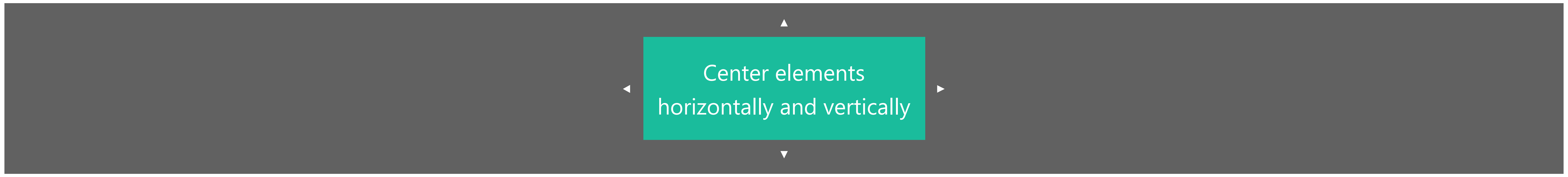
CSS HOME
CSS Introduction
CSS Syntax
CSS Selectors
CSS How To
CSS Comments
CSS Colors
CSS Backgrounds
CSS Borders
CSS Margins
CSS Padding
CSS Height/Width
CSS Box Model
CSS Outline
CSS Text
CSS Fonts
CSS Icons
CSS Links
CSS Lists
CSS Tables
CSS Display
CSS Max-width
CSS Position
CSS Z-index
CSS Overflow
CSS Float
CSS Inline-block
CSS Align
CSS Combinators
CSS Pseudo-class
CSS Pseudo-element
CSS Opacity
CSS Navigation Bar
CSS Dropdowns
CSS Image Gallery
CSS Image Sprites
CSS Attr Selectors
CSS Forms
CSS Counters
CSS Website Layout
CSS Units
CSS Specificity
CSS !important
CSS Math Functions

CSS Advanced
CSS Rounded Corners
CSS Border Images
CSS Backgrounds
CSS Colors
CSS Color Keywords
CSS Gradients
CSS Shadows
CSS Text Effects
CSS Web Fonts
CSS 2D Transforms
CSS 3D Transforms
CSS Transitions
CSS Animations
CSS Tooltips
CSS Style Images
CSS Image Reflection
CSS object-fit
CSS object-position
CSS Masking
CSS Buttons
CSS Pagination
CSS Multiple Columns
CSS User Interface
CSS Variables
CSS Box Sizing
CSS Media Queries
CSS MQ Examples
CSS Flexbox

CSS Responsive
RWD Intro
RWD Viewport
RWD Grid View
RWD Media Queries
RWD Images

CSS Layout - Horizontal & Vertical Align

❮ Previous Next ❯


Center elements horizontally and vertically

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:

This div element is centered.

Example

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

Try it Yourself »

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

Center Align Text

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

This text is centered.

Example

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

Try it Yourself »

Tip: For more examples on how to align text, see the CSS Text chapter.

Center an Image

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

BLACK FRIDAY DEALS
UP TO 67% OFF
On all Courses & Certificates!
SHOP NOW

COLOR PICKER

Learn Front-end Development
High Demand:
Front-end skills are in demand!
Secure your future.
Check It Out!

