

# CSS

Web Application Development

1

# SOURCE: W3SCHOOLS

The image shows a browser window with the URL <https://www.w3schools.com>. The page features a green navigation bar with 'TUTORIALS' and 'REFERENCES' dropdown menus. A sidebar on the left lists various topics, with 'Learn CSS' highlighted. A large orange arrow points from this menu item to a second browser window. This second window shows the URL <https://www.w3schools.com/css/default.asp>. The top navigation bar is dark grey with 'CSS' highlighted in green. A sidebar on the left lists CSS topics, with 'CSS HOME' highlighted in green. The main content area displays the title 'CSS Tutorial' and a green button labeled '< Home'. Below this, a green box contains the text: 'CSS is a language that describes the style of an HT' and 'CSS describes how HTML elements should be displa'.



# CSS OVERFLOW

Web Application Development

# CSS OVERFLOW

## Layout - Overflow

The CSS `overflow` property controls what happens to content that is too big to fit into an area.

This text is really long and the height of its container is only 100 pixels. Therefore, a scrollbar is added to help the reader to scroll the content. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_overflow\\_intro](https://www.w3schools.com/css/tryit.asp?filename=trycss_overflow_intro)

# CSS OVERFLOW

## CSS Overflow

The `overflow` property specifies whether to clip content or to add scrollbars when the content of an element is too big to fit in a specified area.

The `overflow` property has the following values:

- `visible` - Default. The overflow is not clipped. It renders outside the element's box
- `hidden` - The overflow is clipped, and the rest of the content will be invisible
- `scroll` - The overflow is clipped, but a scrollbar is added to see the rest of the content
- `auto` - If overflow is clipped, a scrollbar should be added to see the rest of the content

**Note:** The `overflow` property only works for block elements with a specified height.

**Note:** In OS X Lion (on Mac), scrollbars are hidden by default and only shown when being used (even though "`overflow:scroll`" is set).

# CSS OVERFLOW

overflow: visible

By default, the overflow is `visible`, meaning that it is not clipped and it renders outside the element's box:

```
div {  
  width: 200px;  
  height: 50px;  
  background-color: #eee;  
  overflow: visible;  
}
```

You can use the overflow property when you want to have better control of the layout. The overflow property specifies what happens if content overflows an element's box.

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_overflow\\_visible](https://www.w3schools.com/css/tryit.asp?filename=trycss_overflow_visible)

# CSS OVERFLOW

overflow: hidden

With the `hidden` value, the overflow is clipped, and the rest of the content is hidden:

```
div {  
    overflow: hidden;  
}
```

You can use the overflow property when you want to have better control of the layout. The overflow

Try it Yourself:

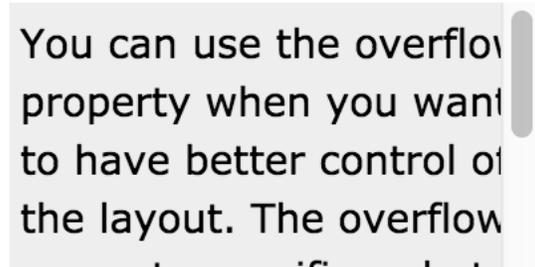
[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_overflow\\_hidden](https://www.w3schools.com/css/tryit.asp?filename=trycss_overflow_hidden)

# CSS OVERFLOW

overflow: scroll

Setting the value to `scroll`, the overflow is clipped and a scrollbar is added to scroll inside the box. Note that this will add a scrollbar both horizontally and vertically (even if you do not need it):

```
div {  
    overflow: scroll;  
}
```



You can use the overflow property when you want to have better control of the layout. The overflow

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_overflow\\_scroll](https://www.w3schools.com/css/tryit.asp?filename=trycss_overflow_scroll)

# CSS OVERFLOW

## overflow-x and overflow-y

The `overflow-x` and `overflow-y` properties specifies whether to change the overflow of content just horizontally or vertically (or both):

`overflow-x` specifies what to do with the left/right edges of the content.

`overflow-y` specifies what to do with the top/bottom edges of the content.

```
div {  
  overflow-x: hidden; /* Hide horizontal scrollbar */  
  overflow-y: scroll; /* Add vertical scrollbar */  
}
```

You can use the overflow property when you want to have better control of the layout. The overflow

**Try it Yourself:**

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_overflow\\_xy](https://www.w3schools.com/css/tryit.asp?filename=trycss_overflow_xy)

# CSS OVERFLOW

## overflow-x and overflow-y

The `overflow-x` and `overflow-y` properties specifies whether to change the overflow of content just horizontally or vertically (or both):

`overflow-x` specifies what to do with the left/right edges of the content.

`overflow-y` specifies what to do with the top/bottom edges of the content.

```
div {  
    overflow-x: hidden; /* Hide horizontal scrollbar */  
    overflow-y: scroll; /* Add vertical scrollbar */  
}
```

You can use the `overflow` property when you want to have better control of the layout. The `overflow`

**Try it Yourself:**

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_overflow\\_xy](https://www.w3schools.com/css/tryit.asp?filename=trycss_overflow_xy)

# CSS OVERFLOW

Test Yourself with Exercises!

- [Exercise 1](#)
- [Exercise 2](#)
- [Exercise 3](#)

# CSS OVERFLOW

## All CSS Overflow Properties

Property	Description
<a href="#"><u>overflow</u></a>	Specifies what happens if content overflows an element's box
<a href="#"><u>overflow-x</u></a>	Specifies what to do with the left/right edges of the content if it overflows the element's content area
<a href="#"><u>overflow-y</u></a>	Specifies what to do with the top/bottom edges of the content if it overflows the element's content area



# CSS FLOAT

Web Application Development

# CSS FLOAT

The CSS `float` property specifies how an element should float.

The CSS `clear` property specifies what elements can float beside the cleared element and on which side.

Float Left

Float Right

# CSS FLOAT

## The float Property

The `float` property is used for positioning and layout on web pages.

The `float` property can have one of the following values:

- `left` - The element floats to the left of its container
- `right`- The element floats to the right of its container
- `none` - The element does not float (will be displayed just where it occurs in the text). This is default
- `inherit` - The element inherits the float value of its parent

In its simplest use, the `float` property can be used to wrap text around images.

# CSS FLOAT

Example - float: right;

The following example specifies that an image should float to the **right** in a text:

```
img {  
    float: right;  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...



Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_layout\\_float](https://www.w3schools.com/css/tryit.asp?filename=trycss_layout_float)

# CSS FLOAT

Example - float: left;

The following example specifies that an image should float to the **left** in a text:

```
img {  
    float: left;  
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_layout\\_float2](https://www.w3schools.com/css/tryit.asp?filename=trycss_layout_float2)

# CSS FLOAT

## Example - No float

In the following example the image will be displayed just where it occurs in the text (float: none;):

```
img {  
    float: none;  
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus

imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_layout\\_float\\_none](https://www.w3schools.com/css/tryit.asp?filename=trycss_layout_float_none)

# CSS FLOAT

## Example - No float

In the following example the image will be displayed just where it occurs in the text (float: none;):

```
img {  
    float: none;  
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus

imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_layout\\_float\\_none](https://www.w3schools.com/css/tryit.asp?filename=trycss_layout_float_none)

# CSS FLOAT

## The clear Property

The `clear` property specifies what elements can float beside the cleared element and on which side.

The `clear` property can have one of the following values:

- `none` - Allows floating elements on both sides. This is default
- `left` - No floating elements allowed on the left side
- `right` - No floating elements allowed on the right side
- `both` - No floating elements allowed on either the left or the right side
- `inherit` - The element inherits the clear value of its parent

The most common way to use the `clear` property is after you have used a `float` property on an element.

When clearing floats, you should match the clear to the float: If an element is floated to the left, then you should clear to the left. Your floated element will continue to float, but the cleared element will appear below it on the web page.

The following example clears the float to the left. Means that no floating elements are allowed on the left side (of the div):

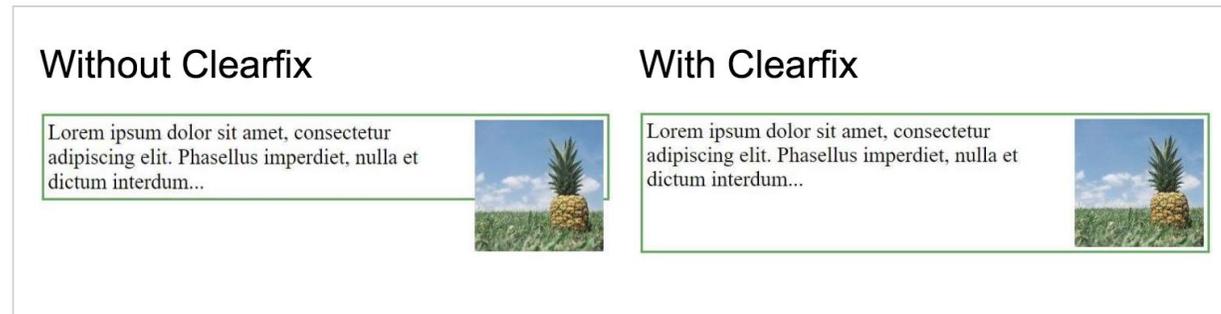
```
div {  
    clear: left;  
}
```

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_layout\\_clear](https://www.w3schools.com/css/tryit.asp?filename=trycss_layout_clear)

# CSS FLOAT

## The clearfix Hack

If an element is taller than the element containing it, and it is floated, it will "overflow" outside of its container:



