

Tailwindcss - A great CSS system that you have to be able to do in the 2020's!

This document belongs to the currently best Tailwind CSS course in the world:

[>> CLICK HERE TO HAVE A LOOK <<](#)

Tailwind CSS- deeper knowledge

[Define own breakpoints and own container class](#)

[Typography Plugin for Tailwind CSS](#)

[Configure the typography plugin](#)

[Object-Fit \(Cover\)](#)

[Go to Top](#)

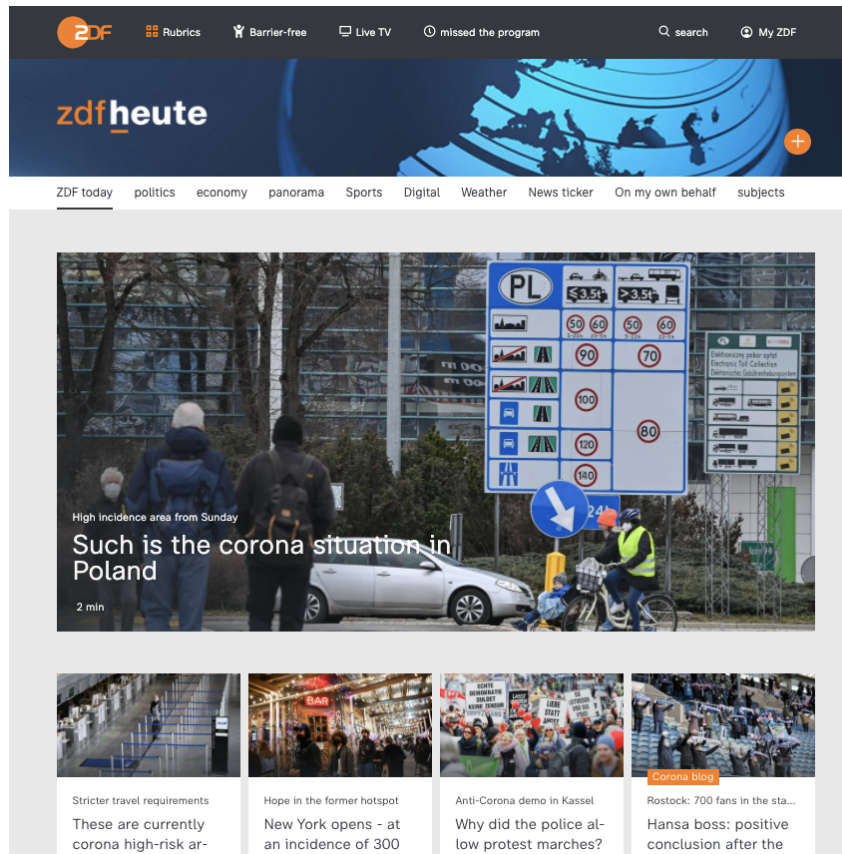
Copyright: Martin Krebs Eberth, mkrebseberth@gmail.com

Copying and dissemination via original PDF is permitted, printing or dissemination, even in excerpts, without naming the [Tailwindcss course](#) - see header - is prohibited.

Tailwindcss - A great CSS system that you have to be able to do in the 2020's!
This document belongs to the currently best Tailwind CSS course in the world:
[>> CLICK HERE TO HAVE A LOOK <<](#)

Define own breakpoints and own container class

Analysis of the (german) news portal "www.heute.de"



On the news site "heute.de" we can see the following:

- Breakpoints with layout changes at 640px and 1,000px
- Container with a maximum width of 1,140px, with resolutions below: full width.

We do this in 4 steps as follows:

Step 1: New configuration file

1. We save the "*company_styles.js*" from the last section of the course as "*news_styles.js*".
2. We include this in the *tailwind.config.js*: `require('./news_styles')`
3. We save the *index.html* from the last section of the course as "*news_example.html*".

[Go to Top](#)

Copyright: Martin Krebs Eberth, mkrebseberth@gmail.com

Copying and dissemination via original PDF is permitted, printing or dissemination, even in excerpts, without naming the [Tailwindcss course](#) - see header - is prohibited.

Tailwindcss - A great CSS system that you have to be able to do in the 2020's!

This document belongs to the currently best Tailwind CSS course in the world:

[>> CLICK HERE TO HAVE A LOOK <<](#)

Step 2: Define new breakpoints

In the **news_styles.js** we define the **breakpoints**: (under "theme")

```
screens: {  
  'tablet' : '640px',  
  'computer' : '1000px',  
},
```

Result: We have our breakpoints (don't forget npm) on the page:

screen: Computer

Step 3: Define a new container

Unfortunately, the standard container class is so deeply anchored in Tailwind CSS that we have to define it in a special way.

What we want is that with a resolution < "Computer" the container takes up the full width, after that we want to have a width of 1140px.

