

Internship Report – Frontend Dev(Week2)

Name: Zainab

Father Name: Assad Qayyum

Date: 4th July, 2025

Internship Domain: Front-end Intern

Task: CSS - Display, Positioning (static, relative, absolute, fixed)

Task Overview: (Day5)

Today's task (Day 5 of Week 2) in my front-end internship was focused on learning two essential layout properties in CSS: display and position. These properties help in controlling how elements appear and behave on the web page.

Content Covered:

CSS Display property:

- Block
- Inline
- inline-block
- none
- flex and grid

CSS Positioning property:

- Static
- Relative
- Absolute
- fixed

Introduction:

In web development, understanding how elements are placed and structured on a page is key to creating clean and responsive designs. CSS provides layout properties like display and position to help developers control how elements appear and interact.

1. CSS Display property:

The display property in CSS defines how an element is displayed in the document layout. Every HTML element has a default display value, but CSS allows us to change it to better control layout behavior.

Common display values:

a) Block:

- Takes the full width of its container.
- Starts on a new line.
- **Example:** <div>, <p>, <section>

b) inline

- It takes only as much width as its content.
- Stays on the same line.
- Width and height cannot be set.
- **Example:** , <a>

c) inline-block:

- Just like inline but allows width and height to be defined.
- Sits inline but behaves like a block.
- **Example:** Buttons, nav links

d) none

- Hides the element completely — it does not take up space on the page.
- **Example:** Hiding elements

e) flex

- Turn a container into a flexible box.
- Make arranging child elements in rows or columns easier and useful for responsive design.
- **Example:** Layouts with spacing

f) **grid**

- Allows content to be laid out in rows and columns.
- Useful for complex layouts like dashboards or cards.
- **Example:** Advanced layouts

Code Implementation:

```
display-css.html X position-css.html
display-css.html > html > head > style > .section-title
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>CSS Display Property Demo</title>
5   <style>
6     body {
7       font-family: Arial, sans-serif;
8       margin: 30px;
9       background-color: #f9f9f9;
10    }
11
12    h1 {
13      text-align: center;
14      color: #333;
15    }
16
17    .section {
18      background-color: #fff;
19      padding: 15px 20px;
20      border-radius: 10px;
21      margin: 20px 0;
22      box-shadow: 0 0 8px rgba(0,0,0,0.05);
23    }
24
25    .section-title {
26      font-size: 18px;
27      font-weight: bold;
28      margin-bottom: 5px;
29      color: #222;
30    }
```

Ln 29, Col 19 Spaces: 4

```
display-css.html X position-css.html
display-css.html > html > head > style > .section-title
2 <html>
3 <head>
5 <style>
25 .section-title {
30 }
31
32 .section-description {
33   font-size: 14px;
34   color: #555;
35   margin-bottom: 10px;
36 }
37
38 .block-example {
39   background-color: lightcoral;
40   display: block;
41   padding: 10px;
42   margin-bottom: 10px;
43   color: white;
44 }
45
46 .inline-example {
47   background-color: lightgreen;
48   display: inline;
49   padding: 5px;
50   margin-right: 10px;
51 }
52
53 .inline-block-example {
54   background-color: lightblue;
55   display: inline-block;
```

