WEB DEVELOPMENT

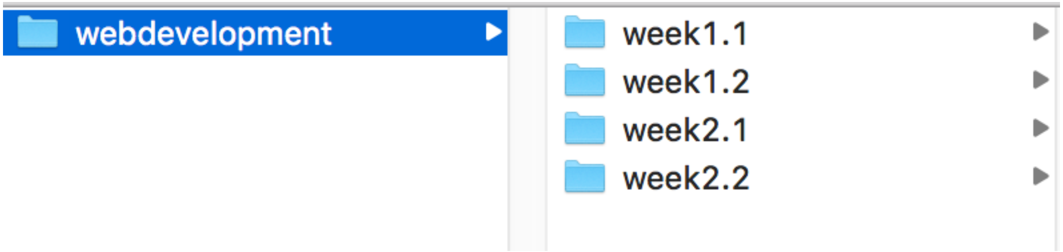
Practicum - Week 1.1

Exercise guidelines

In order to ensure that the practicum sessions run smoothly, we require a certain folder structure. Before starting an exercise series, **always** read the guidelines stated in the accompanying document

Folder structure

Create a folder "webdevelopment" on your HD which will hold subfolders per week according to the following pattern:



webdevelopment

week1.1
week1.2
week2.1
week2.2

No spaces and lowercase
Every new exercise, we will add a folder by this naming convention. The first few weeks have already been set up when you clone this repository, for your convenience. From week 4 onwards, you'll need to add the folders yourselves.

Take note that complying with the naming convention is imperative if you want to be graded. Our grading script only picks up correctly named folders.

Incorrect name == no grades.

Submission deadline

The deadline for this exercise is **Sunday, September 26th, 23:59.**

Submission rules

Submissions are handled through our own HOWEST gitlab, located at https://git.ti.howest.be/TI/2021-2022/s1/web-development/students
You will find your personal folder there. Follow the git configuration instructions on Leho if you have not already configured your IDE to work with the repository.

The final commit before the deadline will be treated as the handed-in version. Any additional commits will be neglected.

The same rules apply as if it were an examination: this is an **individual exercise**. Copying from another student, plagiarism, hacking or any other form of fraud is forbidden. If this is detected, this will be calculated in the final score and the persons involved will appear before the Examination Board.

Reference material

See lectures for course material and references to external sources.

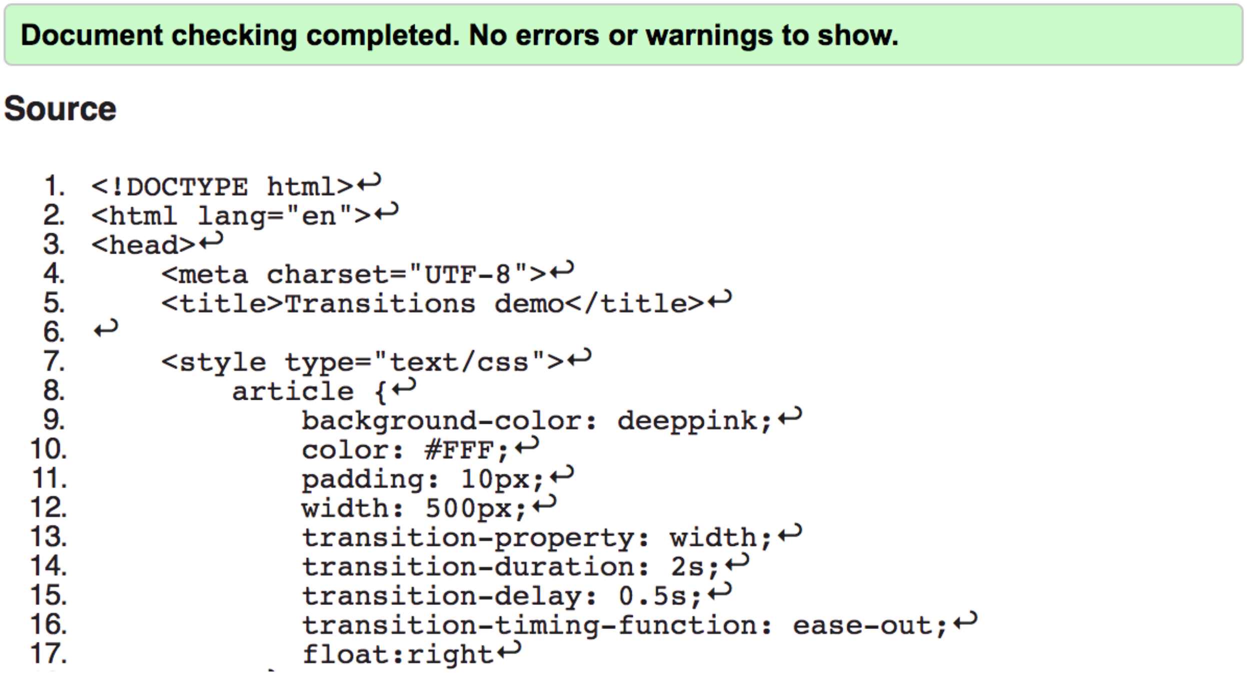
The **folder "final result"** contains an image of the expected end result. Any intermediary steps are not obligatory to be completed in that order, but serve as additional help only.