Then we can add `overflow: auto;` to the containing element to fix this problem:

```
.clearfix {  
    overflow: auto;  
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_layout\\_clearfix](https://www.w3schools.com/css/tryit.asp?filename=trycss_layout_clearfix)

# CSS FLOAT

## The clearfix Hack Continued

The `overflow: auto` clearfix works well as long as you are able to keep control of your margins and padding (else you might see scrollbars). The **new, modern clearfix hack** however, is safer to use, and the following code is used for most webpages:

```
.clearfix::after {  
    content: "";  
    clear: both;  
    display: table;  
}
```

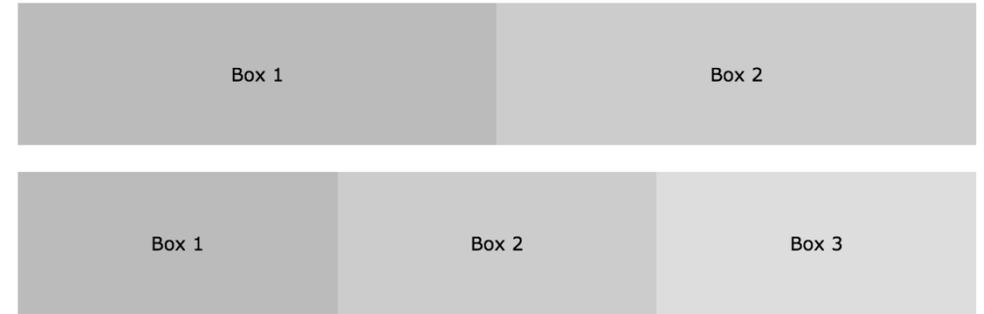
Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_layout\\_clearfix2](https://www.w3schools.com/css/tryit.asp?filename=trycss_layout_clearfix2)

You will learn more about the `::after` pseudo-element in a later slide.

# CSS FLOAT

## Grid of Boxes / Equal Width Boxes



With the `float` property, it is easy to float boxes of content side by side:

```
* {
  box-sizing: border-box;
}

.box {
  float: left;
  width: 33.33%; /* three boxes (use 25% for four, and 50% for two, etc) */
  padding: 50px; /* if you want space between the images */
}
```

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_float\\_boxes](https://www.w3schools.com/css/tryit.asp?filename=trycss_float_boxes)

### What is box-sizing?

You can easily create three floating boxes side by side. However, when you add something that enlarges the width of each box (e.g. padding or borders), the box will break. The `box-sizing` property allows us to include the padding and border in the box's total width (and height), making sure that the padding stays inside of the box and that it does not break.

You can read more about the `box-sizing` property in our [CSS Box Sizing Chapter](#).

# CSS FLOAT

## Images Side By Side

The grid of boxes can also be used to display images side by side:

```
.img-container {  
  float: left;  
  width: 33.33%; /* three containers (use 25% for four, and 50% for two,  
etc) */  
  padding: 5px; /* if you want space between the images */  
}
```



Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_float\\_images\\_side](https://www.w3schools.com/css/tryit.asp?filename=trycss_float_images_side)

# CSS FLOAT

## Equal Height Boxes

In the previous example, you learned how to float boxes side by side with an equal width. However, it is not easy to create floating boxes with equal heights. A quick fix however, is to set a fixed height, like in the example below:

```
.box {  
  height: 500px;  
}
```



Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss float boxes height](https://www.w3schools.com/css/tryit.asp?filename=trycss_float_boxes_height)

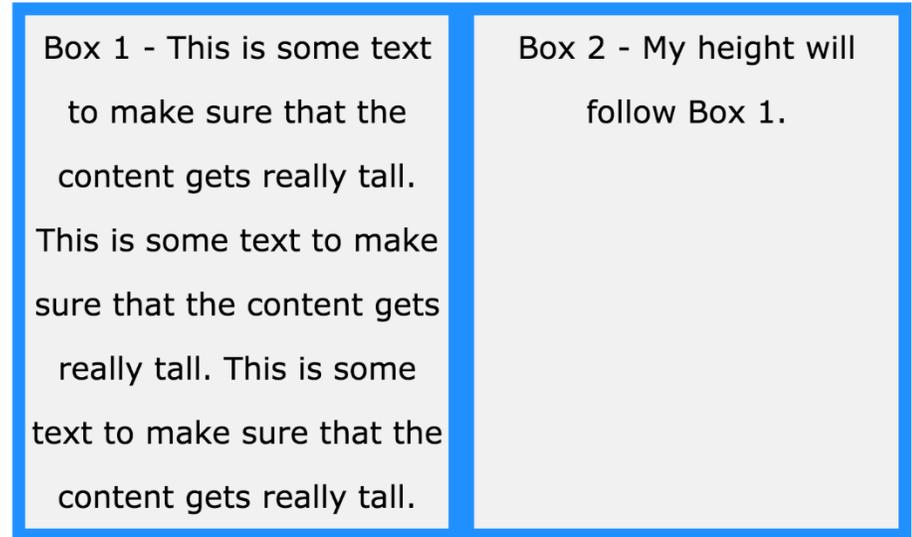
# CSS FLOAT

## Equal Height Boxes Continued

**However**, this is not very flexible. It is ok if you can guarantee that the boxes will always have the same amount of content in them. But many times, the content is not the same. If you try the example above on a mobile phone, you will see that the second box's content will be displayed outside of the box. This is where CSS3 Flexbox comes in handy - as it can automatically stretch boxes to be as long as the longest box.

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_float\\_boxes\\_flex](https://www.w3schools.com/css/tryit.asp?filename=trycss_float_boxes_flex)



The only problem with Flexbox is that it does not work in Internet Explorer 10 or earlier versions. You can read more about the Flexbox Layout Module in our [CSS Flexbox Chapter](#).

# CSS FLOAT

## Navigation Menu

Use `float` with a list of hyperlinks to create a horizontal menu:

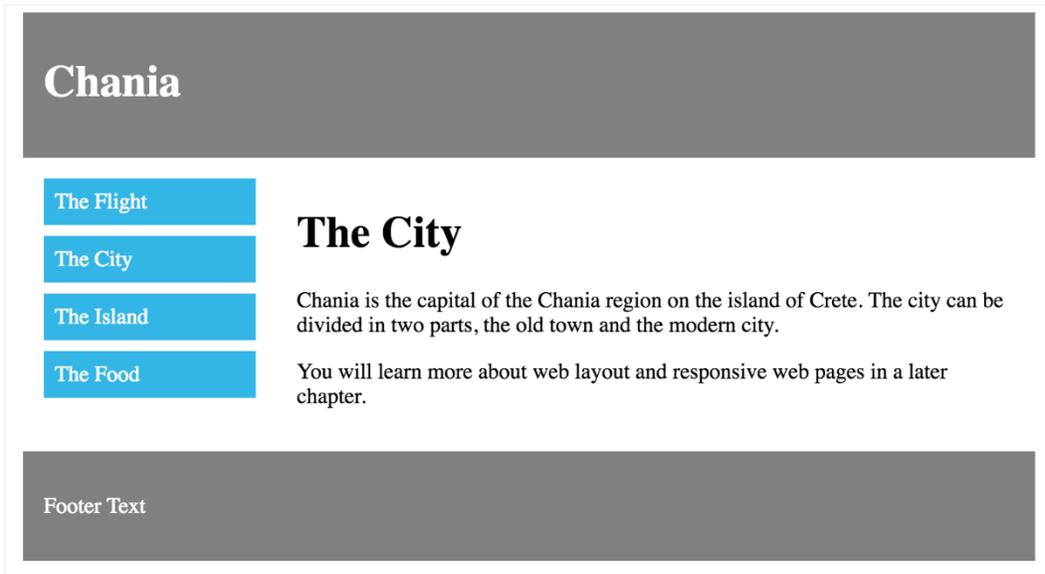


Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_float5](https://www.w3schools.com/css/tryit.asp?filename=trycss_float5)

# CSS FLOAT

## Web Layout Example

It is also common to do entire web layouts using the `float` property:



```
.header, .footer {
    background-color: grey;
    color: white;
    padding: 15px;
}

.column {
    float: left;
    padding: 15px;
}

.clearfix::after {
    content: "";
    clear: both;
    display: table;
}

.menu {
    width: 25%;
}

.content {
    width: 75%;
}
```

**Try it Yourself:**

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_layout\\_cols](https://www.w3schools.com/css/tryit.asp?filename=trycss_layout_cols)

# CSS FLOAT

## More Examples

- An image with border and margins that floats to the right in a paragraph  
Let an image float to the right in a paragraph. Add border and margins to the image.
- An image with a caption that floats to the right  
Let an image with a caption float to the right.
- Let the first letter of a paragraph float to the left  
Let the first letter of a paragraph float to the left and style the letter.
- Creating a website with float  
Use float to create a homepage with a navbar, header, footer, left content and main content.

# CSS FLOAT

## All CSS Float Properties

Property	Description
<a href="#"><u>box-sizing</u></a>	Defines how the width and height of an element are calculated: should they include padding and borders, or not
<a href="#"><u>clear</u></a>	Specifies what elements can float beside the cleared element and on which side
<a href="#"><u>float</u></a>	Specifies how an element should float
<a href="#"><u>overflow</u></a>	Specifies what happens if content overflows an element's box
<a href="#"><u>overflow-x</u></a>	Specifies what to do with the left/right edges of the content if it overflows the element's content area
<a href="#"><u>overflow-y</u></a>	Specifies what to do with the top/bottom edges of the content if it overflows the element's content area



# CSS INLINE-BLOCK

Web Application Development

# CSS INLINE-BLOCK

## The display: inline-block Value

Compared to `display: inline`, the major difference is that `display: inline-block` allows to set a width and height on the element.

Also, with `display: inline-block`, the top and bottom margins/paddings are respected, but with `display: inline` they are not.

Compared to `display: block`, the major difference is that `display: inline-block` does not add a line-break after the element, so the element can sit next to other elements.

The following example shows the different behavior of `display: inline`, `display: inline-block` and `display: block`:

```
span.a {
  display: inline; /* the default for span */
  width: 100px;
  height: 100px;
  padding: 5px;
  border: 1px solid blue;
  background-color: yellow;
}

span.b {
  display: inline-block;
  width: 100px;
  height: 100px;
  padding: 5px;
  border: 1px solid blue;
  background-color: yellow;
}

span.c {
  display: block;
  width: 100px;
  height: 100px;
  padding: 5px;
  border: 1px solid blue;
  background-color: yellow;
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_inline-block\\_span1](https://www.w3schools.com/css/tryit.asp?filename=trycss_inline-block_span1)

# CSS INLINE-BLOCK

## Using inline-block to Create Navigation Links

One common use for `display: inline-block` is to display list items horizontally instead of vertically. The following example creates horizontal navigation links:

```
.nav {
  background-color: yellow;
  list-style-type: none;
  text-align: center;
  padding: 0;
  margin: 0;
}

.nav li {
  display: inline-block;
  font-size: 20px;
  padding: 20px;
}
```

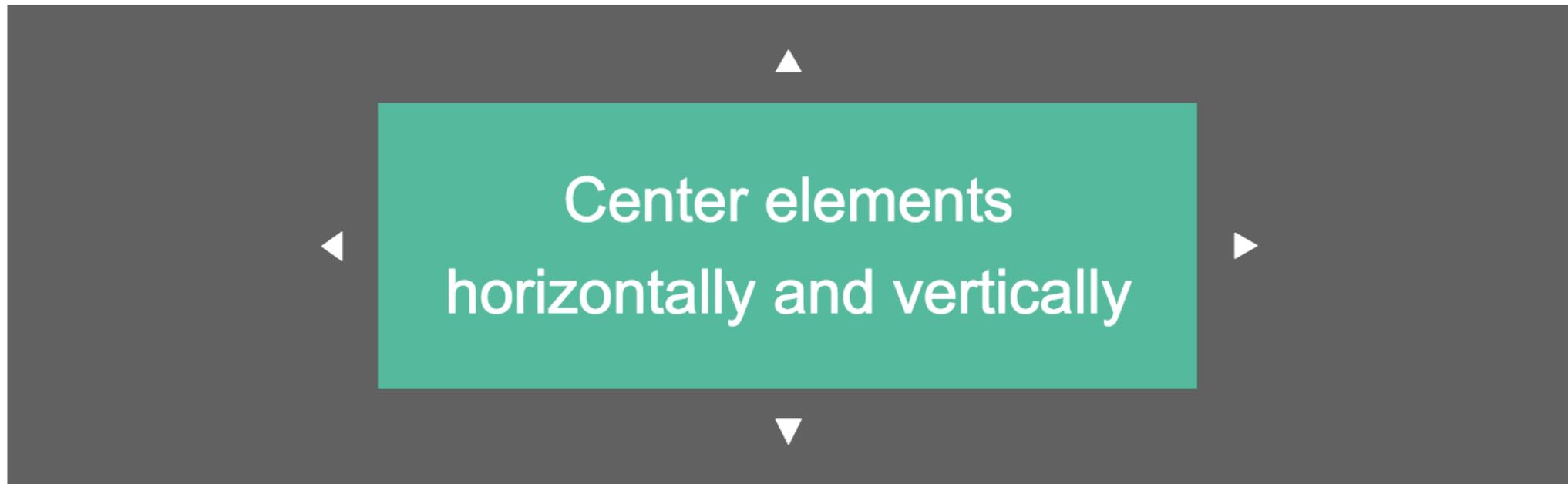
Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_inline-block\\_nav](https://www.w3schools.com/css/tryit.asp?filename=trycss_inline-block_nav)



# CSS ALIGN

Web Application Development

# CSS ALIGN



# CSS ALIGN

## Center Align Elements

To horizontally center a block element (like `<div>`), use `margin: auto;`

Setting the width of the element will prevent it from stretching out to the edges of its container.

The element will then take up the specified width, and the remaining space will be split equally between the two margins:

```
.center {  
    margin: auto;  
    width: 50%;  
    border: 3px solid green;  
    padding: 10px;  
}
```

**Try it Yourself:**

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_align\\_container](https://www.w3schools.com/css/tryit.asp?filename=trycss_align_container)

**Note:** Center aligning has no effect if the `width` property is not set (or set to 100%).

# CSS ALIGN

## Center Align Text

To just center the text inside an element, use `text-align: center;`



This text is centered.

```
.center {  
  text-align: center;  
  border: 3px solid green;  
}
```

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_align\\_text](https://www.w3schools.com/css/tryit.asp?filename=trycss_align_text)

**Tip:** For more examples on how to align text, see the [CSS Text](#) chapter.

# CSS ALIGN

## Center an Image

To center an image, set left and right margin to `auto` and make it into a `block` element:

```
img {  
    display: block;  
    margin-left: auto;  
    margin-right: auto;  
    width: 40%;  
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_align\\_image](https://www.w3schools.com/css/tryit.asp?filename=trycss_align_image)

# CSS ALIGN

Left and Right Align - Using position

One method for aligning elements is to use `position: absolute;`

```
.right {  
    position: absolute;  
    right: 0px;  
    width: 300px;  
    border: 3px solid #73AD21;  
    padding: 10px;  
}
```

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_align\\_pos](https://www.w3schools.com/css/tryit.asp?filename=trycss_align_pos)

**Note:** Absolute positioned elements are removed from the normal flow, and can overlap elements.

# CSS ALIGN

Left and Right Align - Using float

Another method for aligning elements is to use the `float` property:

```
.right {  
    float: right;  
    width: 300px;  
    border: 3px solid #73AD21;  
    padding: 10px;  
}
```

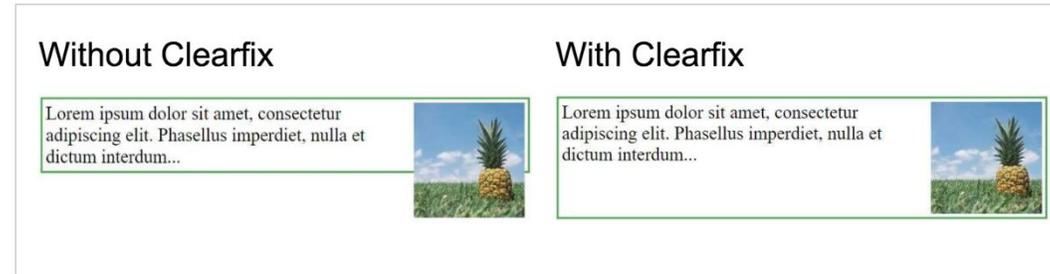
Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_align\\_float](https://www.w3schools.com/css/tryit.asp?filename=trycss_align_float)

**Note:** If an element is taller than the element containing it, and it is floated, it will overflow outside of its container. You can use the "**clearfix**" hack to fix this (see example on the next slide).

# CSS ALIGN

## The clearfix Hack



Then we can add `overflow: auto;` to the containing element to fix this problem:

```
.clearfix {  
    overflow: auto;  
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_layout\\_clearfix](https://www.w3schools.com/css/tryit.asp?filename=trycss_layout_clearfix)

# CSS ALIGN

## Center Vertically - Using padding

There are many ways to center an element vertically in CSS. A simple solution is to use top and bottom padding:

```
.center {  
    padding: 70px 0;  
    border: 3px solid green;  
}
```



I am vertically centered.

Try it Yourself:

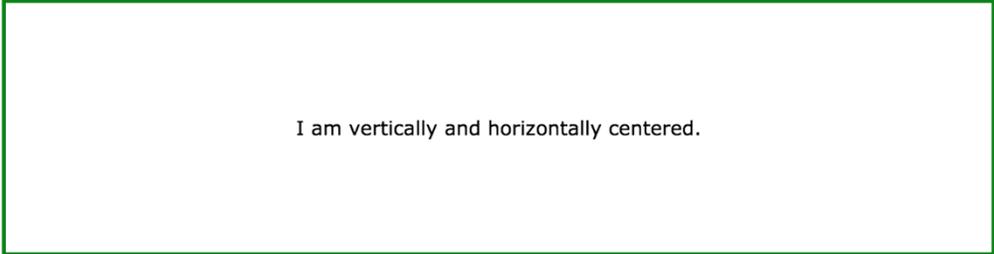
[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_align\\_padding](https://www.w3schools.com/css/tryit.asp?filename=trycss_align_padding)

# CSS ALIGN

## Center Vertically - Using padding Continued

To center both vertically and horizontally, use `padding` and `text-align: center`:

```
.center {  
    padding: 70px 0;  
    border: 3px solid green;  
    text-align: center;  
}
```



I am vertically and horizontally centered.