Ln 29, Col 19 Spaces: 4

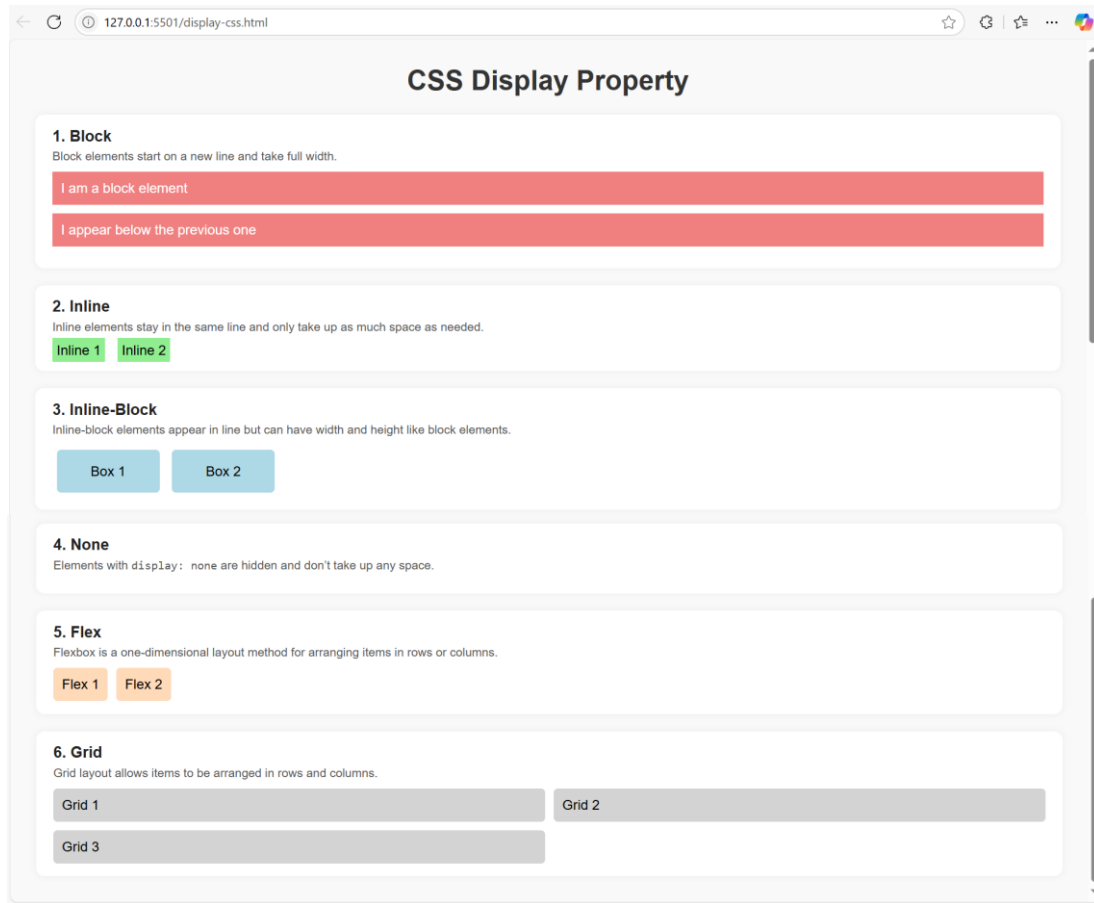
```
display-css.html X position-css.html
display-css.html > html > head > style > .section-title
2 <html>
3 <head>
5 <style>
53 .inline-block-example {
54   background-color: lightblue;
55   display: inline-block;
56   width: 120px;
57   height: 50px;
58   text-align: center;
59   line-height: 50px;
60   margin: 5px;
61   border-radius: 5px;
62 }
63
64 .none-example {
65   display: none;
66 }
67
68 .flex-container {
69   display: flex;
70   gap: 10px;
71 }
72
73 .flex-item {
74   background-color: peachpuff;
75   padding: 10px;
76   border-radius: 5px;
77 }
78
```

Ln 29, Col 19

```
display-css.html X position-css.html
display-css.html > html > body > div.section > div.grid-container > div.grid-item
2 <html>
3 <head>
5 <style>
79 .grid-container {
80   display: grid;
81   grid-template-columns: 1fr 1fr;
82   gap: 10px;
83 }
84
85 .grid-item {
86   background-color: lightgray;
87   padding: 10px;
88   border-radius: 5px;
89 }
90 </style>
91 </head>
92 <body>
93
94 <h1>CSS Display Property</h1>
95
96 <div class="section">
97   <div class="section-title">1. Block</div>
98   <div class="section-description">Block elements start on a new line and take full width.</div>
99   <div class="block-example">I am a block element</div>
100  <div class="block-example">I appear below the previous one</div>
101 </div>
102
103 <div class="section">
104   <div class="section-title">2. Inline</div>
```

```
display-css.html X position-css.html
display-css.html > html > body > div.section > div.grid-container > div.grid-item
2 <html>
92 <body>
103 <div class="section">
104   <div class="section-title">2. Inline</div>
105   <div class="section-description">Inline elements stay in the same line and only take up as much space as needed.</div>
106   <span class="inline-example">Inline 1</span>
107   <span class="inline-example">Inline 2</span>
108 </div>
109
110 <div class="section">
111   <div class="section-title">3. Inline-Block</div>
112   <div class="section-description">Inline-block elements appear in line but can have width and height like block el</div>
113   <div class="inline-block-example">Box 1</div>
114   <div class="inline-block-example">Box 2</div>
115 </div>
116
117 <div class="section">
118   <div class="section-title">4. None</div>
119   <div class="section-description">Elements with <code>display: none</code> are hidden and don't take up any space.</div>
120   <p class="none-example">You can't see me!</p>
121 </div>
122
123 <div class="section">
124   <div class="section-title">5. Flex</div>
125   <div class="section-description">Flexbox is a one-dimensional layout method for arranging items in rows or column</div>
126   <div class="flex-container">
127     <div class="flex-item">Flex 1</div>
128     <div class="flex-item">Flex 2</div>
129   </div>
```

```
display-css.html X position-css.html
display-css.html > html > body > div.section > div.grid-container > div.grid-item
2 <html>
92 <body>
122
123 <div class="section">
124   <div class="section-title">5. Flex</div>
125   <div class="section-description">Flexbox is a one-dimensional layout method for arranging items in rows or column</div>
126   <div class="flex-container">
127     <div class="flex-item">Flex 1</div>
128     <div class="flex-item">Flex 2</div>
129   </div>
130 </div>
131
132 <div class="section">
133   <div class="section-title">6. Grid</div>
134   <div class="section-description">Grid layout allows items to be arranged in rows and columns.</div>
135   <div class="grid-container">
136     <div class="grid-item">Grid 1</div>
137     <div class="grid-item">Grid 2</div>
138     <div class="grid-item">Grid 3</div>
139   </div>
140 </div>
141
142 </body>
143 </html>
```