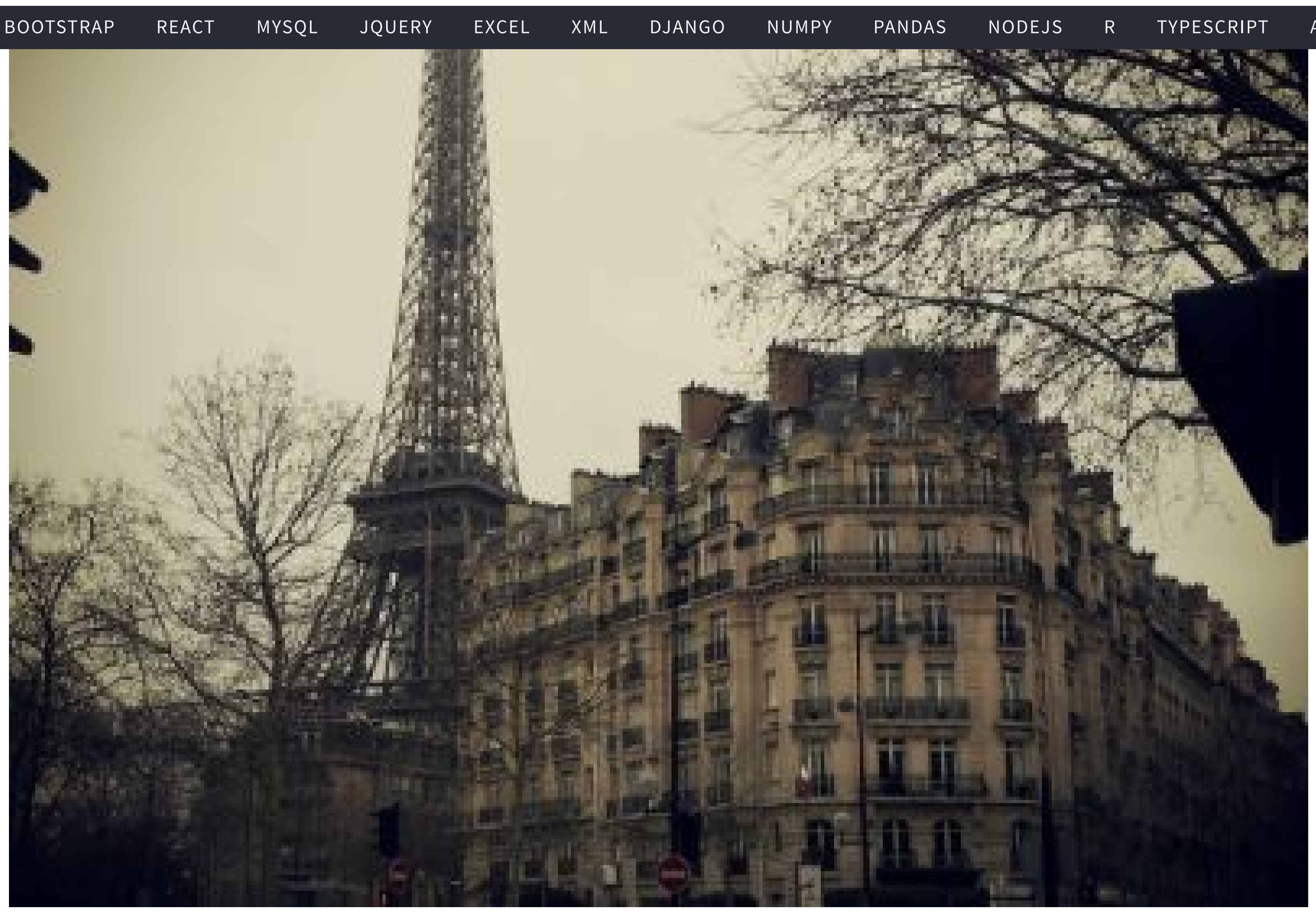
CSS HOME
CSS Introduction
CSS Syntax
CSS Selectors
CSS How To
CSS Comments
CSS Colors
CSS Backgrounds
CSS Borders
CSS Margins
CSS Padding
CSS Height/Width
CSS Box Model
CSS Outline
CSS Text
CSS Fonts
CSS Icons
CSS Links
CSS Lists
CSS Tables
CSS Display
CSS Max-width
CSS Position
CSS Z-index
CSS Overflow
CSS Float
CSS Inline-block
CSS Align
CSS Combinators
CSS Pseudo-class
CSS Pseudo-element
CSS Opacity
CSS Navigation Bar
CSS Dropdowns
CSS Image Gallery
CSS Image Sprites
CSS Attr Selectors
CSS Forms
CSS Counters
CSS Website Layout
CSS Units
CSS Specificity
CSS !important
CSS Math Functions

CSS Advanced

CSS Rounded Corners
CSS Border Images
CSS Backgrounds
CSS Colors
CSS Color Keywords
CSS Gradients
CSS Shadows
CSS Text Effects
CSS Web Fonts
CSS 2D Transforms
CSS 3D Transforms
CSS Transitions
CSS Animations
CSS Tooltips
CSS Style Images
CSS Image Reflection
CSS object-fit
CSS object-position
CSS Masking
CSS Buttons
CSS Pagination
CSS Multiple Columns
CSS User Interface
CSS Variables
CSS Box Sizing
CSS Media Queries
CSS MQ Examples
CSS Flexbox

CSS Responsive

RWD Intro
RWD Viewport
RWD Grid View
RWD Media Queries
RWD Images



Example

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

Try it Yourself »

Left and Right Align - Using position

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

In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since.

