

CSS challenge #5

POSITION INTRO

5 Flavours of position

MDN: The [position](#) CSS property sets how an element is positioned in a document.

The [top](#), [right](#), [bottom](#), and [left](#) properties determine the final location of positioned elements.

Bron: [position op MDM](#)

Filmpje: [youtu.be/jx5jml0UIXU](#)

position: static

sample: codepen.io/shoof/pen/zYyxPZr

De standaard waarde voor **position** is **static**. De positie van een element wordt gebaseerd op de plaats in HTML. Verder niets bijzonders. Dit wordt ook wel flow lay-out genoemd.

Section

1e div

2e div

3e div

```
div:nth-of-type(2) {  
  position: static;  
}
```

Nb. Static is de standaard waarde voor position. Die hoeft je dus niet op te geven. Dat is mooi, want de meeste elementen moeten gewoon op de plek staan waar ze staan.

5 Flavours of position

MDN: The [position](#) CSS property sets how an element is positioned in a document.

The [top](#), [right](#), [bottom](#), and [left](#) properties determine the final location of positioned elements.

Bron: [position op MDM](#)

Filmpje: [youtu.be/jx5jml0UIXU](#)

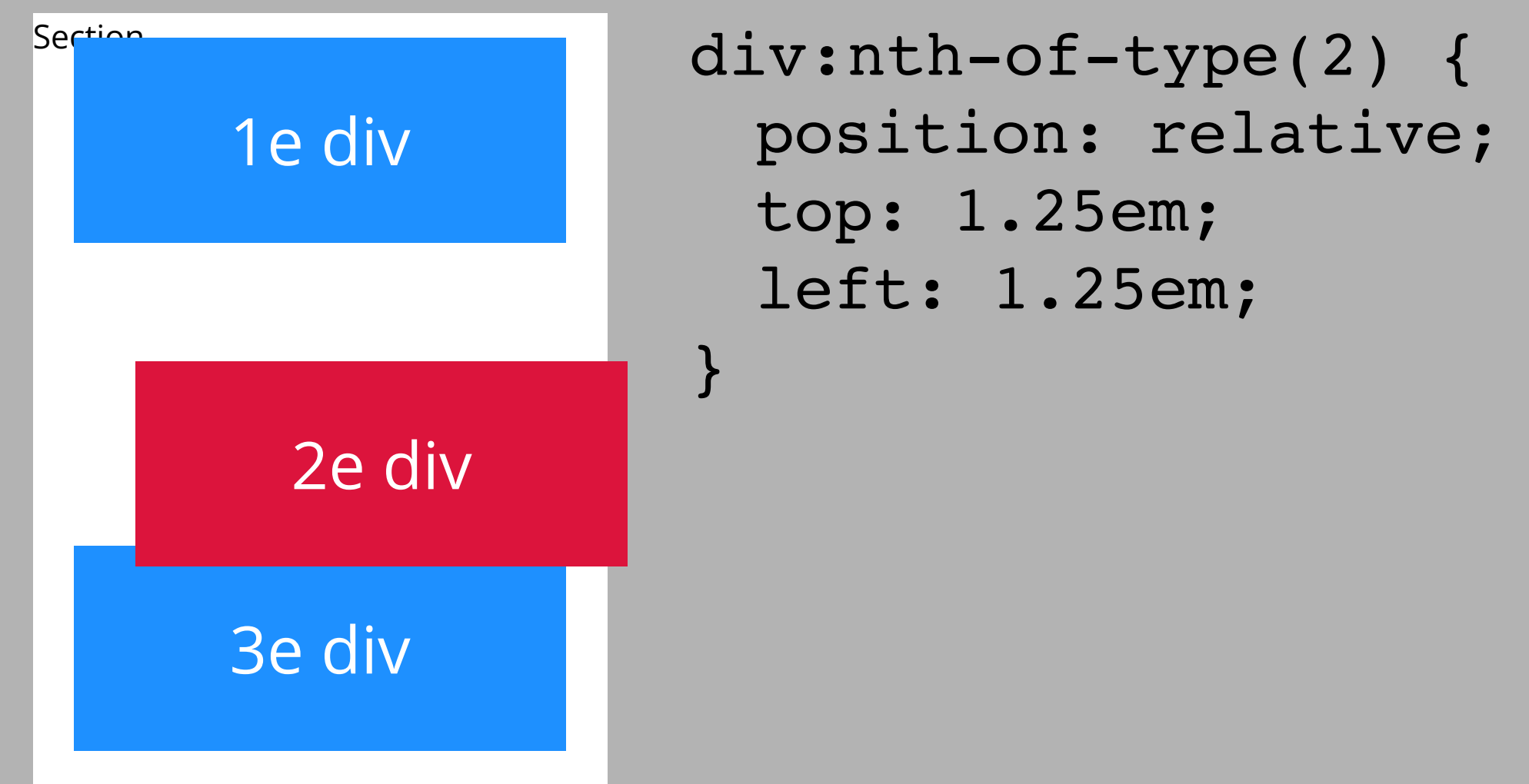
Niet te vaak gebruiken

Je kunt beter **translate (of transform)** gebruiken om een element te verplaatsen.

