

CSS Opacity

CSS Image Opacity

The opacity property specifies the opacity/transparency of an element.

The opacity property can take a value from 0.0 - 1.0:

- 0.0 - The element will be completely transparent
- 0.5 - The element will be 50% transparent
- 1.0 - Default. The element will be fully opaque

Example

```
img {  
  opacity: 0.5;  
}
```

Opacity and :hover

The opacity property is often used with :hover to change the opacity on mouse-over:

Example

```
img {  
  opacity: 0.5;  
}
```

```
img:hover {  
  opacity: 1.0;  
}
```

Reversed Hover Effect

Here is an example of reversed hover effect:

Example

```
img:hover {  
  opacity: 0.5;  
}
```

Transparent Boxes

When using the opacity property to add transparency to the background of an element, all child elements inherit the same transparency. This can make the text inside a transparent element hard to read:

opacity 1

opacity 0.6

opacity 0.3

opacity 0.1

Example

```
div {  
  opacity: 0.3;  
}
```

CSS Navigation Bars

A vertical navbar:

- [Home](#)
- [News](#)
- [Contact](#)
- [About](#)

A horizontal navbar:

- [Home](#)

- [News](#)
 - [Contact](#)
 - [About](#)
-

CSS Navigation Bars (Intro)

Having an easy-to-use navigation is important for any website!

CSS navigation bars are an important component of web design. Navigation bars helps users to easily navigate between different sections of a website.

Navigation bars are typically built with HTML list elements (`` and ``), and then styled with CSS to get a great look.

Navigation bars are typically located at the top or at the side of a webpage.

Navigation Bar = List of Links

A navigation bar needs standard HTML as a base.

A navigation bar is basically a list of links, so using the `` and `` elements makes perfect sense:

Example

```
<ul>
  <li><a href="default.asp">Home</a></li>
  <li><a href="news.asp">News</a></li>
  <li><a href="contact.asp">Contact</a></li>
  <li><a href="about.asp">About</a></li>
</ul>
```

Now let's remove the bullets and the margins and padding from the `` element:

Example

```
ul {
  list-style-type: none;
  margin: 0;
```

```
padding: 0;
}
```

Vertical Nav Bar

CSS Vertical Navigation Bar

In a vertical navigation bar, the navigation links are stacked vertically (on top of each other), and is typically aligned along the left or right side of a webpage.

CSS Vertical Navbar Example

Here, we create a basic vertical navigation bar with a gray background color, and we also change the background color and the text color of the links when the user mouse over them:

Example

```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 200px;
  background-color: #f1f1f1;
}
```

```
li a {
  display: block;
  color: black;
  padding: 8px 16px;
  text-decoration: none;
}
```

```
/* Change the link and background color on hover */
```

```
li a:hover {  
  background-color: #555555;  
  color: white;  
}
```

Active State

Now we add an "active" class to highlight the link corresponding to the current page to let the user know which page/section he/she is on:

Example

```
.active {  
  background-color: #04AA6D;  
  color: white;  
}
```

Center Navbar Links & Add Borders

To center the navbar links, add text-align: center; to `` or `<a>`.

If you want a border around the navbar, add the border property to ``.

If you also want borders inside the navbar, add a border-bottom property to all `` elements, except for the last one:

Example

```
ul {  
  border: 1px solid #555555;  
}  
  
li {  
  text-align: center;  
  border-bottom: 1px solid #555555;  
}
```

Horizontal Nav Bar

CSS Horizontal Navigation Bar

In a horizontal navigation bar, the navigation links are stacked horizontally (next to each other), and is typically aligned on the top of a webpage.

The basics of a horizontal navigation bar is an unordered list (), with list items (), each holding a link (<a>), as shown in the [Navbar Intro](#) page.

It is also common to add a <nav> element around the element, that will serve as a container for your navigation bar.

Horizontal Navbar Using Float

One way of creating a horizontal navigation bar is to add the [float](#) property to the elements.

Here we create a basic horizontal navigation bar with a dark background color and change the background color of the links when the user moves the mouse over them:

Example

Horizontal navbar with float:

```
ul {  
  list-style-type: none;  
  margin: 0;  
  padding: 0;  
  overflow: hidden;  
  background-color: #333333;  
}  
  
ul li {  
  float: left;  
}  
  
ul li a {
```

```
display: block;
color: white;
text-align: center;
padding: 14px 16px;
text-decoration: none;
}

ul li a:hover {
  background-color: #111111;
}
```

Horizontal Navbar Using Flex

A more modern way of creating a horizontal navigation bar, is to use CSS flexbox.

The following example creates an equal-looking horizontal navbar as above, but with flexbox:

Example

Horizontal navbar with flex:

```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  background-color: #333333;
  display: flex;
}

ul li a {
  display: block;
  color: white;
  padding: 14px 16px;
  text-decoration: none;
}

ul li a:hover {
```

```
background-color: #111111;  
}
```

The solution here are `display: flex;`. This creates a flex context and by default the items are shown from left to right.

And just by adding one line to the `ul` block: `justify-content: center;`, you will have a horizontally centered navbar:

Example

Horizontal centered navbar with flex:

```
ul {  
  list-style-type: none;  
  margin: 0;  
  padding: 0;  
  background-color: #333333;  
  display: flex;  
  justify-content: center;  
}
```