2. CSS Positioning:

The position property in CSS defines how an element is positioned on a page and how it behaves when we apply top, left, right, or bottom properties. Different values of position help control whether the element stays in normal flow or is moved elsewhere.

Types of position:

a) static

- This is the default value.
- Elements follow the normal document flow.
- Cannot be moved using top/left/etc.

b) relative

- The element stays in its original position but can be moved relative to itself using top/left/right/bottom.
- The space it originally occupied remains.

c) absolute

- Removed from the normal flow.
- Positioned relative to the nearest positioned ancestor (parent with relative, absolute, or fixed).
- If no such parent exists, it is positioned relative to the <body>.

d) fixed

- Removed from flow like absolute, but is positioned relative to the browser window.
- It stays fixed even when the page is scrolled (e.g., sticky footer or "Back to Top" button).

Code Implementation:

```
position-css.html X
position-css.html > ...
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>CSS Position Property Demo</title>
5   <style>
6     body {
7       font-family: Arial, sans-serif;
8       padding: 20px;
9       background-color: #1e1e1e;
10      color: #f1f1f1;
11    }
12
13    h1 {
14      text-align: center;
15      color: #00d8ff;
16      margin-bottom: 30px;
17    }
18
19    .section {
20      background-color: #2b2b2b;
21      padding: 15px 20px;
22      border-radius: 8px;
23      margin-bottom: 15px;
24      position: relative;
25      font-size: 14px;
26    }
27
28    .section-title {
29      font-weight: bold;
30      margin-bottom: 5px;
```

```
display-css.html position-css.html X
position-css.html > ...
2 <html>
3 <head>
5 <style>

28 .section-title {
29     font-weight: bold;
30     margin-bottom: 5px;
31     color: #ffd700;
32 }
33
34 .box {
35     padding: 8px 12px;
36     margin-top: 8px;
37     border-radius: 4px;
38     width: fit-content;
39 }
40
41 .static-box {
42     background-color: #555;
43 }
44
45 .relative-box {
46     position: relative;
47     top: 10px;
48     left: 15px;
49     background-color: #4e8cff;
50 }
51
52 .absolute-container {
53     position: relative;
54     background-color: #333;
```

