Assignment 2: HTML continued, links and CSS introduction

Note: If you have not completed assignment 1, go ahead and do it now, as you need it for this assignment. You do not have to submit the code for assignment 1. Your code for assignment 1 will not be graded.

In this assignment you are going to style the website you created in the previous exercise.

Part 1: Housekeeping (10%)

Considering the vast amount of websites on the Internet it is important to differentiate yourself. One way of doing that is to change the style of your website to make it more appealing.

To prevent that the assets powering your website grows into a big ball of mud, some house keeping is needed. To keep things neatly packaged you are going to create two directories: `css` and `img`.

Move the image files to the `img` directory. Create an empty file called `style.css` and drop it inside the `css` directory.

The directory structure of your website will now look like the following.

```
.
├── css
│   └── style.css
├── about.html
└── img
    └── me.jpg
```

Part 2: Update (10%)

Now that your website directory structure is sorted out it is time to update your website.

Update the `img` element to point to the new location of the image of you.

Include the empty `style.css` file inside the `head` element of your `about.html` file.

Part 3: Start Styling (10%)

Now for the fun part! Add rules to your `style.css` file so that these criteria are met:

- The headers must have a web safe[1] font that is different from the default.
- Change the background color of the page. Make sure to consider readability when choosing font and colors.
- The fonts of the paragraphs must have a fallback font in case the client visiting your website does not have one of the fonts installed.
- The image must have a maximum width of 150 pixels.
- The image must have a border.

Part 4: Inline Styling (10%)

Although considered a bad practice [1], inlining CSS is not an uncommon way of styling elements on a webpage. Increase the border width around the image by 2 pixels *without* changing any CSS in your `style.css` file.

Part 5: More Housekeeping (20%)

Attached you will find a `zip` archive containing an article formatted in HTML. Extract the files to your assignment directory and organize them so that your assignment directory is structured like this:

```
.
├── css
│   ├── article.css
│   └── style.css
├── about.html
├── article.html
└── img
    └── cup-icon.svg
```

```
        └── me.jpg
        └── te.png
```

Upon opening `article.html` you will quickly notice that the Gods of crappy web design have smote you. Your job is to fix the article.

Update the path of all the style and image references so that all assets are loaded correctly. You can double check this by opening the Developer Console of your browser.

It can become cumbersome for your visitor to have to click the back-button every time he or she wants to get back to reading the article after clicking a link. To make this more convenient, make sure that all links pointing to any *external* websites opens in a new tab in the browser.

Part 6: Internal Linking (10%)

There is a bibliography at the bottom of the article. The references in the bibliography are being referred to from the text with footnotes. To make it easier for the reader to quickly see what a footnote is referring to, make the footnotes internally link to the corresponding reference in the bibliography.

Part 7: (20%)

To put the icing on the cake, so to speak, make the following changes in `article.css` :

- The cover image (of the teapot) must have the same width as the article body.
- The margin between the title and the byline should be removed.
- There should be a horizontal line separating the byline and the article body.
- The font has to be changed to a font intended for readability: Georgia or Verdana.
- The width of the article body should only accommodate 65-75 characters.
- The line-height must be 1.5 (150%).
- The font font size of the article body must be between 18px and 22px.
- The background color of the entire page must be be light gray.
- The font color must be dark gray.

Questions (10%)

1. Why does inline style CSS override rules defined in `style` elements and external stylesheets?
2. Give a brief example of when to use ID (`#id`) and when to use classes (`.class`) in CSS.
3. What does `RGBA` mean and what colors can you express with it?
4. Why do we include CSS files inside the `head` element and not inside the `body` element?
5. What CSS selector matches all the `p` elements inside the `article` element in the following HTML?

```
<p>This should not match.</p>
<article>
  <p>This should match.</p>
  <p>This should also match.</p>
  <p>Do not forget about this one!</p>
</article>
```

Footnotes

1. We consider the following fonts "web safe" in this course. `Verdana` , `Geneva` , `Georgia` , `Helvetica` , `Arial` , `Times New Roman` & `Courier` .

2. Maintaining a website riddled with style-attributes is difficult. It will also often lead to repeating code, violating the Don't repeat yourself (DRY) principle. Moreover, it prevents the browser from caching the CSS.

Deliverables

Submission should be uploaded as a zip file into It's Learning before the deadline. Submissions are ONLY accepted via Its Learning. We DON'T accept late assignments. Emails or any other messages with late assignments are automatically discarded without further communication