Try it Yourself:

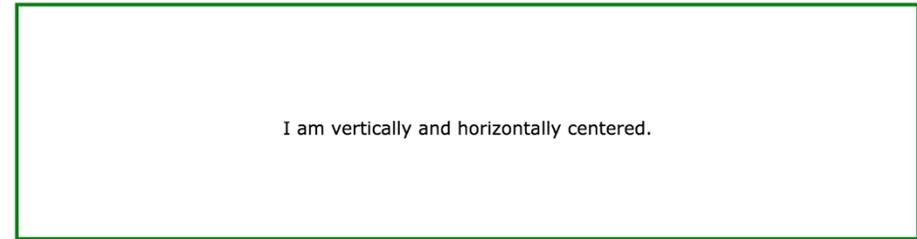
[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_align\\_padding2](https://www.w3schools.com/css/tryit.asp?filename=trycss_align_padding2)

# CSS ALIGN

## Center Vertically - Using line-height

Another trick is to use the `line-height` property with a value that is equal to the `height` property.

```
.center {  
  line-height: 200px;  
  height: 200px;  
  border: 3px solid green;  
  text-align: center;  
}
```



```
/* If the text has multiple lines, add the following: */  
.center p {  
  line-height: 1.5;  
  display: inline-block;  
  vertical-align: middle;  
}
```

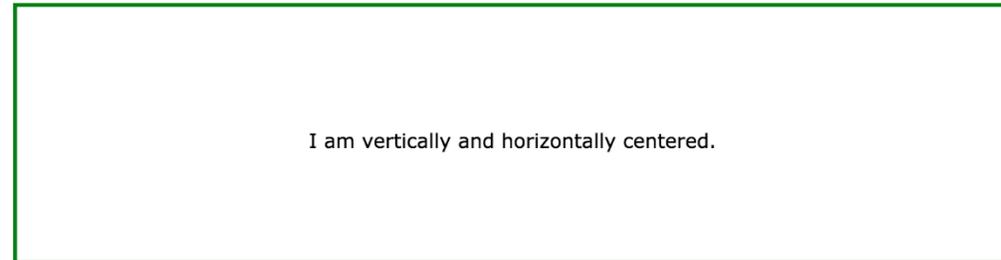
Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_align\\_line-height](https://www.w3schools.com/css/tryit.asp?filename=trycss_align_line-height)

# CSS ALIGN

## Center Vertically - Using position & transform

If padding and line-height are not options, a third solution is to use positioning and the transform property:

```
.center {  
  height: 200px;  
  position: relative;  
  border: 3px solid green;  
}  
  
.center p {  
  margin: 0;  
  position: absolute;  
  top: 50%;  
  left: 50%;  
  transform: translate(-50%, -50%);  
}
```



Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_align\\_transform](https://www.w3schools.com/css/tryit.asp?filename=trycss_align_transform)

**Tip:** You will learn more about the transform property in our [2D Transforms Chapter](#).

# CSS ALIGN

Test Yourself with Exercises!

- [Exercise 1](#)
- [Exercise 2](#)



# CSS COMBINATORS

Web Application Development

# CSS COMBINATORS

## CSS Combinators

A combinator is something that explains the relationship between the selectors.

A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator.

There are four different combinators in CSS:

- descendant selector (space)
- child selector (>)
- adjacent sibling selector (+)
- general sibling selector (~)

# CSS COMBINATORS

## Descendant Selector

The descendant selector matches all elements that are descendants of a specified element.

The following example selects all <p> elements inside <div> elements:

```
div p {  
    background-color: yellow;  
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_sel\\_element\\_element](https://www.w3schools.com/css/tryit.asp?filename=trycss_sel_element_element)

# CSS COMBINATORS

## Child Selector

The child selector selects all elements that are the immediate children of a specified element.

The following example selects all <p> elements that are immediate children of a <div> element:

```
div > p {  
    background-color: yellow;  
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_sel\\_element\\_gt](https://www.w3schools.com/css/tryit.asp?filename=trycss_sel_element_gt)

# CSS COMBINATORS

## Adjacent Sibling Selector

The adjacent sibling selector selects all elements that are the adjacent siblings of a specified element.

Sibling elements must have the same parent element, and "adjacent" means "immediately following".

The following example selects all <p> elements that are placed immediately after <div> elements:

```
div + p {  
    background-color: yellow;  
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_sel\\_element\\_plus](https://www.w3schools.com/css/tryit.asp?filename=trycss_sel_element_plus)

# CSS COMBINATORS

## General Sibling Selector

The general sibling selector selects all elements that are siblings of a specified element.

The following example selects all <p> elements that are siblings of <div> elements:

```
div ~ p {  
    background-color: yellow;  
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_sel\\_element\\_tilde](https://www.w3schools.com/css/tryit.asp?filename=trycss_sel_element_tilde)

# CSS COMBINATORS

Test Yourself with Exercises!

- [Exercise 1](#)
- [Exercise 2](#)
- [Exercise 3](#)
- [Exercise 4](#)



# CSS PSEUDO-CLASS

Web Application Development

# CSS PSEUDO-CLASSES

What are Pseudo-classes?

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

# CSS PSEUDO-CLASSES

## Syntax

The syntax of pseudo-classes:

```
selector:pseudo-class {  
    property:value;  
}
```

# CSS PSEUDO-CLASSES

## Anchor Pseudo-classes

Links can be displayed in different ways:

```
/* unvisited link */
a:link {
    color: #FF0000;
}

/* visited link */
a:visited {
    color: #00FF00;
}

/* mouse over link */
a:hover {
    color: #FF00FF;
}

/* selected link */
a:active {
    color: #0000FF;
}
```

**Note:** `a:hover` **MUST** come after `a:link` and `a:visited` in the CSS definition in order to be effective! `a:active` **MUST** come after `a:hover` in the CSS definition in order to be effective! Pseudo-class names are not case-sensitive.

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_link](https://www.w3schools.com/css/tryit.asp?filename=trycss_link)

# CSS PSEUDO-CLASSES

## Pseudo-classes and CSS Classes

Pseudo-classes can be combined with CSS classes:

When you hover over the link in the example, it will change color:

```
a.highlight:hover {  
    color: #ff0000;  
}
```

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_pseudo-class](https://www.w3schools.com/css/tryit.asp?filename=trycss_pseudo-class)

# CSS PSEUDO-CLASSES

Hover on <div>

An example of using the `:hover` pseudo-class on a `<div>` element:

```
div:hover {  
    background-color: blue;  
}
```

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_pseudo-class\\_hover\\_div](https://www.w3schools.com/css/tryit.asp?filename=trycss_pseudo-class_hover_div)

# CSS PSEUDO-CLASSES

## Simple Tooltip Hover

Hover over a <div> element to show a <p> element (like a tooltip):

```
p {  
  display: none;  
  background-color: yellow;  
  padding: 20px;  
}  
  
div:hover p {  
  display: block;  
}
```

Hover over me to show the <p> element.

Tada! Here I am!

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_pseudo-class\\_hover\\_tooltip](https://www.w3schools.com/css/tryit.asp?filename=trycss_pseudo-class_hover_tooltip)

# CSS PSEUDO-CLASSES

## CSS - The :first-child Pseudo-class

The :first-child pseudo-class matches a specified element that is the first child of another element.

### Match the first <p> element

In the following example, the selector matches any <p> element that is the first child of any element:

```
p:first-child {  
    color: blue;  
}
```

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_first-child1](https://www.w3schools.com/css/tryit.asp?filename=trycss_first-child1)

# CSS PSEUDO-CLASSES

Match the first <i> element in all <p> elements

In the following example, the selector matches the first <i> element in all <p> elements:

```
p i:first-child {  
    color: blue;  
}
```

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_first-child2](https://www.w3schools.com/css/tryit.asp?filename=trycss_first-child2)

# CSS PSEUDO-CLASSES

Match all `<i>` elements in all first child `<p>` elements

In the following example, the selector matches all `<i>` elements in `<p>` elements that are the first child of another element:

```
p:first-child i {  
    color: blue;  
}
```

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_first-child3](https://www.w3schools.com/css/tryit.asp?filename=trycss_first-child3)

# CSS PSEUDO-CLASSES

## CSS - The :lang Pseudo-class

The `:lang` pseudo-class allows you to define special rules for different languages.

In the example below, `:lang` defines the quotation marks for `<q>` elements with `lang="no"`:

```
<html>
<head>
<style>
q:lang(no) {
    quotes: "~" "~";
}
</style>
</head>

<body>
<p>Some text <q lang="no">A quote in a paragraph</q> Some text.</p>
</body>
</html>
```

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_lang](https://www.w3schools.com/css/tryit.asp?filename=trycss_lang)

# CSS PSEUDO-CLASSES

## More Examples

- Add different styles to hyperlinks

This example demonstrates how to add other styles to hyperlinks.

- Use of :focus

This example demonstrates how to use the :focus pseudo-class.

# CSS PSEUDO-CLASSES

Test Yourself with Exercises!

- [Exercise 1](#)
- [Exercise 2](#)
- [Exercise 3](#)
- [Exercise 4](#)

# CSS PSEUDO-CLASSES

## All CSS Pseudo Classes

Selector	Example	Example description
<a href="#">:active</a>	a:active	Selects the active link
<a href="#">:checked</a>	input:checked	Selects every checked <input> element
<a href="#">:disabled</a>	input:disabled	Selects every disabled <input> element
<a href="#">:empty</a>	p:empty	Selects every <p> element that has no children
<a href="#">:enabled</a>	input:enabled	Selects every enabled <input> element
<a href="#">:first-child</a>	p:first-child	Selects every <p> elements that is the first child of its parent
<a href="#">:first-of-type</a>	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
<a href="#">:focus</a>	input:focus	Selects the <input> element that has focus
<a href="#">:hover</a>	a:hover	Selects links on mouse over
<a href="#">:in-range</a>	input:in-range	Selects <input> elements with a value within a specified range
<a href="#">:invalid</a>	input:invalid	Selects all <input> elements with an invalid value
<a href="#">:lang(<i>language</i>)</a>	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"
<a href="#">:last-child</a>	p:last-child	Selects every <p> elements that is the last child of its parent
<a href="#">:last-of-type</a>	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
<a href="#">:link</a>	a:link	Selects all unvisited links
<a href="#">:not(<i>selector</i>)</a>	:not(p)	Selects every element that is not a <p> element
<a href="#">:nth-child(<i>n</i>)</a>	p:nth-child(2)	Selects every <p> element that is the second child of its parent
<a href="#">:nth-last-child(<i>n</i>)</a>	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
<a href="#">:nth-last-of-type(<i>n</i>)</a>	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
<a href="#">:nth-of-type(<i>n</i>)</a>	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent

