Group work and problem solving - Miniprojekt

Group work

Usually professional programming is group work, where you have to communicate with many people and also make your code fit other people's code. To make this process easier, projects use project structures like Waterfall or Agile structures.

1. Discuss the Waterfall and Agile project methods. What are the main differences between them?

2. Because the agile projects are more popular in the modern development, let's focus more on them. What are some important things that you have to remember in when working in agile project?

Problem solving

When you have been writing HTML and CSS, you probably have noticed that VS Code has features that help you to write better code. One of these features is that VS Code checks your code's syntax while you write it and informs you if there are some problems. This tool might not always pick up on every problem so you can also use a HTML validator to validate your HTML. One of these validators is W3C HTML validator show you errors and warnings in your HTML.

In these tasks you have to use the validator and VS Code to correct a small HTML file.

3. Upload the "problem-solving.html" file into the W3C HTML validator and check how many errors and warnings you get.

4. The validator tries to help you by giving the line where it sees the problem. Check the first warning the validator list and write in the answer the line in which the problem occurs.

5. The validator line directions can be sometimes bit different than you would expect. This happens because the validator is just pointing you to the part of the code where it sees the problem, but the actual problem might result from a different part of the code. That's why it is good to learn to look through the code around the problem if you don't find the solution straight away. Let's look at the problems 4 and 5 from the validator. Write in the answer the line that the validator gave for the problems and on which line you can actually correct these problems. You can also correct the HTML in your code editor.

6. We can now reupload the "problem-solving.html" to the validator and see that the problem 4 and 5 have disappeared since we corrected them by adding the angle bracket. Then look into the problem 9 and try to find out why this problem messes up the end of the HTML document. Correct the problem in the HTML file using VS Code.

7. Let's reupload the HTML file to the validator. Now if you take a look into problems 4, 7 and 8, you can see that there is a problem with our list element tags on lines 14, 16, 18 and 20. Correct these problems in the HTML file using VS Code.

8. If you again reupload the HTML file to the validator you can see that we have 9 problems left. Let's then look into problems 5 and 7. These are telling us that we have an error with strong tags. Solve these problems so that end of the first list element is bold, in the second list element "strong strong emphasised?" is bold and "strong emphasised? what is this?" is emphasised. Do this in a way so that there is no violation of the nesting rules of the tags.

   Hint: Nesting rules mean that if you put HTML tags inside each other, you have to close the nested ones before you can close the outer ones. For example like this `<strong><em></em></strong>` and not like this `<strong><em></strong></em>`

9. If you now again reupload the HTML file to the validator you can see that there are only three problems left. These are metadata errors.

   The first one is the warning about defining the language for our HTML file. Defining the language makes it easier for search engines to categorise you website and so isn't necessary but might make your website more accessible to the right audience if you add it.

   The second problem is about character encoding which defines what kind of characters or letters your website can use. If it's not defined it will default into windows-1252 encoding. This is important to define if you want to use something other than Latin alphabet.

   The last problem is about document type declaration. In the first line of every HTML5 document you are supposed to declare that the document is HTML5 document with `<!DOCTYPE html>`. This gives the browser information about what version of HTML was used to write the document and ensures that document is parsed the same way in different browsers.

   Fix these problems by adding the lang attribute with "en" as value to the html tag, defining the "utf-8" character set in the head of the document and adding the document type to the first line.

CSS validation

10. Take the "problem-solving.css" file and upload it to the W3 CSS validator. In "More Options" tab pick the "all" value for "Warnings:" and then click the "Check" button. How many errors and warnings do you see?

11. Sometimes there might be some mistyped property values that the code editor doesn't catch. Look at the first error and correct the value in the CSS file.

12. If you now reupload the CSS file to the validator there should be one less errors. Let's then look at the errors on the lines 9 and 10. Fix these errors in the CSS file.

    Hint: If you put your cursor in VS Code over the property names and click "Quick Fix", VS Code will help you to solve this.

13. Reupload the CSS file again to the validator. Let's look into the last two errors. Correct the "center" and "content-right" classes in the CSS file.

    Hint: Is there something weird about "display" in ".center"? ".content-right" might also look bit weird on a closer look.

14. If we reupload the CSS file now to the validator we can see that there is still one error. Validator tells us where it sees the error on line 26 and if we go to the code editor (e.g. VScode) we get more information. Fix the error using the code editor information.

    Hint: Look at the CSS rule that is set for the "h2" tag.

15. If you now last time reupload the CSS file to the validator you can see that there shouldn't be any errors, but there is still two warnings. From which lines are these warnings resulting and what are they trying to tell?