We achieve this by doing 2 steps in our configuration file:

1. Completely deactivate the standard container class of Tailwind CSS within **corePlugins**: (in tailwind.config.js or the new "news_styles.js")

```
module.exports = {  
  corePlugins: {  
    container: false,  
  },  
}
```

2. By using the following snippet under "**plugins [...]**" in the outsourced **news_styles.js**:

```
plugins: [  
  function ({ addComponents }) {  
    addComponents({  
      '.container': {  
        maxWidth: '100%',  
        '@screen desktop': {  
          maxWidth: '1140px',  
        },  
      },  
    })  
  }  
]
```

[Go to Top](#)

Copyright: Martin Krebs Eberth, mkrebseberth@gmail.com

Copying and dissemination via original PDF is permitted, printing or dissemination, even in excerpts, without naming the [Tailwindcss course](#) - see header - is prohibited.

Tailwindcss - A great CSS system that you have to be able to do in the 2020's!

This document belongs to the currently best Tailwind CSS course in the world:

[>> CLICK HERE TO HAVE A LOOK <<](#)

I found this solution on this website:

<https://stefvanlooveren.me/blog/custom-container-width-tailwind-css>

You can find more about other possibilities to use “plugins” here in the Tailwind documentation:

<https://tailwindcss.com/docs/plugins>

Step 4: center the container

In ***tailwind.css***:

```
@layer components {  
  .container {  
    @apply mx-auto;  
  }  
}
```

[Go to Top](#)

Copyright: Martin Krebs Eberth, mkrebseberth@gmail.com

Copying and dissemination via original PDF is permitted, printing or dissemination, even in excerpts, without naming the [Tailwindcss course](#) - see header - is prohibited.

Tailwindcss - A great CSS system that you have to be able to do in the 2020's!

This document belongs to the currently best Tailwind CSS course in the world:

[>> CLICK HERE TO HAVE A LOOK <<](#)

Typography Plugin for Tailwind CSS

With the standard installation of Tailwind CSS, the so-called "Preflight" plug-in takes effect, which completely resets all text styles, such as headings and list elements.

In many cases it would be easier if at least the basic HTML elements were styled.

The **official Tailwind CSS typography plugin** is available for this.

Here are the most important links for this plugin:

[Typography Plugin on GitHub](#)

[Example page for the plugin](#)

Please create the folder + subfolder **/sandbox/basic_elements** in your course installation and in there you create an empty file: **typography_plugin.html**

Now please copy the text from the file **"typography_plugin.txt"** which is attached to this lecture into this empty HTML file (typography_plugin.html)

When we open this page with our Visual Studio Code Live Server, we see a completely unstyled page:

Typographie-Plugin Eine Ebene hoch

Until now, trying to style an article, document, or blog post with Tailwind has been a tedious task that required a keen eye for typography and a lot of complex custom CSS. By default, Tailwind removes all of the default browser styling from paragraphs, headings, lists and more. This ends up being really useful for building application UIs because you spend less time undoing user-agent styles, but when you *really* are just trying to style some content that came from a rich-text editor in a CMS or a markdown file, it can be surprising and unintuitive.

We get lots of complaints about it actually, with people regularly asking us things like:

Why is Tailwind removing the default styles on my h1 elements? How do I disable this? What do you mean I lose all the other base styles too?

We hear you, but we're not convinced that simply disabling our base styles is what you really want. You don't want to have to remove annoying margins every time you use a p element in a piece of your dashboard UI. And I doubt you really want your blog posts to use the user-agent styles either — you want them to look *awesome*, not *awful*.

The @tailwindcss/typography plugin is our attempt to give you what you *actually* want, without any of the downsides of doing something stupid like disabling our base styles.

It adds a new prose class that you can slap on any block of vanilla HTML content and turn it into a beautiful, well-formatted document:

```
<article class="prose">
  <h1>Garlic bread with cheese: What the science tells us</h1>
  <p>
    For years parents have espoused the health benefits of eating garlic bread with cheese to their
    children, with the food earning such an iconic status in our culture that kids will often dress
    up as warm, cheesy loaf for Halloween.
  </p>
  <p>
    But a recent study shows that the celebrated appetizer may be linked to a series of rabies cases
    springing up around the country.
  </p>
  <!-- ... -->
</article>
```

For more information about how to use the plugin and the features it includes, read the documentation.

What to expect from here on out

What follows from here is just a bunch of absolute nonsense I've written to dogfood the plugin itself. It includes every sensible typographic element I could think of, like **bold text**, unordered lists, ordered lists, code blocks, block quotes, and even *italics*.

It's important to cover all of these use cases for a few reasons:

We want everything to look good out of the box.

Really just the first reason, that's the whole point of the plugin.

Here's a third pretend reason though a list with three items looks more realistic than a list with two items.

Now we're going to try out another header style.

Typography should be easy

Now we're going to install and activate the typography plugin:

[Go to Top](#)

Copyright: Martin Krebs Eberth, mkrebseberth@gmail.com

Copying and dissemination via original PDF is permitted, printing or dissemination, even in excerpts, without naming the [Tailwindcss course](#) - see header - is prohibited.

Tailwindcss - A great CSS system that you have to be able to do in the 2020's!