# CSS PSEUDO-CLASSES

## All CSS Pseudo Classes Continued

<a href="#">:only-of-type</a>	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
<a href="#">:only-child</a>	p:only-child	Selects every <p> element that is the only child of its parent
<a href="#">:optional</a>	input:optional	Selects <input> elements with no "required" attribute
<a href="#">:out-of-range</a>	input:out-of-range	Selects <input> elements with a value outside a specified range
<a href="#">:read-only</a>	input:read-only	Selects <input> elements with a "readonly" attribute specified
<a href="#">:read-write</a>	input:read-write	Selects <input> elements with no "readonly" attribute
<a href="#">:required</a>	input:required	Selects <input> elements with a "required" attribute specified
<a href="#">:root</a>	root	Selects the document's root element
<a href="#">:target</a>	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)
<a href="#">:valid</a>	input:valid	Selects all <input> elements with a valid value
<a href="#">:visited</a>	a:visited	Selects all visited links

# CSS PSEUDO-CLASSES

## All CSS Pseudo Elements

elector	Example	Example description
<a href="#"><u>::after</u></a>	p::after	Insert content after every <p> element
<a href="#"><u>::before</u></a>	p::before	Insert content before every <p> element
<a href="#"><u>::first-letter</u></a>	p::first-letter	Selects the first letter of every <p> element
<a href="#"><u>::first-line</u></a>	p::first-line	Selects the first line of every <p> element
<a href="#"><u>::selection</u></a>	p::selection	Selects the portion of an element that is selected by a user



# CSS PSEUDO-ELEMENT

Web Application Development

# CSS PSEUDO-ELEMENTS

What are Pseudo-Elements?

A CSS pseudo-element is used to style specified parts of an element.

For example, it can be used to:

- Style the first letter, or line, of an element
- Insert content before, or after, the content of an element

# CSS PSEUDO-ELEMENTS

## Syntax

The syntax of pseudo-elements:

```
selector::pseudo-element {  
    property:value;  
}
```

**Notice the double colon notation - `::first-line` versus `:first-line`**

The double colon replaced the single-colon notation for pseudo-elements in CSS3. This was an attempt from W3C to distinguish between **pseudo-classes** and **pseudo-elements**.

The single-colon syntax was used for both pseudo-classes and pseudo-elements in CSS2 and CSS1.

For backward compatibility, the single-colon syntax is acceptable for CSS2 and CSS1 pseudo-elements.

# CSS PSEUDO-ELEMENTS

## The ::first-line Pseudo-element

The `::first-line` pseudo-element is used to add a special style to the first line of a text.

The following example formats the first line of the text in all `<p>` elements:

```
p::first-line {
    color: #ff0000;
    font-variant: small-caps;
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_firstline](https://www.w3schools.com/css/tryit.asp?filename=trycss_firstline)

**Note:** The `::first-line` pseudo-element can only be applied to block-level elements.

The following properties apply to the `::first-line` pseudo-element:

- font properties
- color properties
- background properties
- word-spacing
- letter-spacing
- text-decoration
- vertical-align
- text-transform
- line-height
- clear

# CSS PSEUDO-ELEMENTS

## The ::first-letter Pseudo-element

The `::first-letter` pseudo-element is used to add a special style to the first letter of a text.

The following example formats the first letter of the text in all `<p>` elements:

```
p::first-letter {
    color: #ff0000;
    font-size: xx-large;
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_firstletter](https://www.w3schools.com/css/tryit.asp?filename=trycss_firstletter)

**Note:** The `::first-letter` pseudo-element can only be applied to block-level elements.

The following properties apply to the `::first-letter` pseudo-element:

- font properties
- color properties
- background properties
- margin properties
- padding properties
- border properties
- text-decoration
- vertical-align (only if "float" is "none")
- text-transform
- line-height
- float
- clear

# CSS PSEUDO-ELEMENTS

## Pseudo-elements and CSS Classes

Pseudo-elements can be combined with CSS classes:

```
p.intro::first-letter {  
    color: #ff0000;  
    font-size:200%;  
}
```

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_pseudo-element](https://www.w3schools.com/css/tryit.asp?filename=trycss_pseudo-element)

The example above will display the first letter of paragraphs with class="intro", in red and in a larger size.

# CSS PSEUDO-ELEMENTS

## Multiple Pseudo-elements

Several pseudo-elements can also be combined.

In the following example, the first letter of a paragraph will be red, in an xx-large font size. The rest of the first line will be blue, and in small-caps. The rest of the paragraph will be the default font size and color:

```
p::first-letter {  
    color: #ff0000;  
    font-size: xx-large;  
}  
  
p::first-line {  
    color: #0000ff;  
    font-variant: small-caps;  
}
```

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_firstline\\_letter](https://www.w3schools.com/css/tryit.asp?filename=trycss_firstline_letter)

# CSS PSEUDO-ELEMENTS

## CSS - The ::before Pseudo-element

The `::before` pseudo-element can be used to insert some content before the content of an element.

The following example inserts an image before the content of each `<h1>` element:

```
h1::before {  
    content: url(smiley.gif);  
}
```

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_before](https://www.w3schools.com/css/tryit.asp?filename=trycss_before)

# CSS PSEUDO-ELEMENTS

## CSS - The ::after Pseudo-element

The `::after` pseudo-element can be used to insert some content after the content of an element.

The following example inserts an image after the content of each `<h1>` element:

```
h1::after {  
    content: url(smiley.gif);  
}
```

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_after](https://www.w3schools.com/css/tryit.asp?filename=trycss_after)

# CSS PSEUDO-ELEMENTS

## CSS - The ::selection Pseudo-element

The `::selection` pseudo-element matches the portion of an element that is selected by a user.

The following CSS properties can be applied to `::selection`: `color`, `background`, `cursor`, and `outline`.

The following example makes the selected text red on a yellow background:

```
::selection {  
    color: red;  
    background: yellow;  
}
```

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_selection](https://www.w3schools.com/css/tryit.asp?filename=trycss3_selection)

# CSS PSEUDO-ELEMENTS

Test Yourself with Exercises!

- [Exercise 1](#)
- [Exercise 2](#)
- [Exercise 3](#)

# CSS PSEUDO-ELEMENTS

## All CSS Pseudo Elements

Selector	Example	Example description
<a href="#"><u>::after</u></a>	p::after	Insert something after the content of each <p> element
<a href="#"><u>::before</u></a>	p::before	Insert something before the content of each <p> element
<a href="#"><u>::first-letter</u></a>	p::first-letter	Selects the first letter of each <p> element
<a href="#"><u>::first-line</u></a>	p::first-line	Selects the first line of each <p> element
<a href="#"><u>::selection</u></a>	p::selection	Selects the portion of an element that is selected by a user

# CSS PSEUDO-ELEMENTS

## All CSS Pseudo Classes

Selector	Example	Example description
<a href="#">:active</a>	a:active	Selects the active link
<a href="#">:checked</a>	input:checked	Selects every checked <input> element
<a href="#">:disabled</a>	input:disabled	Selects every disabled <input> element
<a href="#">:empty</a>	p:empty	Selects every <p> element that has no children
<a href="#">:enabled</a>	input:enabled	Selects every enabled <input> element
<a href="#">:first-child</a>	p:first-child	Selects every <p> elements that is the first child of its parent
<a href="#">:first-of-type</a>	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
<a href="#">:focus</a>	input:focus	Selects the <input> element that has focus
<a href="#">:hover</a>	a:hover	Selects links on mouse over
<a href="#">:in-range</a>	input:in-range	Selects <input> elements with a value within a specified range
<a href="#">:invalid</a>	input:invalid	Selects all <input> elements with an invalid value

# CSS PSEUDO-ELEMENTS

## All CSS Pseudo Classes Continued

<a href="#">:lang(language)</a>	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"
<a href="#">:last-child</a>	p:last-child	Selects every <p> elements that is the last child of its parent
<a href="#">:last-of-type</a>	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
<a href="#">:link</a>	a:link	Selects all unvisited links
<a href="#">:not(selector)</a>	:not(p)	Selects every element that is not a <p> element
<a href="#">:nth-child(n)</a>	p:nth-child(2)	Selects every <p> element that is the second child of its parent
<a href="#">:nth-last-child(n)</a>	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
<a href="#">:nth-last-of-type(n)</a>	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
<a href="#">:nth-of-type(n)</a>	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
<a href="#">:only-of-type</a>	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
<a href="#">:only-child</a>	p:only-child	Selects every <p> element that is the only child of its parent
<a href="#">:optional</a>	input:optional	Selects <input> elements with no "required" attribute
<a href="#">:out-of-range</a>	input:out-of-range	Selects <input> elements with a value outside a specified range
<a href="#">:read-only</a>	input:read-only	Selects <input> elements with a "readonly" attribute specified
<a href="#">:read-write</a>	input:read-write	Selects <input> elements with no "readonly" attribute
<a href="#">:required</a>	input:required	Selects <input> elements with a "required" attribute specified
<a href="#">:root</a>	root	Selects the document's root element
<a href="#">:target</a>	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)
<a href="#">:valid</a>	input:valid	Selects all <input> elements with a valid value
<a href="#">:visited</a>	a:visited	Selects all visited links



# CSS OPACITY

Web Application Development

# CSS OPACITY

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

# CSS OPACITY

## Transparent Image



opacity 0.2



opacity 0.5



opacity 1  
(default)

The `opacity` property can take a value from 0.0 - 1.0. The lower value, the more transparent.