position: relative

sample: codepen.io/shoof/pen/bGONYqQ

Met **position relative** behoudt een element zijn plek in de flow. Vanaf die plek kun je het element met `top`, `right`, `bottom` en `left` verplaatsen.



Nb. Daar wordt `position relative` eigenlijk niet meer voor gebruikt. Dat kun je beter met `translate` doen.

Nb. `Position relative` wordt nog wel veel gebruikt als basis om kinderen in het element te kunnen positioneren (zie volgende slide).

5 Flavours of position

MDN: The [position](#) CSS property sets how an element is positioned in a document.

The [top](#), [right](#), [bottom](#), and [left](#) properties determine the final location of positioned elements.

Bron: [position op MDM](#)

Filmpje: youtu.be/jx5jml0UIXU

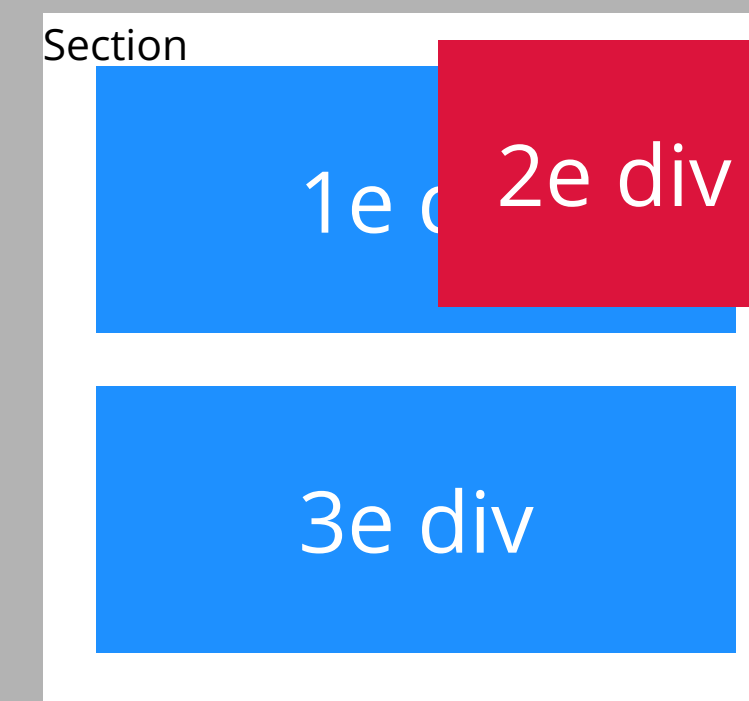
Niet te vaak gebruiken

Meestal kun je beter **flex** of **grid** gebruiken om te lay-outen.

position: absolute

sample: codepen.io/shooft/pen/oNJgoWW

Met **position absolute** verliest een element zijn plek in de flow. Het element wordt met `top`, `right`, `bottom` en `left` gepositioneerd in de eerste parent (die zelf gepositioneerd is).



Nb. Het element krijgt de breedte die de content inneemt (ook block elementen).

```
<!-- parent -->
section {
  <!-- gepositioneerd,
  childs worden hierdoor
  binnen het element
  gepositioneerd -->
  position: relative;
}
<!-- child -->
div:nth-of-type(2) {
  position: absolute;
  top: .75em;
  right: 0em;
}
```

5 Flavours of position

MDN: The [position](#) CSS property sets how an element is positioned in a document.

The [top](#), [right](#), [bottom](#), and [left](#) properties determine the final location of positioned elements.

Bron: [position op MDM](#)

Filmpje: [youtu.be/jx5jml0UIXU](#)

Niet te vaak gebruiken

Meestal kun je beter **flex** of **grid** gebruiken om te lay-outen.

position: fixed

sample: codepen.io/shoof/pen/poqvdwX

Met **position fixed** verliest een element zijn plek in de flow. Het element wordt met `top`, `right`, `bottom` en `left` gepositioneerd in de viewport. Het element blijft op die vaste plek staan als de gebruiker scrolt (en neemt dus altijd ruimte in).