Validation

Remember: **all** HTML files need to be valid, always.
A single mistake to this principle (even if it's just one file) will result in a score of zero
Pay attention and validate often!

The only accepted validator is the official W3 validator on https://validator.w3.org/
We have a CI in place when you push your code, but that validator is less strict than the W3 validator. See the CI as an initial filter.

A page is **valid** when the **validator displays the green bar** (screenshot below)
Every other result is not valid (including warnings!)



Document checking completed. No errors or warnings to show.

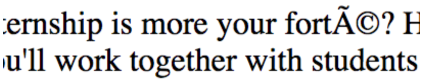
Source

```
<!DOCTYPE html>↩
<html lang="en">↩
<head>↩
    <meta charset="UTF-8">↩
    <title>Transitions demo</title>↩
↩
    <style type="text/css">↩
        article {↩
            background-color: deeppink;↩
            color: #FFF;↩
            padding: 10px;↩
            width: 500px;↩
            transition-property: width;↩
            transition-duration: 2s;↩
            transition-delay: 0.5s;↩
            transition-timing-function: ease-out;↩
            float:right↩
```

Basic HTML elements

1. Create an **HTML** page. What would be a good name for you first page?
2. Add the meta tag “author” in the head and fill out your full name. **From now on, add this step to all exercises**
   <meta name="author" content="Jill VandenDriessche" />
3. Transform the contents from the file **“plain text.txt”** to **html markup**

   ⓘ Be careful: style is irrelevant, semantics are primordial. Results will differ slightly cross browser, that’s not a problem at this point
4. **Validate** the page and correct mistakes if necessary
5. If you did not have any mistakes, **try and force one**. Try and understand the error message, this is the key to successful debugging
6. Give the page a **useful title**
7. When viewing the page in the browser, you might encounter strange symbols as depicted below



ernship is more your fortÃ©? H
u'll work together with students

Why is this? Or if you don’t have these, **why not**?

In any case, fix it. There are two ways to do this, apply the easiest way.

Nested lists

1. Make **another HTML page**. What will you call it?
2. Don’t forget both **title and meta tags**
3. Implement the text from the file **“plain text nested list.txt”** as a nested list. Mind the levels! Ensure that the semesters have numbers, but the courses do not.
4. **Validate** (and weep...)
5. Add **comments** through the file, marking the start of every semester

Linking

1. Consult the **Mozilla developer documentation (MDN)** to figure out how links work (https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a)
2. Link the first page to the second, by adding the text **“view course catalogue”**