Example

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

Try it Yourself »

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

Left and Right Align - Using float

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

Example

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

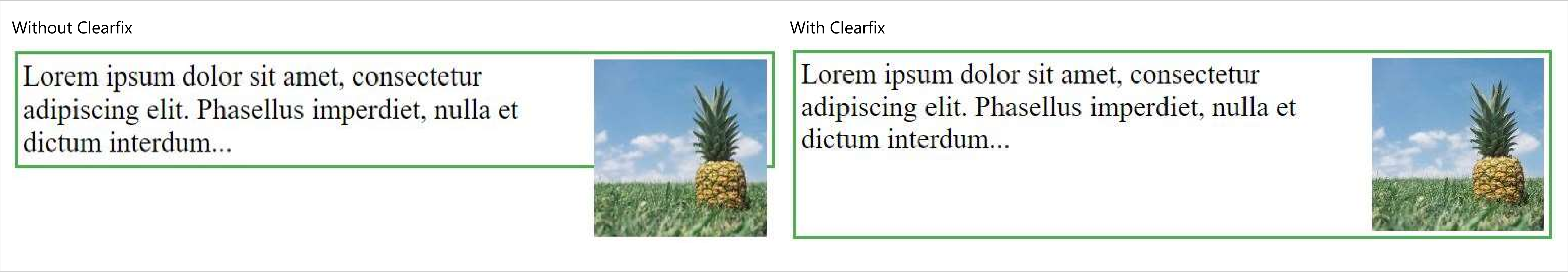
CSS HOME
CSS Introduction
CSS Syntax
CSS Selectors
CSS How To
CSS Comments
CSS Colors
CSS Backgrounds
CSS Borders
CSS Margins
CSS Padding
CSS Height/Width
CSS Box Model
CSS Outline
CSS Text
CSS Fonts
CSS Icons
CSS Links
CSS Lists
CSS Tables
CSS Display
CSS Max-width
CSS Position
CSS Z-index
CSS Overflow
CSS Float
CSS Inline-block
CSS Align
CSS Combinators
CSS Pseudo-class
CSS Pseudo-element
CSS Opacity
CSS Navigation Bar
CSS Dropdowns
CSS Image Gallery
CSS Image Sprites
CSS Attr Selectors
CSS Forms
CSS Counters
CSS Website Layout
CSS Units
CSS Specificity
CSS !important
CSS Math Functions

CSS Advanced
CSS Rounded Corners
CSS Border Images
CSS Backgrounds
CSS Colors
CSS Color Keywords
CSS Gradients
CSS Shadows
CSS Text Effects
CSS Web Fonts
CSS 2D Transforms
CSS 3D Transforms
CSS Transitions
CSS Animations
CSS Tooltips
CSS Style Images
CSS Image Reflection
CSS object-fit
CSS object-position
CSS Masking
CSS Buttons
CSS Pagination
CSS Multiple Columns
CSS User Interface
CSS Variables
CSS Box Sizing
CSS Media Queries
CSS MQ Examples
CSS Flexbox

CSS Responsive
RWD Intro
RWD Viewport
RWD Grid View
RWD Media Queries
RWD Images

The clearfix Hack

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 below).


Without Clearfix
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum...
With Clearfix
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum...

Then we can add the clearfix hack to the containing element to fix this problem:

Example

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

Try it Yourself »

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`:

I am vertically centered.

Example

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

Try it Yourself »

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

I am vertically and horizontally centered.