Section

1e div

3e div

```
div:nth-of-type(2) {  
  position: fixed;  
  bottom: .75em;  
  right: .75em;  
}
```

Nb. Het element krijgt de breedte die de content inneemt (ook block elementen).

2e div

5 Flavours of position

MDN: The [position](#) CSS property sets how an element is positioned in a document.

The [top](#), [right](#), [bottom](#), and [left](#) properties determine the final location of positioned elements.

Bron: [position op MDM](#)

Filmpje: youtu.be/jx5jml0UIXU

position: sticky

sample: codepen.io/shoof/pen/JjwoOyE

Met **position sticky** behoudt een element zijn plek in de flow. Maar op het moment dat het element uit beeld schuift blijft het element *plakken* aan de rand (tot de parent helemaal uit beeld is verdwenen).

rand van het scherm

Section

1e div

2e div

3e div

Op zijn normale plek.

Plakken als het element uit beeld schuift.

Zelf ook uit beeld als de parent uit beeld schuift.

```
section {
  position: relative;
}
div:nth-of-type(2) {
  position: sticky;
  top: 0em;
}
```

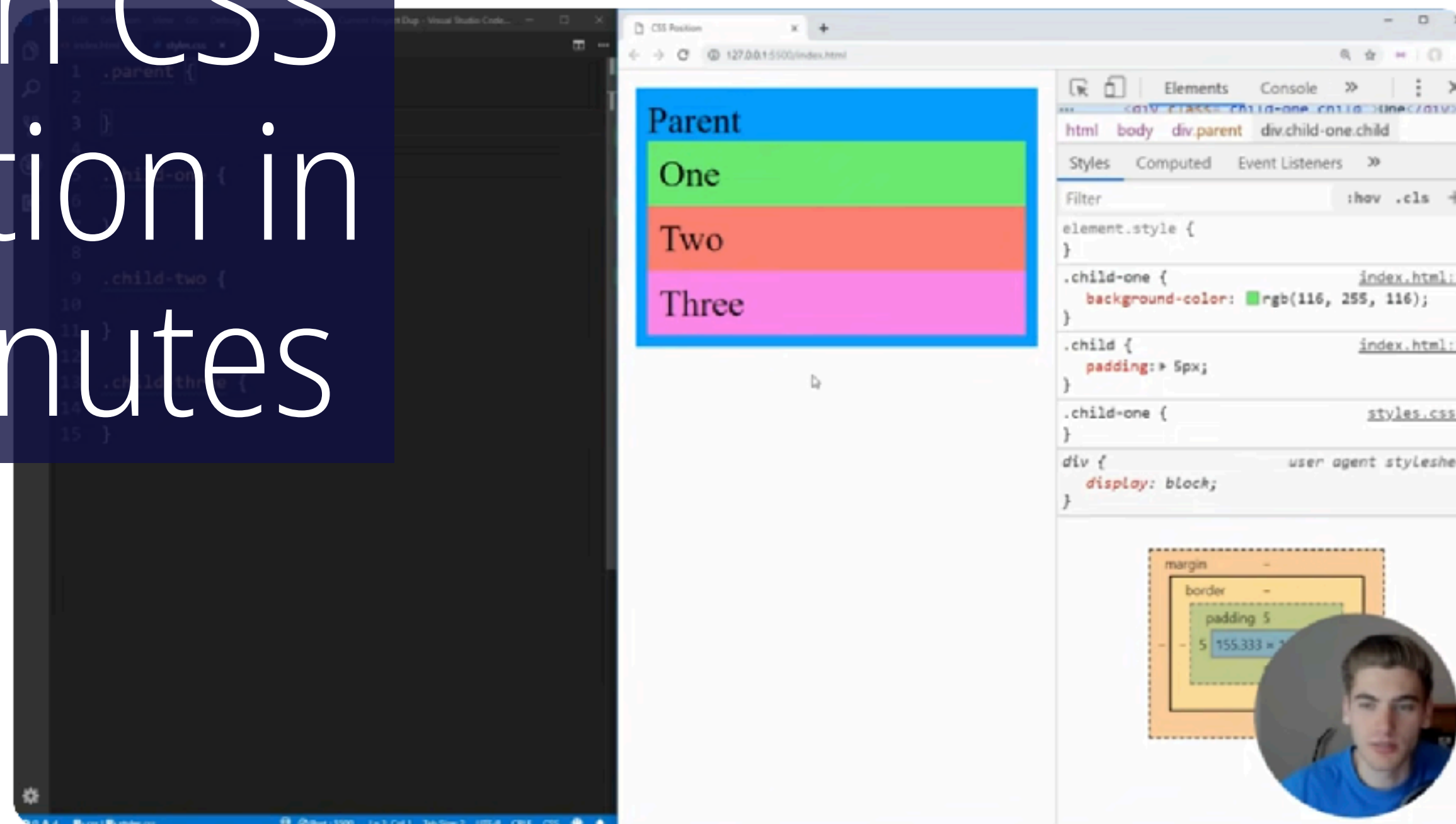



Learn CSS Position In 9 Minutes

YouTube

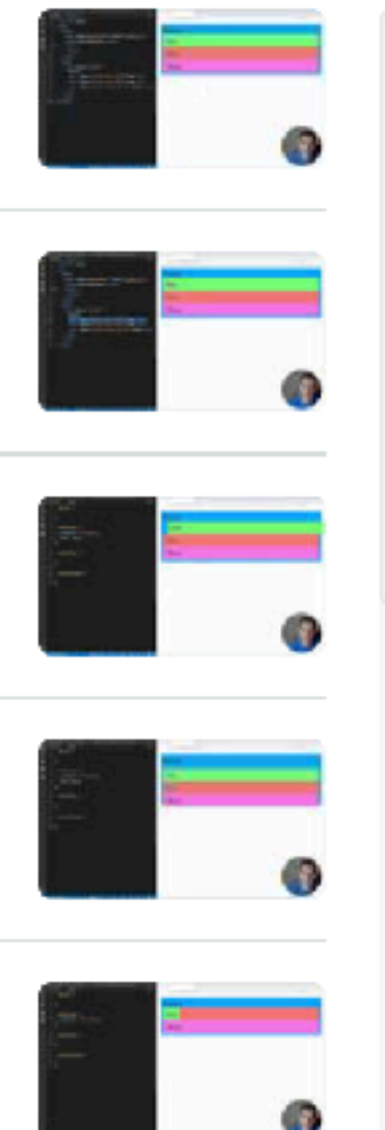
Learn CSS Position in 9 Minutes

YouTube · Web Dev Simplified · 30 apr 2019



In deze video

- 00:20 Example
- 01:08 Relative Position
- 01:48 Relative
- 03:18 Absolute
- 04:10 Position Absolute



Gerelateerde onderwerpen



HTML element

- 🔍 CSS positioning tutorial
- 🔍 Learn CSS position
- 🔍 How to change position in CSS
- 🔍 CSS positions examples


Web Dev Simplified:
youtu.be/jx5jml0UIXU



Handling Text Over Images in CSS

Handling Text Over Images in CSS

Mar 23, 2021



Build apps
monday app marketplace

Monetize apps on
monday.com marketplace:
start building and reach
180K+ customers.

ADS VIA CARBON

You might come across a UI component that has text above an image. In some cases, the text will be hard to read depending on the image being used. There are some different solutions like adding a gradient overlay, or a tinted background image, text-shadow, and others. I got encouraged to write this article after seeing [this tweet](#) from Addy Osmani.

