

Web Development with HTML&CSS: Publish Your Own Website in a Week!

Heng Y.

Schoolhouse.world

29 August 2024

What we covered yesterday

- ▶ Basics of CSS
- ▶ Types of CSS styles
- ▶ Identification tags in HTML
- ▶ CSS selectors
- ▶ Cascading styles

What we will cover today

- ▶ Text settings
- ▶ The box model and sizing
- ▶ Basic CSS positioning
- ▶ Flexbox positioning

Get ready

- ▶ From the last 3 lessons, you should now have a basic webpage with HTML/CSS developed in JSFiddle.
- ▶ Please re-open your JSFiddle project. If you are unable to find it, click your user avatar in the top right corner and click "Your fiddles."

Text alignment

- ▶ `text-align` property controls the positioning of text within the element
- ▶ Values include `left`, `right`, `center`, and `justify`
- ▶ Recall that `<p>` and other text elements are block elements so they span the entire page
- ▶ Therefore, `text-align: center;` will center the text in the middle of the page by default

Text decorations

- ▶ The `text-decoration` property controls how text is styled.
- ▶ Values include `overline`, `line-through`, `underline`, and `none`
- ▶ This can be used to remove default styling of links for aesthetic purposes

Text transform

- ▶ The `text-transform` property controls the letters of the text.
- ▶ Values include `uppercase`, `lowercase`, `capitalize`;
- ▶ CSS properties do not change semantic meaning of text

Text spacing settings

- ▶ All these settings take a value in the length units (px, em, etc)
- ▶ `text-indent`
- ▶ `letter-spacing`
- ▶ `line-height`
- ▶ `word-spacing`

The box model

- ▶ Every element in CSS has a box around it
- ▶ This box is used to manipulate sizing

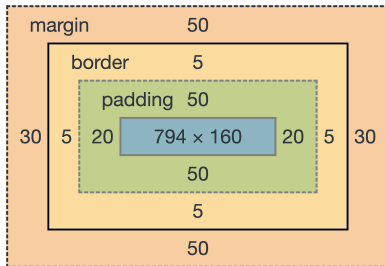


Figure: CSS box model

Border radius

- ▶ If we add a background color to an element, we can clearly see the basic box.
- ▶ We can add border radius to elements which will be applied to the box
- ▶ The `border-radius` property is used to do this
- ▶ It takes a number (usually in px) to determine how "round" the element should be

Margins, borders, and padding

- ▶ Each of these properties can be applied with the directional suffixes or without
- ▶ For example, `margin-right` or `padding` are both valid
- ▶ Margin and padding take length values.

Changing the size of the box

- ▶ The `width` and `height` properties are used to set the sizes of each box
- ▶ For responsive design, absolute units (px) are not recommended here due to screen resizing
- ▶ Instead, use `em` or percentage units

Percentage units

- ▶ The % unit specifies length relative to the parent element
- ▶ The viewport is the visible area of the webpage
- ▶ The vw unit specifies length relative to the viewport's width
- ▶ The vh unit specifies length relative to the viewport's height

CSS layout basics

- ▶ Normal flow: HTML displayed in the order of the code
- ▶ We can control the inline or block nature of components
- ▶ Flexbox allows to create rows or columns

Non-normal flow

- ▶ Not all elements have to be in the normal flow
- ▶ We can place elements at other locations
- ▶ The `position` CSS property allows us to determine what layout is applied to the element
- ▶ The possible values of the `position` property are
 - ▶ `static` - default positioning
 - ▶ `relative`
 - ▶ `fixed`
 - ▶ `absolute`
 - ▶ `sticky`

Using relative positioning

- ▶ When `position: relative;` the element is positioned relative of its normal position
- ▶ We use the `left`, `right`, `top`, and `bottom` to determine how much the element is offset
- ▶ We may use any of the length units for these properties
- ▶ For example, if we want the element 5 pixels below its normal position and 3 pixels to the left of its normal position, we can use this example:

```
position: relative;  
top: 5px;  
left: 3px;
```


Using absolute and fixed positioning

- ▶ In absolute positioning, the element is placed relative to the page
- ▶ In fixed positioning, the element is placed relative to the viewport
- ▶ In both examples, the `left`, `right`, `top`, and `bottom` properties are used for positioning.
- ▶ The example show positioning an element that will always be in the bottom right corner of the page
- ▶ Note that 0 does not need units

```
position: fixed;  
bottom: 0;  
right: 0;
```

Using sticky positioning

- ▶ In sticky positioning, the element behaves as if it were in relative positioning until the user scrolls past the element
- ▶ After the user scrolls past, the element will behave as fixed positioning
- ▶ This example will be in normal position until the user scrolls past, then it will be stick to the top of the page

```
position: sticky;  
top: 0;
```

Z-Index

- ▶ When we use the position properties, we can make elements overlap
- ▶ The z-index property determines which element is on top
- ▶ The value is any number, higher values mean the element is on a higher layer
- ▶ If no z-index is specified, the element that shows up **last** in the HTML will be positioned highest in the layers

Flexbox

- ▶ Flexbox means "flexible box"
- ▶ Flexbox allows for easy multi-column and multi-row layouts and distribution of screen space according to proportions
- ▶ The first step is to set `display: flex` on the parent container (a block container)
- ▶ This will arrange the child elements in a row.

Flexbox parent options

- ▶ The property `flex-direction` allows to set if we are using rows or columns
- ▶ Possible values include `column`, `row`, `column-reverse`, and `row-reverse`
- ▶ The property `justify-content` allows to set where the children are aligned with regard to the main axis
- ▶ Possible values include `center`, `flex-start` (default), `flex-end`, and `space-between`

Flexbox child options

- ▶ The `flex` option is set on the child elements
- ▶ This allows us to set the proportion of the container that is taken up
- ▶ This is a relative number. For example to have one `<div>` be 2X larger, make one div be `flex: 1` and the other `flex: 2`
- ▶ In the default value `initial` the element will take only as much room as it needs rather than growing or shrinking

Thank you!

- ▶ Tomorrow, we will finish up HTML and CSS and learn how to deploy our website.
- ▶ Please let me know if you have a GitHub account. If not, we will make one tomorrow together.
- ▶ Please leave feedback to let me know how I'm doing!