Example

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

Tutorials ▾ Exercises ▾ Certificates ▾ Services ▾ Jobs Spaces Get Certified Sign Up Log in
HTML CSS JAVASCRIPT SQL PYTHON JAVA PHP HOW TO W3.CSS C C++ C# BOOTSTRAP REACT MYSQL JQUERY EXCEL XML DJANGO NUMPY PANDAS NODEJS R TYPESCRIPT ANGULAR GIT POSTGRESQL MONGODB ASP AWS AI GO KOTLIN SASS VUE GEN AI SCIPY CYBERSECURITY DATA SCIENCE

CSS HOME
CSS Introduction
CSS Syntax
CSS Selectors
CSS How To
CSS Comments
CSS Colors
CSS Backgrounds
CSS Borders
CSS Margins
CSS Padding
CSS Height/Width
CSS Box Model
CSS Outline
CSS Text
CSS Fonts
CSS Icons
CSS Links
CSS Lists
CSS Tables
CSS Display
CSS Max-width
CSS Position
CSS Z-index
CSS Overflow
CSS Float
CSS Inline-block
CSS Align
CSS Combinators
CSS Pseudo-class
CSS Pseudo-element
CSS Opacity
CSS Navigation Bar
CSS Dropdowns
CSS Image Gallery
CSS Image Sprites
CSS Attr Selectors
CSS Forms
CSS Counters
CSS Website Layout
CSS Units
CSS Specificity
CSS !important
CSS Math Functions

CSS Advanced
CSS Rounded Corners
CSS Border Images
CSS Backgrounds
CSS Colors
CSS Color Keywords
CSS Gradients
CSS Shadows
CSS Text Effects
CSS Web Fonts
CSS 2D Transforms
CSS 3D Transforms
CSS Transitions
CSS Animations
CSS Tooltips
CSS Style Images
CSS Image Reflection
CSS object-fit
CSS object-position
CSS Masking
CSS Buttons
CSS Pagination
CSS Multiple Columns
CSS User Interface
CSS Variables
CSS Box Sizing
CSS Media Queries
CSS MQ Examples
CSS Flexbox

CSS Responsive
RWD Intro
RWD Viewport
RWD Grid View
RWD Media Queries
RWD Images

Center Vertically - Using line-height

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

I am vertically and horizontally centered.

Example

```
.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 »

Center Vertically - Using position & transform

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

I am vertically and horizontally centered.

Example

```
.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 »

Tip: You will learn more about the transform property in our 2D Transforms Chapter.

Center Vertically - Using Flexbox

You can also use flexbox to center things. Just note that flexbox is not supported in IE10 and earlier versions:

I am vertically and horizontally centered.

CSS HOME
CSS Introduction
CSS Syntax
CSS Selectors
CSS How To
CSS Comments
CSS Colors
CSS Backgrounds
CSS Borders
CSS Margins
CSS Padding
CSS Height/Width
CSS Box Model
CSS Outline
CSS Text
CSS Fonts
CSS Icons
CSS Links
CSS Lists
CSS Tables
CSS Display
CSS Max-width
CSS Position
CSS Z-index
CSS Overflow
CSS Float
CSS Inline-block
CSS Align
CSS Combinators
CSS Pseudo-class
CSS Pseudo-element
CSS Opacity
CSS Navigation Bar
CSS Dropdowns
CSS Image Gallery
CSS Image Sprites
CSS Attr Selectors
CSS Forms
CSS Counters
CSS Website Layout
CSS Units
CSS Specificity
CSS !important
CSS Math Functions

CSS Advanced

CSS Rounded Corners
CSS Border Images
CSS Backgrounds
CSS Colors
CSS Color Keywords
CSS Gradients
CSS Shadows
CSS Text Effects
CSS Web Fonts
CSS 2D Transforms
CSS 3D Transforms
CSS Transitions
CSS Animations
CSS Tooltips
CSS Style Images
CSS Image Reflection
CSS object-fit
CSS object-position
CSS Masking
CSS Buttons
CSS Pagination
CSS Multiple Columns
CSS User Interface
CSS Variables
CSS Box Sizing
CSS Media Queries
CSS MQ Examples
CSS Flexbox

CSS Responsive

RWD Intro
RWD Viewport
RWD Grid View
RWD Media Queries
RWD Images

```
.center {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 200px;
  border: 3px solid green;
}
```

Try it Yourself »

Tip: You will learn more about Flexbox in our CSS Flexbox Chapter.

