

CSS Tutorial

IDE/DiXiT Spring School 2015

This tutorial will cover the step by step process of designing a CSS which generates the layout of our example website. Follow the steps, remember to regularly save your progress and check your declarations in a web browser as you move on.

Within the brackets, you will find suggestions based on the values used in our example website. If you feel comfortable enough, feel free to experiment with the application of other values.

- (1) First of all, reset the margins and paddings of all elements

```
* {  
  margin: 0;  
  padding: 0  
}
```

- (2) Basic layout

Let's start with the design of the basic layout for **index.html**, **edition.html**, **about.html**: the default text color, the default font-family, the background colors (shaded in gray), paddings and margins.

- html (#c0b7a8)
- body (#f8fbef; #333333; Georgia, Times, serif)
- header (#32649c; padding top and bottom of 20px)
- footer (#c0b7a8; padding top and bottom of 20px)
- section with the id "content" (margin bottom and margin top of 30px; a minimum height of 500px for pages with less content)

- (3) Width and alignment

For the website we want to define a width of 960 pixels and position the content in the middle of the browser window. However, the background of the header, of the navigation and of the footer should run across the whole browser window. Thus, you need the child selector to assign 960px just to the child elements of header, footer and nav.

Use the value *0 auto* for the property *margin* to center the entire content of a page: the assigned width will be applied to the defined elements, the remaining margins are added equally on either side. For our example the following definition is needed:

```

section#content,
header > *,
footer > div,
nav > * {
    margin: 0 auto;
    width: 960px
}

```

(4) Navigation

By default, the browser displays an unordered list as vertical list with preceding bullet points. We'd like to have a horizontal navigation.

- nav (#c0b7a8, padding to the top and to the bottom of 10px)
- li (display as inline)

(5) Image floating

Let the image of the **index.html** float to the left, to display text and image side by side

- figure (floating to the left; a solid border with 1px in grey; padding of 10px, margin to the right of 10px)

The images of the edition text (**edition.html**) should also float to the left. Keep in mind that the images of the edition are nested within a <div> element with the class "page". Find a fitting selector to float and resize the image.

- img (floating to the left, width of 300px)

The edition text is longer than the image, this means that the text floats around the image. If you want to have your text and image side-by-side, you have to assign a left-margin to the text, of the size of the image and the desired gap between text and image. The same applies to the <h5> element.

- div (a left margin of 320px)
- h5 (a left margin of 320px)

(6) Text styling

The headlines (h1-h5) and the paragraph need to be styled:

- it's up to you to define the color, the font-family, the font-size and margins
- <h5> should be aligned to the right
- p (top margin of 10px)

The edition text (**edition.html**) contains several elements with different @class attributes. Make appropriate definitions for the classes:

- uppercase
- red
- blue
- initial (larger font-size then the default size)
- place (e.g. background-color, font-style)

- name (e.g. background-color, font-style)
- date (e.g. background-color, font-style)

(7) Links

By default the browser displays hyperlinks as blue and underlined. With pseudo classes you can change the conditions of the hyperlinks:

- link, active (#DF7401, underline)
- hover (red)
- visited (#aaaaaa, *no text-decoration*)

The navigation also contains <a> elements, but the links should have a different appearance. In addition to your default definition for hyperlinks, you can add another definition for the navigation. Therefore, you need a descendant selector. *Hint: nav ul li a:link*

- link, visited (black; #dfd7ca; arial, helvetica, sans-serif; no text-decoration, padding top and bottom: 3px, right and left: 8px)
- hover, active (white background)

(8) Multiple columns

Define a two-column layout for **about.html**. A <section> is nested within another <section>, which has the @id "content". The nested <section> should build the left column and the <aside> element should build the right column. Find a fitting selector to address the nested section. Look at the example in the presentation (descendent selector).

- the nested section (floating to the left, bottom margin of 20px, width of 620px)
- aside (floating to the left, margin to the left of 20px, width of 300px)

Have fun and do not hesitate to play around!