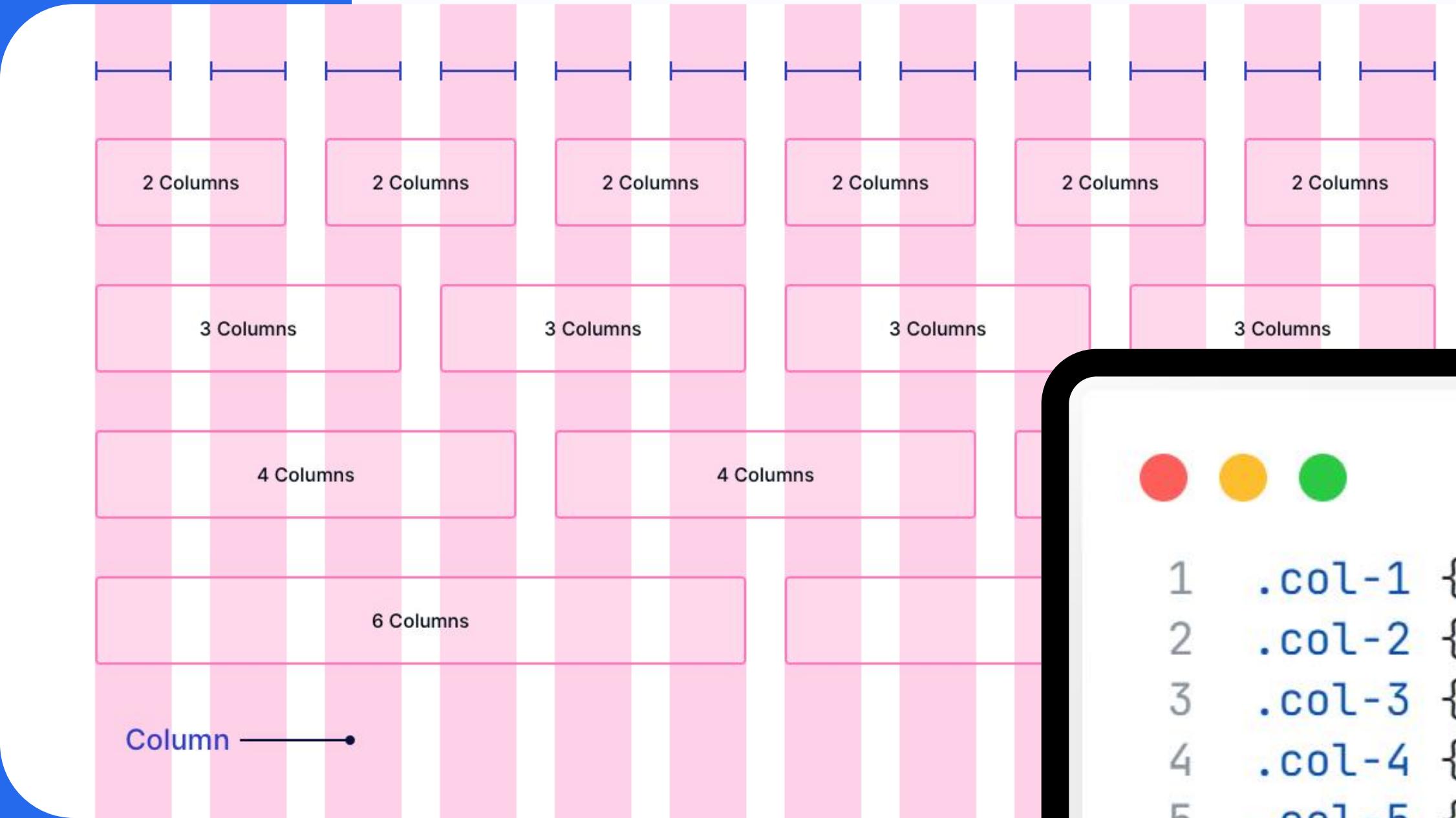
- Add hover effects to cards for interactivity.
- Implement a loading animation for images.
- Include a footer or header with additional information or navigation links.

Responsive Web Design Initialization



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

Responsive Grid View



```
1 .col-1 {width: 8.33%;}
2 .col-2 {width: 16.66%;}
3 .col-3 {width: 25%;}
4 .col-4 {width: 33.33%;}
5 .col-5 {width: 41.66%;}
6 .col-6 {width: 50%;}
7 .col-7 {width: 58.33%;}
8 .col-8 {width: 66.66%;}
9 .col-9 {width: 75%;}
10 .col-10 {width: 83.33%;}
11 .col-11 {width: 91.66%;}
12 .col-12 {width: 100%;}
13
```

CSS Media Query



```
1 /* Extra small devices (phones, 600px and down) */
2 @media only screen and (max-width: 600px) {}

3
4 /* Small devices (portrait tablets and large phones, 600px and up) */
5 @media only screen and (min-width: 600px) {}

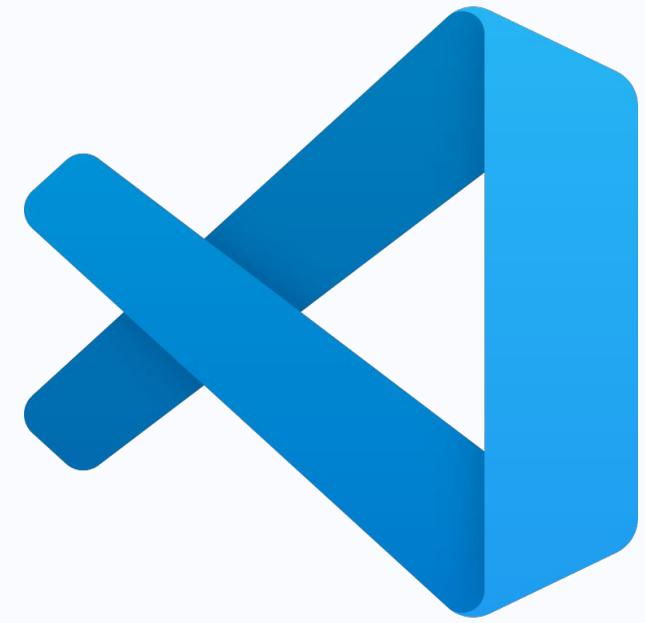
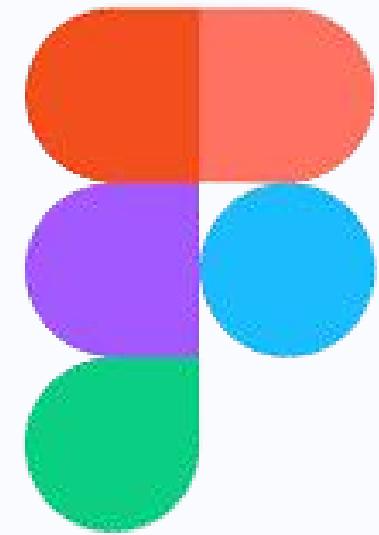
6
7 /* Medium devices (landscape tablets, 768px and up) */
8 @media only screen and (min-width: 768px) {}

9
10 /* Large devices (laptops/desktops, 992px and up) */
11 @media only screen and (min-width: 992px) {}

12
13 /* Extra large devices (large laptops and desktops, 1200px and up) */
14 @media only screen and (min-width: 1200px) {}
```



UTY



**Figma x VSCode
is amazing too...**

Pengembangan Web Praktik

Tahun ajaran 2024/2025