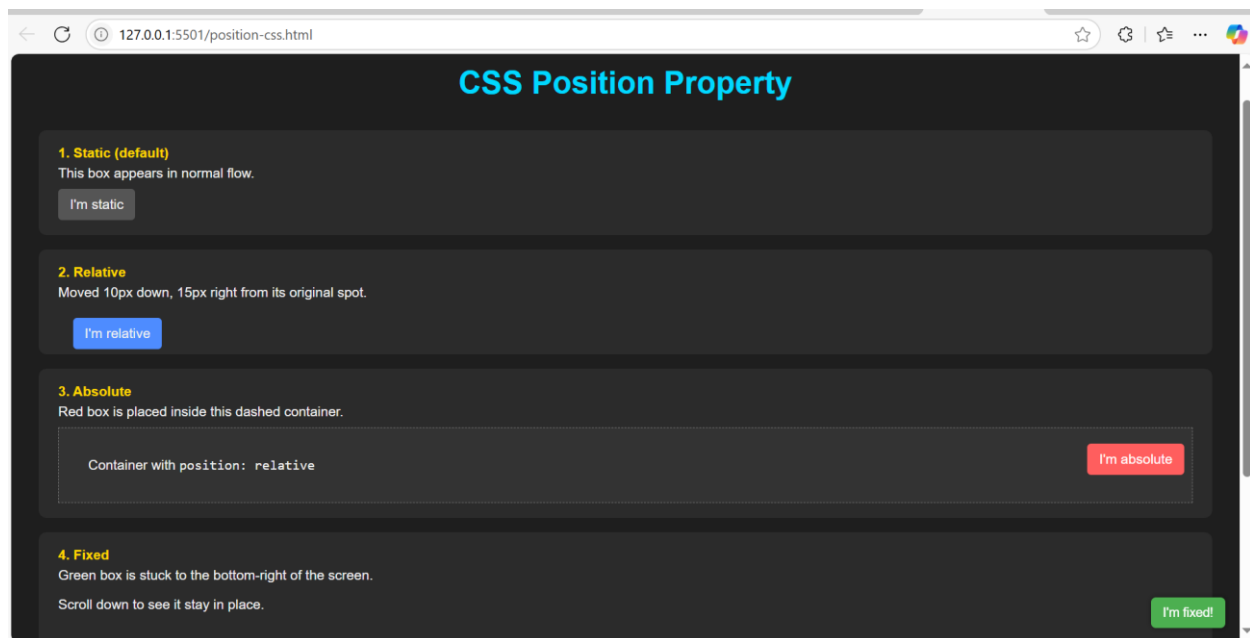
```
display-css.html position-css.html X
position-css.html > ...
2 <html>
3 <head>
5 <style>

52 .absolute-container {
53     position: relative;
54     background-color: #333;
55     padding: 30px;
56     border: 1px dashed #777;
57     margin-top: 8px;
58 }
59
60 .absolute-box {
61     position: absolute;
62     top: 8px;
63     right: 8px;
64     background-color: #ff5e5e;
65     color: white;
66 }
67
68 .fixed-box {
69     position: fixed;
70     bottom: 15px;
71     right: 15px;
72     background-color: #4caf50;
73     color: white;
74     padding: 8px 12px;
75     border-radius: 5px;
76     font-size: 13px;
77     box-shadow: 0 0 5px rgba(0,0,0,0.4);
78     z-index: 100;
```

Ln 122, Col 1 Spaces: 4 UTF-8 CRLF {} HTML


```
display-css.html position-css.html X
position-css.html > ...
2 <html>
3 <head>
5 <style>
68 .fixed-box {
75     border-radius: 5px;
76     font-size: 13px;
77     box-shadow: 0 0 5px rgba(0,0,0,0.4);
78     z-index: 100;
79 }
80 </style>
81 </head>
82 <body>
83
84 <h1>CSS Position Property</h1>
85
86 <!-- Static -->
87 <div class="section">
88     <div class="section-title">1. Static (default)</div>
89     <div>This box appears in normal flow.</div>
90     <div class="box static-box">I'm static</div>
91 </div>
92
93 <!-- Relative -->
94 <div class="section">
95     <div class="section-title">2. Relative</div>
96     <div>Moved 10px down, 15px right from its original spot.</div>
97     <div class="box relative-box">I'm relative</div>
98 </div>
99
100 <!-- Absolute -->
```

```
display-css.html position-css.html X
position-css.html > ...
2 <html>
82 <body>
99
100 <!-- Absolute -->
101 <div class="section">
102     <div class="section-title">3. Absolute</div>
103     <div>Red box is placed inside this dashed container.</div>
104     <div class="absolute-container">
105         <div class="box absolute-box">I'm absolute</div>
106         Container with <code>position: relative</code>
107     </div>
108 </div>
109
110 <!-- Fixed -->
111 <div class="section">
112     <div class="section-title">4. Fixed</div>
113     <div>Green box is stuck to the bottom-right of the screen.</div>
114     <p style="margin-bottom: 200px;">Scroll down to see it stay in place.</p>
115 </div>
116
117 <!-- Fixed floating box -->
118 <div class="fixed-box">I'm fixed!</div>
119
120 </body>
121 </html>
122 |
```



Conclusion:

In short, today's task helped me understand two foundational layout tools in CSS: display and position. I learned the behavior of each value, how and when to use them, and practiced them with simple layout examples. Mastering these properties will help me build better, responsive, and structured web pages as I move forward in front-end development.