**Note:** IE8 and earlier use `filter: alpha(opacity=x)`. The `x` can take a value from 0 - 100. A lower value makes the element more transparent.

```
img {  
  opacity: 0.5;  
  filter: alpha(opacity=50); /* For IE8 and earlier */  
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_image\\_opacity](https://www.w3schools.com/css/tryit.asp?filename=trycss_image_opacity)

# CSS OPACITY

## Transparent Hover Effect

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

```
img {  
    opacity: 0.5;  
    filter: alpha(opacity=50); /* For IE8 and earlier */  
}  
  
img:hover {  
    opacity: 1.0;  
    filter: alpha(opacity=100); /* For IE8 and earlier */  
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_image\\_transparency](https://www.w3schools.com/css/tryit.asp?filename=trycss_image_transparency)

# CSS OPACITY

## Transparent Hover Effect Continued

The first CSS block is similar to the code in Example 1. In addition, we have added what should happen when a user hovers over one of the images. In this case we want the image to NOT be transparent when the user hovers over it. The CSS for this is `opacity:1;`

When the mouse pointer moves away from the image, the image will be transparent again.

An example of reversed hover effect:

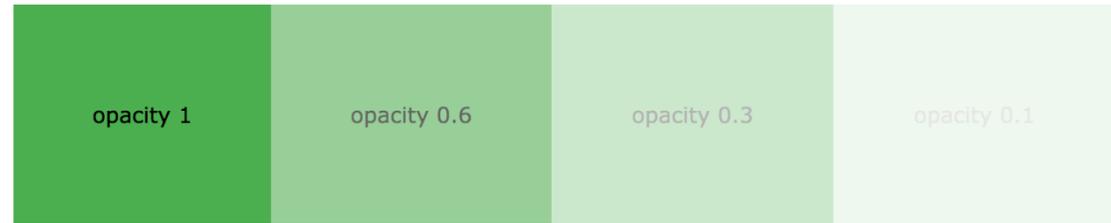
```
img:hover {  
    opacity: 0.5;  
    filter: alpha(opacity=50); /* For IE8 and earlier */  
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_image\\_transparency2](https://www.w3schools.com/css/tryit.asp?filename=trycss_image_transparency2)

# CSS OPACITY

## Transparent Box



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

```
div {  
  opacity: 0.3;  
  filter: alpha(opacity=30); /* For IE8 and earlier */  
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_opacity\\_box](https://www.w3schools.com/css/tryit.asp?filename=trycss_opacity_box)

# CSS OPACITY

Transparency using RGBA



If you do not want to apply opacity to child elements, like in our example above, use **RGBA** color values. The following example sets the opacity for the background color and not the text:

You learned from our [CSS Colors Chapter](#), that you can use RGB as a color value. In addition to RGB, you can use an RGB color value with an alpha channel (RGBA) - which specifies the opacity for a color.

An RGBA color value is specified with: `rgba(red, green, blue, alpha)`. The *alpha* parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).

**Tip:** You will learn more about RGBA Colors in our [CSS Colors Chapter](#).

```
div {  
    background: rgba(76, 175, 80, 0.3) /* Green background with 30%  
opacity */  
}
```

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_opacity\\_box2](https://www.w3schools.com/css/tryit.asp?filename=trycss_opacity_box2)

# CSS OPACITY

## Text in Transparent Box

First, we create a `<div>` element (`class="background"`) with a background image, and a border. Then we create another `<div>` (`class="transbox"`) inside the first `<div>`. The `<div class="transbox">` have a background color, and a border - the div is transparent. Inside the transparent `<div>`, we add some text inside a `<p>` element.



```
<html>
<head>
<style>
div.background {
    background: url(klematis.jpg) repeat;
    border: 2px solid black;
}

div.transbox {
    margin: 30px;
    background-color: #ffffff;
    border: 1px solid black;
    opacity: 0.6;
    filter: alpha(opacity=60); /* For IE8 and earlier
*/
}

div.transbox p {
    margin: 5%;
    font-weight: bold;
    color: #000000;
}
</style>
</head>
<body>

<div class="background">
  <div class="transbox">
    <p>This is some text that is placed in the
transparent box.</p>
  </div>
</div>

</body>
</html>
```

**Try it Yourself:**

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_transparency](https://www.w3schools.com/css/tryit.asp?filename=trycss_transparency)

# CSS OPACITY

Test Yourself with Exercises!

- [Exercise 1](#)
- [Exercise 2](#)



# CSS NAVIGATION BAR

Web Application Development

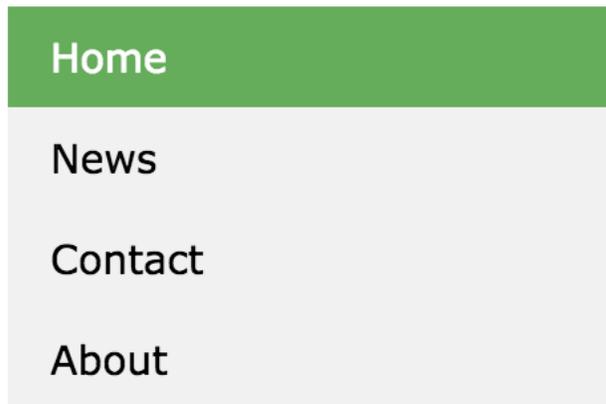
# CSS NAVIGATION BAR

## Navigation Bars

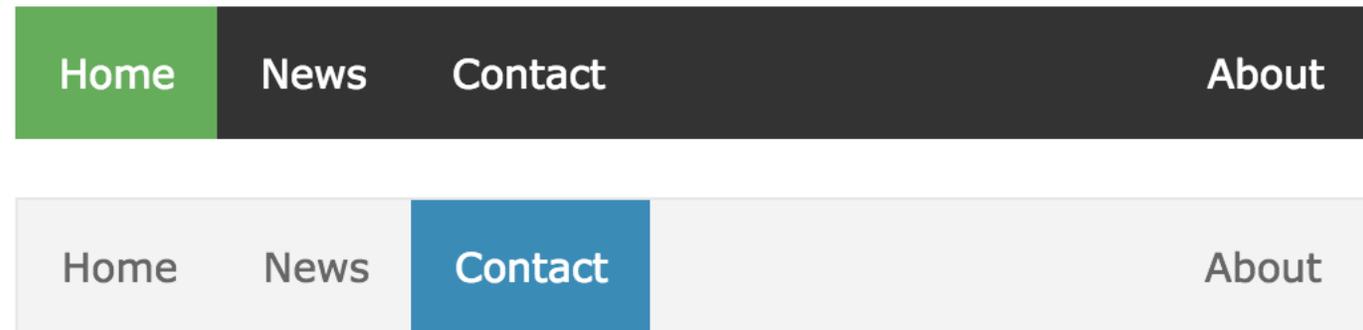
Having easy-to-use navigation is important for any web site.

With CSS you can transform boring HTML menus into good-looking navigation bars.

Vertical



Horizontal



# CSS NAVIGATION BAR

Navigation Bar = List of Links

A navigation bar needs standard HTML as a base.

In our examples we will build the navigation bar from a standard HTML list.

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

```
<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>
```

**Try it Yourself:**

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_basic\\_html](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_basic_html)

# CSS NAVIGATION BAR

Navigation Bar = List of Links Continued

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

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

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_basic\\_css](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_basic_css)

## Example explained:

- `list-style-type: none;`  
Removes the bullets. A navigation bar does not need list markers
- **Set `margin: 0;` and `padding: 0;` to remove browser default settings**

The code in the example to the left is the standard code used in both vertical, and horizontal navigation bars.

# CSS NAVIGATION BAR

## Vertical Navigation Bar

To build a vertical navigation bar, you can style the `<a>` elements inside the list, in addition to the code above:

```
li a {  
    display: block;  
    width: 60px;  
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_vertical](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_vertical)

## Example Explained:

- `display: block;` - Displaying the links as block elements makes the whole link area clickable (not just the text), and it allows us to specify the width (and padding, margin, height, etc. if you want)
- `width: 60px;` - Block elements take up the full width available by default. We want to specify a 60 pixels width

# CSS NAVIGATION BAR

## Vertical Navigation Bar Continued

You can also set the width of `<ul>`, and remove the width of `<a>`, as they will take up the full width available when displayed as block elements. This will produce the same result as our previous example:

```
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
    width: 60px;  
}  
  
li a {  
    display: block;  
}
```

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_vertical2](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_vertical2)

# CSS NAVIGATION BAR

## Vertical Navigation Bar Examples

Create a basic vertical navigation bar with a gray background color and change the background color of the links when the user moves the mouse over them:

```
ul {  
  list-style-type: none;  
  margin: 0;  
  padding: 0;  
  width: 200px;  
  background-color: #f1f1f1;  
}  
  
li a {  
  display: block;  
  color: #000;  
  padding: 8px 16px;  
  text-decoration: none;  
}  
  
/* Change the link color on hover */  
li a:hover {  
  background-color: #555;  
  color: white;  
}
```



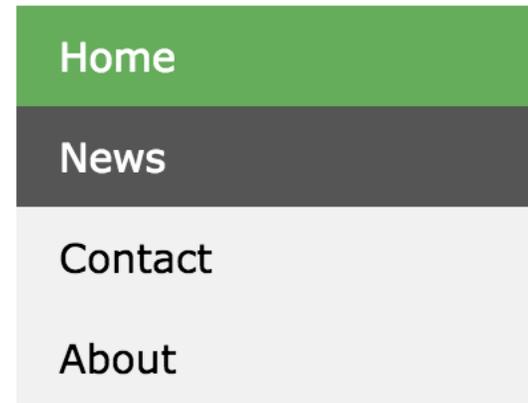
Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_vertical\\_gray](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_vertical_gray)

# CSS NAVIGATION BAR

Active/Current Navigation Link

Add an "active" class to the current link to let the user know which page he/she is on:

```
.active {  
    background-color: #4CAF50;  
    color: white;  
}
```



Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_vertical\\_active](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_vertical_active)

# CSS NAVIGATION BAR

## Center Links & Add Borders

Add `text-align:center` to `<li>` or `<a>` to center the links.

Add the `border` property to `<ul>` add a border around the navbar. If you also want borders inside the navbar, add a `border-bottom` to all `<li>` elements, except for the last one:

```
ul {  
  border: 1px solid #555;  
}  
  
li {  
  text-align: center;  
  border-bottom: 1px solid #555;  
}  
  
li:last-child {  
  border-bottom: none;  
}
```

Home
News
Contact
About

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_vertical\\_borders](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_vertical_borders)

# CSS NAVIGATION BAR

## Full-height Fixed Vertical Navbar

Create a full-height, "sticky" side navigation:

```
ul {  
  list-style-type: none;  
  margin: 0;  
  padding: 0;  
  width: 25%;  
  background-color: #f1f1f1;  
  height: 100%; /* Full height */  
  position: fixed; /* Make it stick, even on scroll */  
  overflow: auto; /* Enable scrolling if the sidenav has too much content */  
}
```



Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_vertical\\_fixed](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_vertical_fixed)

**Note:** This example might not work properly on mobile devices.

# CSS NAVIGATION BAR

## Horizontal Navigation Bar

There are two ways to create a horizontal navigation bar. Using **inline** or **floating** list items.

### Inline List Items

One way to build a horizontal navigation bar is to specify the `<li>` elements as inline, in addition to the "standard" code above:

```
li {  
    display: inline;  
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_horizontal](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_horizontal)

### Example explained:

- `display: inline;` - By default, `<li>` elements are block elements. Here, we remove the line breaks before and after each list item, to display them on one line

# CSS NAVIGATION BAR

## Horizontal Navigation Bar Continued

### Floating List Items

Another way of creating a horizontal navigation bar is to float the <li> elements, and specify a layout for the navigation links:

```
li {  
    float: left;  
}  
  
a {  
    display: block;  
    padding: 8px;  
    background-color: #dddddd;  
}
```

**Try it Yourself:**

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_horizontal\\_float](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_horizontal_float)

# CSS NAVIGATION BAR

## Horizontal Navigation Bar Continued

### Floating List Items Example Explained:

- `float: left;` - use float to get block elements to slide next to each other
- `display: block;` - Displaying the links as block elements makes the whole link area clickable (not just the text), and it allows us to specify padding (and height, width, margins, etc. if you want)
- `padding: 8px;` - Since block elements take up the full width available, they cannot float next to each other. Therefore, specify some padding to make them look good
- `background-color: #dddddd;` - Add a gray background-color to each a element

**Tip:** Add the background-color to `<ul>` instead of each `<a>` element if you want a full-width background color:

```
ul {  
    background-color: #dddddd;  
}
```

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_horizontal\\_float2](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_horizontal_float2)

# CSS NAVIGATION BAR

## Horizontal Navigation Bar Examples

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:

```
ul {  
  list-style-type: none;  
  margin: 0;  
  padding: 0;  
  overflow: hidden;  
  background-color: #333;  
}  
  
li {  
  float: left;  
}  
  
li a {  
  display: block;  
  color: white;  
  text-align: center;  
  padding: 14px 16px;  
  text-decoration: none;  
}  
  
/* Change the link color to #111 (black) on hover */  
li a:hover {  
  background-color: #111;  
}
```



Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_horizontal\\_black](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_horizontal_black)

# CSS NAVIGATION BAR

## Horizontal Navigation Bar Examples - Active/Current Navigation Link

Add an "active" class to the current link to let the user know which page he/she is on:



```
.active {  
    background-color: #4CAF50;  
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_horizontal\\_black\\_active](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_horizontal_black_active)

# CSS NAVIGATION BAR

## Horizontal Navigation Bar Examples - Right-Align Links

Right-align links by floating the list items to the right (`float:right;`):



```
<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li style="float:right"><a class="active" href="#about">About</a><
/li>
</ul>
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_horizontal\\_black\\_right](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_horizontal_black_right)

# CSS NAVIGATION BAR

## Horizontal Navigation Bar Examples - Border Dividers

Add the `border-right` property to `<li>` to create link dividers:



```
/* Add a gray right border to all list items, except the last item
(last-child) */
li {
    border-right: 1px solid #bbb;
}

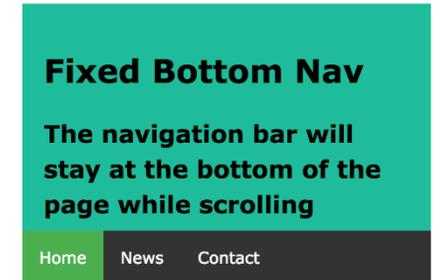
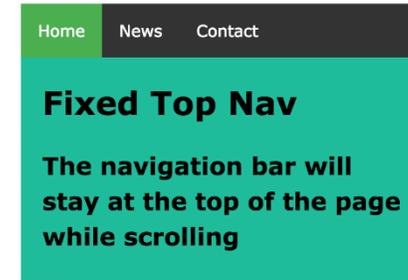
li:last-child {
    border-right: none;
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_horizontal\\_dividers](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_horizontal_dividers)

# CSS NAVIGATION BAR

## Horizontal Navigation Bar Examples - Fixed Navigation Bar



Make the navigation bar stay at the top or the bottom of the page, even when the user scrolls the page:

### Fixed Top

```
ul {  
  position: fixed;  
  top: 0;  
  width: 100%;  
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_horizontal\\_black\\_fixed](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_horizontal_black_fixed)

### Fixed Bottom

```
ul {  
  position: fixed;  
  bottom: 0;  
  width: 100%;  
}
```

Try it Yourself:

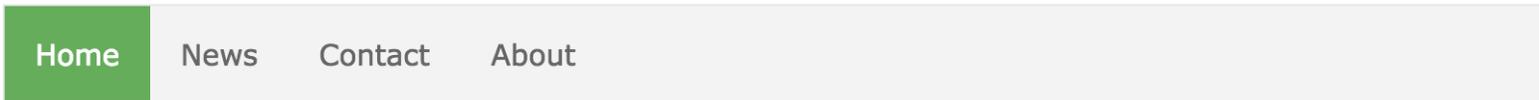
[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_horizontal\\_black\\_fixed2](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_horizontal_black_fixed2)

**Note:** Fixed position might not work properly on mobile devices.

# CSS NAVIGATION BAR

## Horizontal Navigation Bar Examples - Gray Horizontal Navbar

An example of a gray horizontal navigation bar with a thin gray border:



```
ul {  
    border: 1px solid #e7e7e7;  
    background-color: #f3f3f3;  
}  
  
li a {  
    color: #666;  
}
```

**Try it Yourself:**

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_horizontal\\_gray](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_horizontal_gray)

# CSS NAVIGATION BAR

## Horizontal Navigation Bar Examples - Sticky Navbar

Use `position: sticky;` to `<li>` to create a sticky navbar.

A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like `position:fixed`).

```
ul {  
    position: -webkit-sticky; /* Safari */  
    position: sticky;  
    top: 0;  
}
```

Try it Yourself:

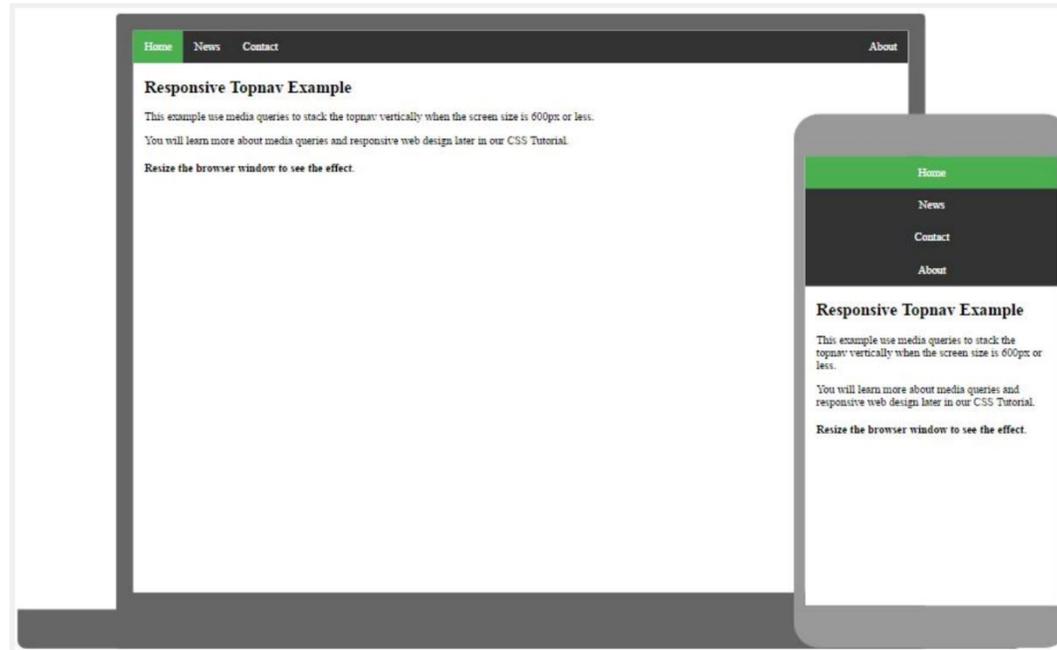
[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_sticky](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_sticky)

**Note:** Internet Explorer, Edge 15 and earlier versions do not support sticky positioning. Safari requires a `-webkit-` prefix (see example above). You must also specify at least one of `top`, `right`, `bottom` or `left` for sticky positioning to work.