This document belongs to the currently best Tailwind CSS course in the world:

[>> CLICK HERE TO HAVE A LOOK <<](#)

Installation: `npm install @tailwindcss/typography`

Integration in the tailwind.config.js:

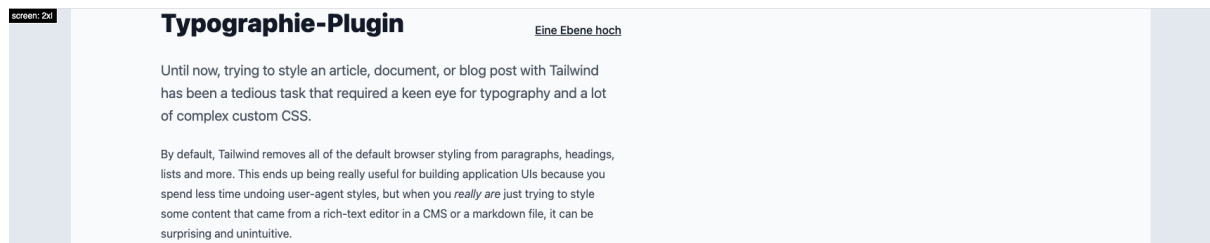
```
plugins: [  
  require('tailwindcss-debug-screens'),  
  require('@tailwindcss/typography'),  
],
```

Use in the document

The element within the plugin is supposed to be used just gets the class “**prose**”:

```
<div id="my_content" class="prose">  
  <p class="lead prose">Until now, trying to style an article, document, or blog  
  <h1>Garlic bread with cheese: What the science tells us</h1>  
  <p>
```

If we now reload the page, everything looks much better, but the content is too narrow:



This can be fixed by adding the “**max-w-min**” class.

Different font sizes

Instead of the “prose” class, you should always(!) use **prose-sm**, **prose-lg**, **prose-xl** and **prose-2xl**.

This changes the font size but also the font styles.

It might make sense to say that the text is a little larger on small screens than on large screens (for all reading glasses users who do not have reading glasses on for mobile phone display) - so you could say, for example:

```
<div class="container prose prose-xl tablet:prose-lg desktop:prose-sm max-w-min
```

Different link colors

Via “prose-COLOR” you can also color text links, in our case **e.g. prose-red**, it looks like this:

and the features it includes, [read the documentation](#).

[Go to Top](#)

Copyright: Martin Krebs Eberth, mkrebseberth@gmail.com

Copying and dissemination via original PDF is permitted, printing or dissemination, even in excerpts, without naming the [Tailwindcss course](#) - see header - is prohibited.

Tailwindcss - A great CSS system that you have to be able to do in the 2020's!

This document belongs to the currently best Tailwind CSS course in the world:

[>> CLICK HERE TO HAVE A LOOK <<](#)

General definition in tailwind.css

When you have made the right settings, you should of course **move** them to **tailwind.css**.

However, I would STRONGLY advise that if you use "prose" in the container class, to **bind it to another class** - e.g. "typo", so that you can also use container blocks WITHOUT the typography plugin:

```
.typo {  
  @apply prose prose-red prose-xl tablet:prose-lg desktop:prose-sm max-w-min;  
}
```

[Go to Top](#)

Copyright: Martin Krebs Eberth, mkrebseberth@gmail.com

Copying and dissemination via original PDF is permitted, printing or dissemination, even in excerpts, without naming the [Tailwindcss course](#) - see header - is prohibited.

Configure the typography plugin

Here you can see the default settings of the typography plug-in:

<https://github.com/tailwindlabs/tailwindcss-typography/blob/master/src/styles.js>

Here you can see the default settings of the typography plug-in:

<https://github.com/tailwindlabs/tailwindcss-typography#customization>

Step 1: Reduce font-weight for h1,h2 (in tailwind.config.js)

```
extend: {  
  typography: {  
    DEFAULT: {  
      css: {  
        h1: 'font-weight: 600',  
        h2: 'font-weight: 600',  
      },  
    },  
  },  
},
```

Step 2: Reduce font-size for h3, **only in prose-xl**

```
extend: {  
  typography: {  
    DEFAULT: {  
      css: {  
        h1: 'font-weight: 600',  
        h2: 'font-weight: 600',  
      },  
    },  
    xl: {  
      css: {  
        h3: {  
          fontSize: '1.2em'  
        },  
      },  
    },  
  },  
},
```

Important: Please be aware, that for the **prose-xl** option the **setting for h3 is an object!**

continues on the next page..

[Go to Top](#)

Tailwindcss - A great CSS system that you have to be able to do in the 2020's!

This document belongs to the currently best Tailwind CSS course in the world:

[>> CLICK HERE TO HAVE A LOOK <<](#)

Step 3: Darker Gray for and

```
ol > li::before {  
  @apply text-gray-700 !important;  
}
```

```
ul > li::before {  
  @apply bg-gray-800 !important;  
}
```

[Go to Top](#)

Copyright: Martin Krebs Eberth, mkrebseberth@gmail.com

Copying and dissemination via original PDF is permitted, printing or dissemination, even in excerpts, without naming the [Tailwindcss course](#) - see header - is prohibited.