WEB DEVELOPMENT

Practicum - Week 2.1

Exercise guidelines

🗁 Folder structure

For this exercise, **we will work in the folder called week 2.1**. in your repository folder. Only put the files necessary to complete the exercise in here.

DO NOT add any source materials in this folder (screenshots, pdf files, ...)

🗁 Submission deadline

The deadline for this exercise is **Sunday, October 3rd, 23:59.**

🗁 Submission rules

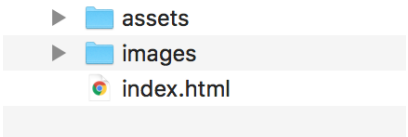
Submission rules are listed in the week 1.1 briefing

Final result

The **folder "final result"** contains an image of the expected end result. Any intermediary steps are not obligatory to be completed in that order, but serve as additional help only.

Setting up basic structure and convert text to HTML

1. Create the following folder structure **within folder week2.1**



The **assets** folder holds all elements that contribute to the website's functionality, but are not part of the server-side code. These are **CSS files, JavaScript files, images that belong to the design**, ...

The **images** folder holds all **content images.** Consult the lecture session for the difference between content and CSS images.

2. Convert the text in **plain text.txt** to the correct HTML elements. You will be needing these semantic tags now!
   Once again: an element's style is of no importance, we will change all of it through CSS

The overpowering "my-word-where-do-I-begin"-feeling is completely normal at this stage. Unfortunately, there is no better way to teach you than to throw you in the deep end. Bit by bit, things will unravel and you will learn.

Remember the lecture: **take the screenshot of the final product and start by marking out the areas you recognise.** Draw them on paper or in a digital programme and start with those. Skip what you don't know for the time being.

If you're already thinking of ID or classnames, **mark them down** on that page. Also: try looking for similar elements. Small differences can leverage CSS' cascading functionality by applying the same styles to everything and only differentiating where necessary (e.g. by adding a class or using a different selector).

Use this page as a master plan to guide you through the process of creating the website.

Make sure your result resembles that of intermediate-result-1.png (source files) before moving to the next step

Adding the CSS

1. Create a CSS file for the reset CSS. Call it **reset.css and place it in the assets/css folder**. Paste the contents of the reset CSS of your choice (suggestion: https://meyerweb.com/eric/tools/css/reset/)
2. Create another CSS file, this time to host your own rules. Call it **screen.css and add it to the same folder**
3. **Link both documents** to your HTML file. Mind the order!
4. Check if coupling the two CSS files worked, e.g. by changing the page's background colour.
   Remember, the most commonly used properties can be found on https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Properties_Reference

Structuring your CSS

Don't forget to add structure to your CSS files, do not start coding blindly. This is a mandatory part of the course, failure to do so will reflect negatively upon your end result.

Structure proposition:

1. Start by styling the "general elements". These are rules that you apply throughout the website (text colour, font size, ...)
   Use only element selectors for this. Most commonly, you will need **body, h1, h2, h3 (and further headings if used), p, a, strong** as a very basic specification.

   Remember: you can overwrite given rules for specific areas of the page. It's better to give a general style to all and overwrite one or two rules, than copy-pasting the majority of applied styles.
2. Afterwards, try grouping the "**building blocks**" for the website, the basic elements that make up your layout, the grand structure. The **header, footer, container, banner bars, ... but no page specific elements**
3. Finally, add th**e page specific elements** page by page

Always write your most general rule first and your most specific rule more below.

The proposed structure can be found in the document **CSS structure.txt** as CSS comments. **Copy this in your CSS document to use as a point of entry.**

Now start coding!

You are armed with a plan and a structure to guide you.

Start with small components

Try finishing the header first or perhaps the footer makes more sense or the articles stand out to you. Start with what you know, search for what you don't later.

There are additional screenshots that show you how the page evolves as the CSS evolves, though this is just an indication. You do not need to follow the same order as the indicative screenshots.

See screenshots in source material

Tips and tricks:

- The header spans across the entire width of the page, regardless of the size of your screen
- You can center a block level element by **giving it a width** and **setting its left and right margins to 'auto'**. The width used for the containing element in this exercise is **60%**.
- Remember: all dimensions, font sizes, border widths, ... are specified in **relative values! Do not use absolute units (px, pt, ...)**
  Whenever a value is not specified, try approximating.
- Leverage **cascading and auto-inheritance** to the best of your ability. Get to know those properties that auto-inherit and those that don't
- **Only the first paragraph** of the news item gets the "line" to the side. The others don't
- The images are not clickable, but the tricky part is that all their sizes vary. They are however wider than their container, so we can leverage scaling to our advantage. A common technique is to apply **"max-width: 100%" to the image itself**, while **relying on the containing element** to fix the size. This allows for flexible layouts.

  *Careful: make sure your images are not *too* large for the medium you are serving, otherwise this handiness will cost you precious bandwidth. See Bachelor year 2 for more information on correct image distribution.*

Challenges:

- Clean HTML is good HTML. Try using as few classes and IDs as possible in every exercise. Only add them when it makes sense!
  This exercise can be completed without IDs and classes (or with a single class before viewing lecture 2.2 course material)

Used colours / fonts:

- Light grey: #999999
- Black: #000000
- Bordeaux: #D11B1C
- Font body text: Arial, Helvetica (fallback)
- "FilmFestivalOostende.be": font size 2 rem, bottom margin 2.5rem, 7.5rem in height, uppercase only
- "Ready for the windiest movie event of all?": font size 2.5rem, bottom margin 2.5rem
- Paragraphs: font size 0.8rem, line height 160%, bottom margin 1rem

- Images: 0.625rem bordeaux line just at the top, but only for every other image
- News item: 2.5rem bottom margin, 1.25rem space on this inside on each side
- Titles news items: font size 1.9rem, 1rem bottom margin
- Dates: font size 1.4rem, 0.5rem bottom margin
- First paragraph in news item: 0.5rem space from the left
- Colour for the “line” on news item with background pattern: #CCCCCC (different colour to when there is no background!)
  The line is 0.25 rem wide (in both cases)