Test Yourself With Exercises

Exercise:

Use the margin property to make sure that the <div> element is center aligned according to its parent element.

```
<style>
.intro {
  width: 200px;
          :      ;
}
</style>

<body>

<div class="intro">
Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Phasellus imperdiet, nulla et dictum interdum,
nisi lorem egestas odio,
vitae scelerisque enim ligula venenatis dolor.
</div>

</body>
```

Submit Answer »

Start the Exercise

❮ Previous Log in to track progress Next ❯

schools SPACES UPGRADE NEWSLETTER GET CERTIFIED REPORT ERROR

Top Tutorials
HTML Tutorial
CSS Tutorial
JavaScript Tutorial
How To Tutorial
SQL Tutorial
Python Tutorial
W3.CSS Tutorial
Bootstrap Tutorial
PHP Tutorial
Java Tutorial
C++ Tutorial
jQuery Tutorial

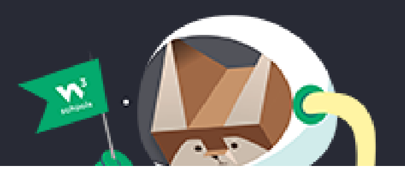
Top References
HTML Reference
CSS Reference
JavaScript Reference
SQL Reference
Python Reference
W3.CSS Reference
Bootstrap Reference
PHP Reference
HTML Colors
Java Reference
Angular Reference
jQuery Reference

Top Examples
HTML Examples
CSS Examples
JavaScript Examples
How To Examples
SQL Examples
Python Examples
W3.CSS Examples
Bootstrap Examples
PHP Examples
Java Examples
XML Examples
jQuery Examples

Get Certified
HTML Certificate
CSS Certificate
JavaScript Certificate
Front End Certificate
SQL Certificate
Python Certificate
PHP Certificate
jQuery Certificate
Java Certificate
C++ Certificate
C# Certificate
XML Certificate

FORUM ABOUT

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using W3Schools, you agree to have read and accepted our terms of use, cookie and privacy policy.



Tutorials ▾ Exercises ▾ Certificates ▾ Services ▾ Jobs Spaces Get Certified Sign Up Log in

HTML CSS JAVASCRIPT SQL PYTHON JAVA PHP HOW TO W3.CSS C C++ C# BOOTSTRAP REACT MYSQL JQUERY EXCEL XML DJANGO NUMPY PANDAS NODEJS R TYPESCRIPT ANGULAR GIT POSTGRESQL MONGODB ASP AWS AI GO KOTLIN SASS VUE GEN AI SCIPY CYBERSECURITY DATA SCIENCE

CSS HOME
CSS Introduction
CSS Syntax
CSS Selectors
CSS How To
CSS Comments
CSS Colors
CSS Backgrounds
CSS Borders
CSS Margins
CSS Padding
CSS Height/Width
CSS Box Model
CSS Outline
CSS Text
CSS Fonts
CSS Icons
CSS Links
CSS Lists
CSS Tables
CSS Display
CSS Max-width
CSS Position
CSS Z-index
CSS Overflow
CSS Float
CSS Inline-block
CSS Align
CSS Combinators
CSS Pseudo-class
CSS Pseudo-element
CSS Opacity
CSS Navigation Bar
CSS Dropdowns
CSS Image Gallery
CSS Image Sprites
CSS Attr Selectors
CSS Forms
CSS Counters
CSS Website Layout
CSS Units
CSS Specificity
CSS !important
CSS Math Functions

CSS Advanced

CSS Rounded Corners
CSS Border Images
CSS Backgrounds
CSS Colors
CSS Color Keywords
CSS Gradients
CSS Shadows
CSS Text Effects
CSS Web Fonts
CSS 2D Transforms
CSS 3D Transforms
CSS Transitions
CSS Animations
CSS Tooltips
CSS Style Images
CSS Image Reflection
CSS object-fit
CSS object-position
CSS Masking
CSS Buttons
CSS Pagination
CSS Multiple Columns
CSS User Interface
CSS Variables
CSS Box Sizing
CSS Media Queries
CSS MQ Examples
CSS Flexbox

CSS Responsive

RWD Intro
RWD Viewport
RWD Grid View
RWD Media Queries
RWD Images