

Live Cohort

DAY - 18

HTML Semantics and CSS Grid Notes

HTML Semantics and CSS Grid Notes

1. HTML Semantics and Their Role

What is Semantic HTML?

Semantic HTML means using tags that describe their purpose clearly — instead of just `<div>` or ``.

It helps browsers, developers, and search engines understand your content better.

Example Tag:

`<header>`, `<main>`, `<article>`, `<footer>`, etc.

Why Semantic HTML Matters

1. ⚡ Improves Accessibility — Screen readers can interpret page structure easily.
2. 🔎 Better SEO — Search engines rank pages higher with meaningful markup.
3. 🧩 Clean Code — Easier to read and maintain.
4. 🧠 Better Understanding — Developers can quickly identify each section's purpose.

Common Semantic Tags

Tag	Description	Example Use
<code><header></code>	Top section of a page or article	Logo, navigation bar

HTML Semantics and CSS Grid Notes

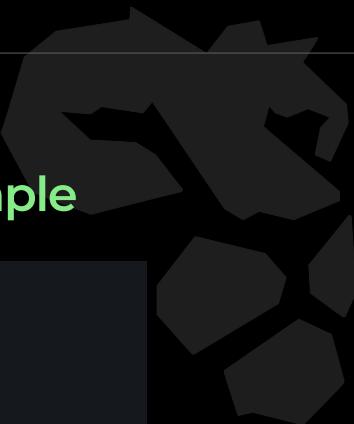
<nav>	Navigation section	Menus, sidebar links
<main>	Main unique content of page	Blog content
<section>	Groups related topics	Service sections
<article>	Independent content block	Blog posts, news
<aside>	Side information	Ads, related links
<footer>	Bottom section	Copyright info
<aside>	Side information	Ads, related links
<figure>	Media group (images, charts)	Image + caption

HTML Semantics and CSS Grid Notes

<figcaption>	Caption for a figure	Describes image
<time>	Represents date/time	Publish date
<address>	Contact info	Email, location

Semantic HTML Example

```
● ● ●
1 <header>
2   <h1>My Portfolio</h1>
3   <nav>
4     <a href="#about">About</a>
5     <a href="#projects">Projects</a>
6     <a href="#contact">Contact</a>
7   </nav>
8 </header>
9
10 <main>
11   <section id="about">
12     <h2>About Me</h2>
13     <p>I'm a frontend developer passionate about web design.</p>
14   </section>
15
16   <article id="project1">
17     <h2>Project One</h2>
18     <p>This project showcases my skills with React and CSS.</p>
19   </article>
20 </main>
21
22 <footer>
23   <p>© 2025 My Portfolio. All rights reserved.</p>
24 </footer>
25
```



💡 Real-life Use Case:

Used in blogs, news sites, or portfolios to create structure and improve SEO.

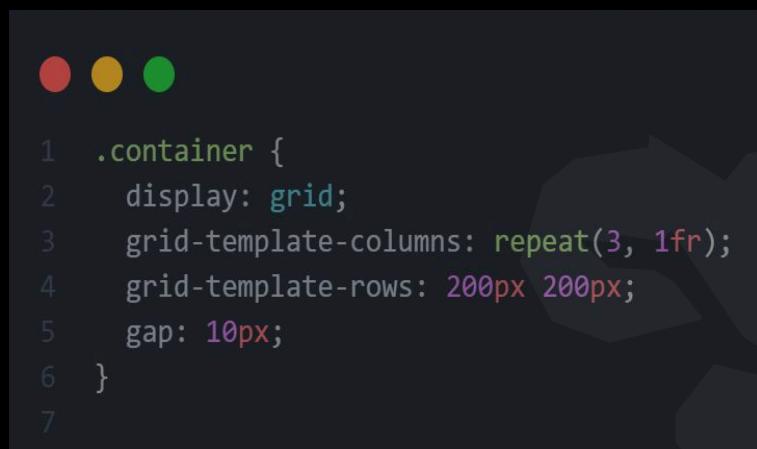
HTML Semantics and CSS Grid Notes

2. CSS Grid and Its Properties

What is CSS Grid?

CSS Grid is a 2D layout system that lets you create advanced page layouts using rows and columns — great for designing entire web pages.

Basic Grid Setup



```
1 .container {  
2   display: grid;  
3   grid-template-columns: repeat(3, 1fr);  
4   grid-template-rows: 200px 200px;  
5   gap: 10px;  
6 }  
7
```

Creates a 3-column, 2-row grid with 10px gap between boxes.

Common Grid Properties

Property	Description	Example
display: grid	Defines a grid container	display: grid;
grid-template-columns	Sets number and width of columns	repeat(3, 1fr)

HTML Semantics and CSS Grid Notes

grid-template-rows	Sets row heights	100px 200px
gap	Adds spacing between items	gap: 20px
justify-items	Aligns content horizontally	center
justify-content	Moves entire grid horizontally	space-around
align-content	Moves grid vertically	space-between
grid-auto-rows	Default height for implicit rows	minmax(100px, auto)
grid-auto-flow	Controls placement flow	row, column

3. Grid Template Areas — Visual Layout Design

Example

HTML Semantics and CSS Grid Notes



```
1 <div class="container">
2   <header>Header</header>
3   <nav>Navigation</nav>
4   <main>Main Content</main>
5   <aside>Sidebar</aside>
6   <footer>Footer</footer>
7 </div>
8
```



```
1 .container {
2   display: grid;
3   grid-template-areas:
4     'header header header'
5     'nav main aside'
6     'footer footer footer';
7   grid-template-columns: 1fr 2fr 1fr;
8   grid-template-rows: 80px auto 60px;
9   gap: 10px;
10 }
11
12 header { grid-area: header; background: lightblue; }
13 nav { grid-area: nav; background: lightgreen; }
14 main { grid-area: main; background: lightyellow; }
15 aside { grid-area: aside; background: lightpink; }
16 footer { grid-area: footer; background: lightgray; }
17
```

💡 Visually maps your webpage layout — just like a wireframe.

💡 Use Case: Ideal for full-page layouts like dashboards or portfolios.

HTML Semantics and CSS Grid Notes

4. Other Useful Grid Properties



```
1 .item1 {  
2   grid-column: 1 / 3; /* spans across two columns */  
3   grid-row: 1 / 2;    /* occupies first row */  
4 }  
5
```

Property	Description
place-items: center;	Shorthand for align-items + justify-items
place-content: center;	Shorthand for align-content + justify-content
minmax(200px, 1fr)	Sets flexible column size
repeat(4, 1fr)	Repeats grid pattern easily

HTML Semantics and CSS Grid Notes

🧠 Summary

Concept	Description	Use Case
Semantic HTML	Adds meaning and structure to content	SEO & Accessibility
CSS Grid	2D layout system for web design	Webpage and dashboard layouts
Grid Areas	Simplifies layout using named regions	Clean, visual arrangement
Combine Both	Semantic HTML + Grid	Professional, responsive, accessible web designs