



CSS Properties And Selectors

Software Development Bootcamp



Topic

CSS Properties



What Are Properties?

CSS **properties** are declarations that specify how selected HTML elements should be styled or formatted.

Each property is assigned a value, which determines the specific effect applied to the selected elements.



Display Property

Elements have a display value determining their layout and interaction with other elements.

By default, most elements are **inline** or **block**.



Inline Elements

Inline elements, flow within the text content, don't start on a new line, and only take up as much width as necessary.

- `<a>`
- ``
- ``
- ``
- `
`
- `<input>`



Block Elements

Block elements create distinct "blocks" in the layout, spanning the full width of their container and stacking vertically

- `<div>`
- `<p>`
- `<h1>` to `<h6>`
- `<table>`
- `<form>`
- `<header>`



Position Property

The position property in CSS determines how an element is positioned within its containing element or the document flow.

It specifies the positioning method used for an element.

- Static
- Relative
- Absolute
- Fixed



Static Positioning

“Static stays”

Elements are positioned according to the normal flow of the document.

```
element {  
    position: static;  
}
```




Relative Positioning

“Relative to itself”

Elements are positioned relative to their normal position.

```
element {  
    position: relative;  
    top: 10px;  
    left: 20px;  
}
```



Absolute Positioning

“Absolute to the nearest ancestor”

Elements are positioned relative to the nearest positioned ancestor.

```
element {  
    position: absolute;  
    top: 10px;  
    left: 20px;  
}
```



Fixed Positioning

“Fixed stays on screen”

Elements are positioned relative to the viewport and do not move when scrolled.

```
element {  
    position: fixed;  
    top: 10px;  
    left: 20px;  
}
```



Exercise

Hello Styles



Topic

CSS Selectors



What Are Selectors?

A CSS selector is a pattern used to select and style specific HTML elements on a web page.

They range from simple element names to complex combinations that allow precise targeting of elements for styling.



Element Selector

The element selector targets HTML elements by their tag name, and applies styles to all instances of the specified element within a document.

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

```
/* Changes ALL <p> tags  
to font color blue and  
font size 16px */
```



Class Selector ‘.’

Targets HTML elements by their class attribute to apply styles to elements with a specific class name.

- Class selectors are useful for applying the same style to multiple elements that share a specific class.

```
.highlight {  
    background-color:  
yellow;  
    font-weight: bold;  
}
```

```
/* The elements with the  
class `highlight` will  
have a yellow background  
and bold text. */
```


ID Selector ' # '

Targets a single HTML element by its unique ID attribute.

```
#header {  
    background-color: blue;  
    color: white;  
    padding: 20px;  
}  
  
/* In this example the  
element with the ID `header`  
will have a blue background,  
white text, and padding of  
20px */
```



Wildcard Selector

Used to apply a common style to all elements

```
* {  
    margin: 0;  
    padding: 0;  
}
```

```
/* In this example the  
wildcard selector resets the  
margin and padding for all  
elements, ensuring a  
consistent starting point  
for further styling. */
```



Order of Specificity

Selectors have an order of specificity: how broad to narrow your element selection will be.

Element selectors

Class selectors

ID selectors

Inline CSS



Pseudo-Class Selectors

CSS pseudo-class selectors are special keywords you can add to CSS rules to style elements in specific situations or states.

- Pseudo-class selectors help you style elements base on user interactions or their position in the HTML without needing extra classes or IDs.
 - Full list on [MDN](#)



Pseudo-Class Selectors

```
/* Changes the style when you move your mouse over an element */  
a:hover {background-color: red;}
```

```
/* Changes the style of elements that are disabled */  
a:disabled {background-color: gray;}
```

```
/* Changes the style when you click on an element */  
a:active {background-color: green;}
```



Exercise

Floating Images