# CSS NAVIGATION BAR

## More Examples - Responsive Topnav

How to use CSS media queries to create a responsive top navigation.



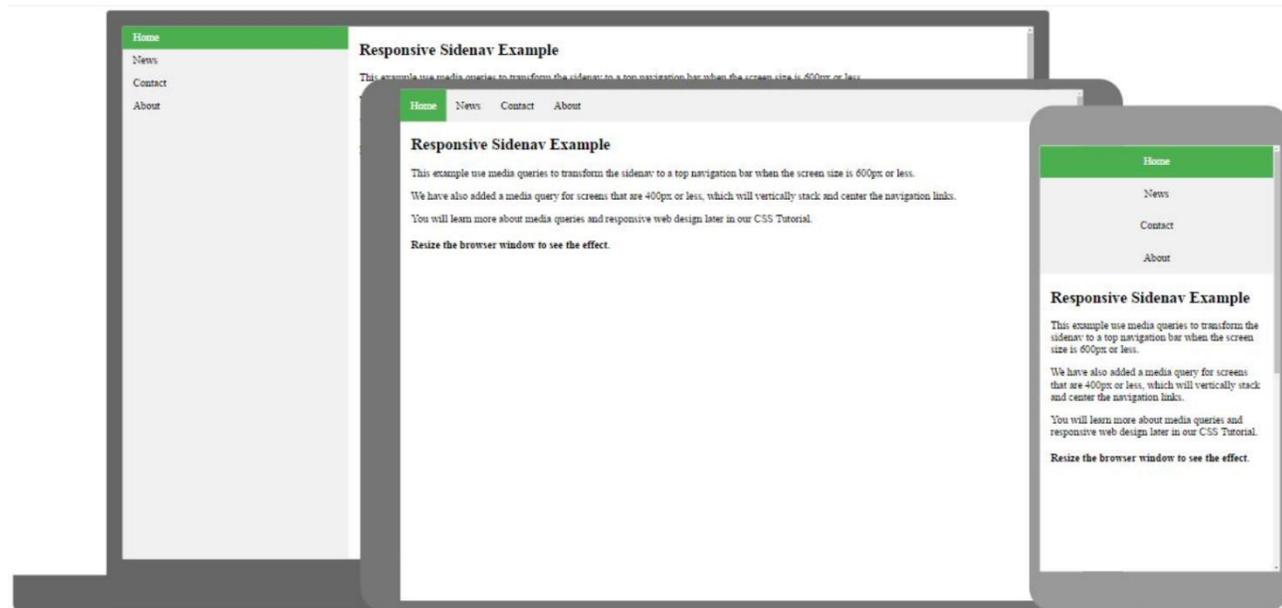
Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_horizontal\\_responsive](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_horizontal_responsive)

# CSS NAVIGATION BAR

More Examples - Responsive Sidenav

How to use CSS media queries to create a responsive side navigation.



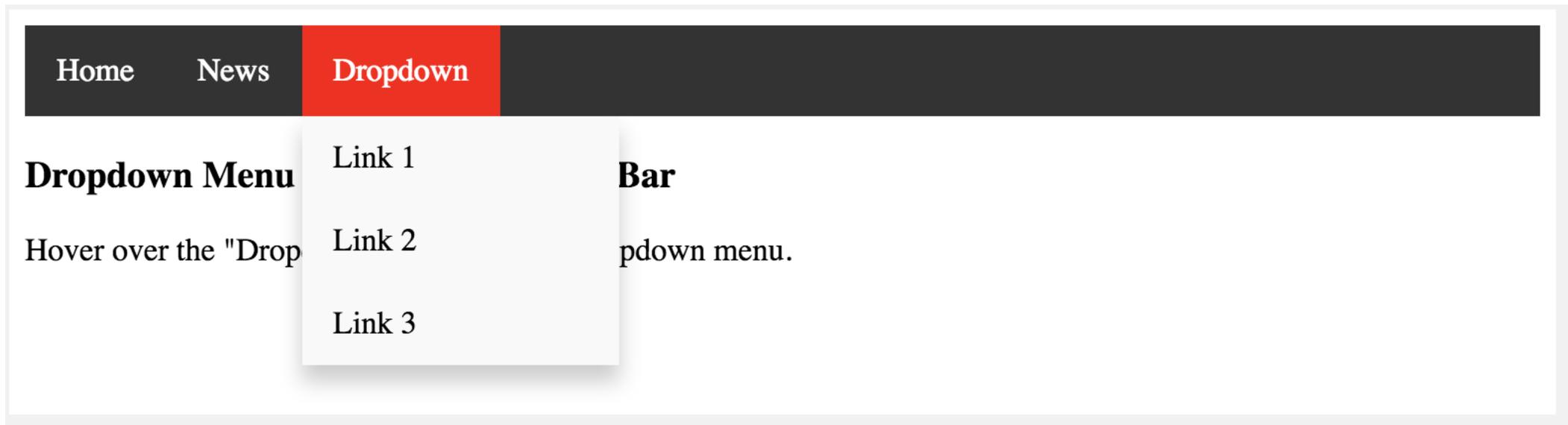
Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_navbar\\_vertical\\_responsive](https://www.w3schools.com/css/tryit.asp?filename=trycss_navbar_vertical_responsive)

# CSS NAVIGATION BAR

More Examples - Dropdown Navbar

How to add a dropdown menu inside a navigation bar.



Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_dropdown\\_navbar](https://www.w3schools.com/css/tryit.asp?filename=trycss_dropdown_navbar)



# CSS DROPDOWNS

Web Application Development

# CSS DROPDOWNS

## Basic Dropdown

Create a dropdown box that appears when the user moves the mouse over an element.

```
<style>
.dropdown {
  position: relative;
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  padding: 12px 16px;
  z-index: 1;
}

.dropdown:hover .dropdown-content {
  display: block;
}
</style>

<div class="dropdown">
  <span>Mouse over me</span>
  <div class="dropdown-content">
    <p>Hello World!</p>
  </div>
</div>
```

Try it Yourself: [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_dropdown\\_text](https://www.w3schools.com/css/tryit.asp?filename=trycss_dropdown_text)

# CSS DROPDOWNS

## Basic Dropdown - Example Explained

**HTML)** Use any element to open the dropdown content, e.g. a `<span>`, or a `<button>` element.

Use a container element (like `<div>`) to create the dropdown content and add whatever you want inside of it.

Wrap a `<div>` element around the elements to position the dropdown content correctly with CSS.

**CSS)** The `.dropdown` class uses `position: relative`, which is needed when we want the dropdown content to be placed right below the dropdown button (using `position: absolute`).

The `.dropdown-content` class holds the actual dropdown content. It is hidden by default, and will be displayed on hover (see below). Note the `min-width` is set to 160px. Feel free to change this. **Tip:** If you want the width of the dropdown content to be as wide as the dropdown button, set the `width` to 100% (and `overflow: auto` to enable scroll on small screens).

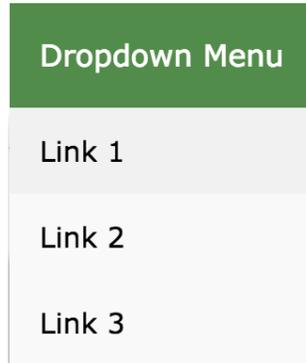
Instead of using a border, we have used the CSS `box-shadow` property to make the dropdown menu look like a "card".

The `:hover` selector is used to show the dropdown menu when the user moves the mouse over the dropdown button.

# CSS DROPDOWNS

## Dropdown Menu

Create a dropdown menu that allows the user to choose an option from a list:



This example is similar to the previous one, except that we add links inside the dropdown box and style them to fit a styled dropdown button.

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_dropdown\\_button](https://www.w3schools.com/css/tryit.asp?filename=trycss_dropdown_button)

```
<style>
/* Style The Dropdown Button */
.dropbtn {
  background-color: #4CAF50;
  color: white;
  padding: 16px;
  font-size: 16px;
  border: none;
  cursor: pointer;
}

/* The container <div> - needed to position the dropdown content */
.dropdown {
  position: relative;
  display: inline-block;
}

/* Dropdown Content (Hidden by Default) */
.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

/* Links inside the dropdown */
.dropdown-content a {
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
}

/* Change color of dropdown links on hover */
.dropdown-content a:hover {background-color: #f1f1f1}

/* Show the dropdown menu on hover */
.dropdown:hover .dropdown-content {
  display: block;
}

/* Change the background color of the dropdown button when the dropdown
content is shown */
.dropdown:hover .dropbtn {
  background-color: #3e8e41;
}
</style>

<div class="dropdown">
  <button class="dropbtn">Dropdown</button>
  <div class="dropdown-content">
    <a href="#">Link 1</a>
    <a href="#">Link 2</a>
    <a href="#">Link 3</a>
  </div>
</div>
```

# CSS DROPDOWNS

## Right-aligned Dropdown Content

If you want the dropdown menu to go from right to left, instead of left to right, add `right: 0;`

```
.dropdown-content {  
    right: 0;  
}
```

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_dropdown\\_right](https://www.w3schools.com/css/tryit.asp?filename=trycss_dropdown_right)



# CSS DROPDOWNS

More Examples

- **Dropdown Image** – How to add an image and other content inside the dropdown box.

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_dropdown\\_image](https://www.w3schools.com/css/tryit.asp?filename=trycss_dropdown_image)

- **Dropdown Navbar** – How to add a dropdown menu inside a navigation bar.

Try it Yourself:

[https://www.w3schools.com/css/tryit.asp?filename=trycss\\_dropdown\\_navbar](https://www.w3schools.com/css/tryit.asp?filename=trycss_dropdown_navbar)



**THE END**

Web Application Development