



# Tasks to Solve During Seminar 1

## Internet Applications, ID1354

**Write a document with your answers to the following tasks, and upload it to Canvas at the end of the seminar.** The format of the document is not important. The problems shall be solved by all group members together. You may also write the document together, but everyone must add their personal section describing what was learned during the seminar.

### Mandatory Task

- All members of the group shall, in turn, explain and motivate their program, and also their report, to the other group members. Write a brief summary of differences, and comment on advantages and disadvantages of the different solutions. Do not write about minor layout differences.
- Go through all requirements specified in the lab tasks, and evaluate that your web sites meet them. Also evaluate whether HTML and CSS files are well-structured and easy to understand.
- Evaluate that your reports meet the requirements in the report template and in the lab tasks.
- A good way to evaluate if the code is flexible and easy to understand is to actually try to change something. Select one or more of your web sites and try to change it *without the help of the author*. First try to change something relatively easy, like the color of an element that appears on all pages. If that succeeds, try something more difficult, for example to move the menu to the right or the left.

### Optional Tasks

Solve as many of these tasks as time allows, and write your solutions in the same document as the mandatory task. If the solution is a piece of code, just paste the code in the document, without bothering about figures or formatting. The tasks may be solved in any order, you do not have to start from task one. **These optional tasks do not improve your grade, as optional tasks solved *before* the seminar does.**



## Task 1, Selectors

CSS selectors are much more powerful than what is mentioned in the lecture notes. Take a look at combinators, pseudo-classes, attribute selectors and pseudo-elements at w3schools ([http://www.w3schools.com/css/css\\_combinators.asp](http://www.w3schools.com/css/css_combinators.asp), [http://www.w3schools.com/css/css\\_pseudo\\_classes.asp](http://www.w3schools.com/css/css_pseudo_classes.asp), [http://www.w3schools.com/css/css\\_attribute\\_selectors.asp](http://www.w3schools.com/css/css_attribute_selectors.asp), [http://www.w3schools.com/css/css\\_pseudo\\_elements.asp](http://www.w3schools.com/css/css_pseudo_elements.asp)) and test some complex selectors.

## Task 2, Assigning a Property Value

It is sometimes difficult to understand why a certain CSS property gets the value it gets. Try to get a good understanding of the cascade process by creating CSS rules with different specificities. This is covered on slide 65 in the lecture notes from lecture three. Also try to specify the **inherit** and **initial** property values. As a starting point, you can use the CSS and HTML in listings 1 and 2 below.

```
p#myId {
    color: red;
}

p.myClass {
    color: green;
}

body p {
    color: blue;
}
```

Listing 1: CSS code

```
<!DOCTYPE html>
<html>
  <head>
    <title>Question 4</title>
    <meta charset="UTF-8">
    <link rel="stylesheet" type="text/css" href="t1.css"/>
  </head>
  <body>
    <p id="myId" class="myClass">content</p>
  </body>
</html>
```

Listing 2: HTML code



### **Task 3, More CSS**

There is much more in CSS than what is covered in this course, for example 2D transforms, 3D transforms, animations, multi-column layout and much more. Look at a feature you find interesting, for example at [w3schools](http://w3schools.com), and write a short example style sheet.