In this article, I will explore the different approaches and solutions for this problem, and how to communicate the UI with a front-end developer to make sure that it's implemented as per the design mockup since some details can be easily missed in CSS.

Introduction

Each solution is supposed to solve a problem. Let's explore the problem for our case. When designing a component that has text above an image, we should take care of making the

Ahmad Shaheed

ishadeed.com/article/handling-text-over-image-css

Welcome to Learn CSS!

Box Model

Selectors

The cascade

Specificity

Inheritance

Color

Sizing Units

Layout

Flexbox

Grid

Logical Properties

Spacing

Pseudo-elements

Pseudo-classes

Borders

Shadows

Focus

Z-index and stacking contexts

Functions

Gradients

Animations

Filters

Blend Modes

Z-index and stacking contexts (advanced)

Z-index and stacking contexts



Was this helpful?

On this page

[Z-index](#)

[Negative z-index](#)

[Stacking context](#)

[Creating a stacking context](#)

[Resources](#)

The CSS Podcast - 019: z-index and stacking contexts

▶ 0:00 / 19:48

Say you've got a couple of elements that are absolutely positioned, and are supposed to be positioned on top of each other. You might write a bit of a HTML like this:

```
<div class="stacked-items">
  <div class="item-1">Item 1</div>
  <div class="item-2">Item 2</div>
</div>
```

But which one sits on top of the other, by default? To know which item would do that, you need to understand z-index and stacking contexts.

Z-index

The **z-index** property explicitly sets a layer order for HTML based on the 3D space of the browser—the Z axis. This is the axis which shows which layers are closer to and further from you. The vertical axis on the web is the Y axis and the



INTRO POSITION

CSS challenge #2