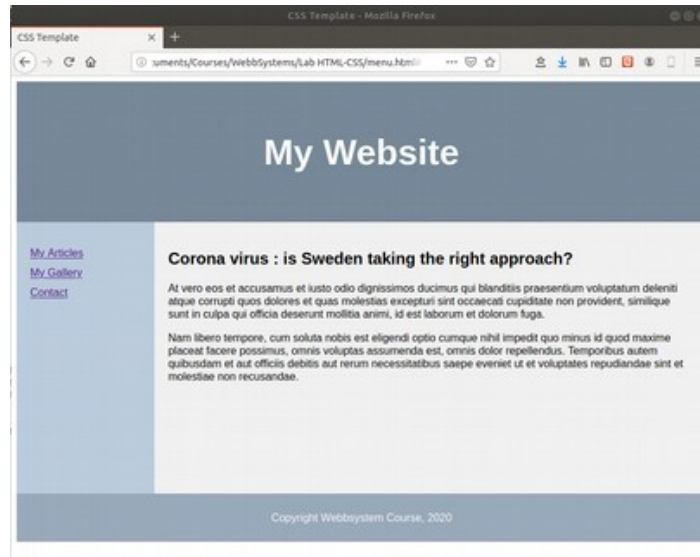


HTML-CSS Lab

The goal of this lab is to create, step by step, a simple web page using HTML and CSS, as illustrated below. This page should display correctly both on computer and mobile phones.



Once the lab is completed, keep your files in a safe place, we'll need them for the next labs!

Preparation

In a new empty folder, create the following files (using the text editor of your choice):

- menu.html
- page1.html
- page2.html
- page3.html
- mystyle.css

In all .html files, copy the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>My toy website</title>
  <meta charset="utf-8">
</head>

<body>

</body>
</html>
```

In the .css file, copy the following code, which sets a font family for the whole document:

```
body {  
    font-family: Arial, Helvetica, sans-serif;  
}
```

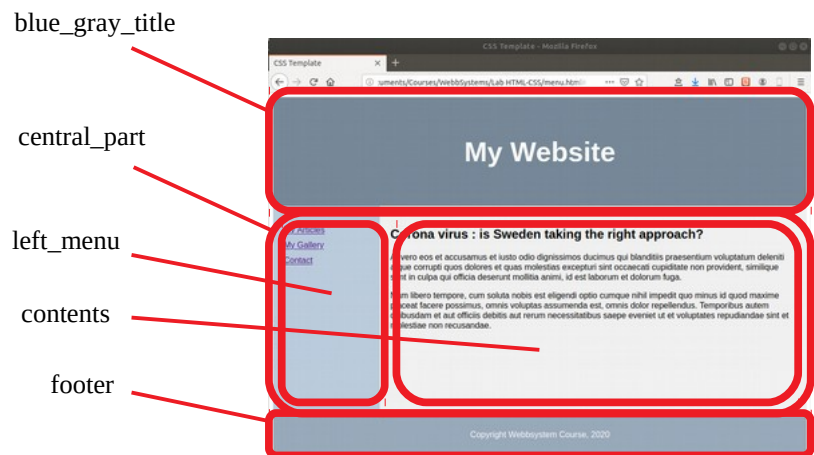
Open menu.html in your web browser. It should display a white page. Open menu.html in your text editor (html editor recommended).

Note 1: each time you modify your code, you need to **save** the file and **refresh** the browser to see the changes.

Note 2: Wrong HTML or CSS code rarely produce error messages: in case of syntax error, just nothing happens! So carefully check your spelling if you encounter a problem.

Divisions

The layout consists of 5 divisions, which we'll use to position the different elements. Each division will have a specific class attribute, and the classes will be defined in the .css file. The class names are indicated on the figure:



Title

In order to format the title, create a division in menu.html as follow:

```
<div class="blue_gray_title">  
    <h2>My Website</h2>  
</div>
```

Save and refresh the browser to see the result. Now we need to format it by defining a style for the corresponding class in the .css file:

```
.blue_gray_title {  
    color: red;  
}
```

Save and refresh the browser to see the result. Your title doesn't appear in red color? This is normal : you need to add a <link> tag in the head of your .html file to allow the browser to find the CSS.

→ Google "css link tag" to figure out how to do that.

Now, use the following CSS properties to make it look like on the picture:
background-color, padding, text-align, font-size, and color.

(You should not touch the HTML code, just modify the CSS.)

Central part

Under the title division, add the following code:

```
<div class="central_part">
  <div class="left_menu">

    <ul>
      <li>link1</li>
      <li>link2</li>
      <li>link3</li>
    </ul>

  </div>

  <div class="contents">
    <h2>Contents</h2>
    <p>bla bla</p>
  </div>
</div>
```

Note that the divisions `left_menu` and `contents` are inside the division `central_part`.

Replace `link1`, `link2`, `link3` by hyperlinks to `page1.html`, `page2.html`, and `page3.html`. In each of these pages, put a hyperlink “<< Back to main menu” which you can click to return to the menu.

Test it.

Formatting the menu

Add a class selector for `left-menu` in the `.css` file:

```
.left_menu {
}
```

Use the CSS properties `width`, `height`, `background`, and `padding` to get a good looking menu.

Formatting the contents

Add a class selector for `contents` in the `.css` file:

```
.contents {
}
```

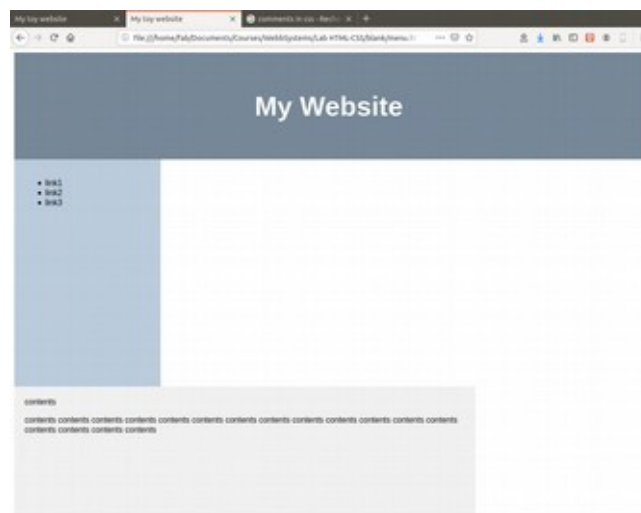
Add a paragraph of text in the contents (instead of “bla bla”) in order to see how it is positioned in the division.

Add a CSS property `padding` (and choose a value) so that the text is not touching the edge of the content box.

Also use the CSS properties `width`, `height`, and `background` to get a good looking contents.

Try to adjust the `width` of both the menu and the contents to get both divisions beside each other.

If you have used pixels (px) to specify the width, your menu and contents will not change size when you change the size of the window. Therefore you have to use a **relative** unit. Use for instance `width:20%` for the menu and `width:70%` for the contents. You should get something like this:



Check that the width of menu and contents changes when you change the size of the browser's window.

Now the problem is that the contents do not come beside the menu. This is normal: remember that `<div>` elements are **block** elements, so they bring a line return with themselves. In order to fix this problem, add a `float` property (and choose a value) to both `left-menu` and `contents` styles.

Now menu and contents should appear on the same line, but if you horizontally squeeze the window too much, the contents will jump below the menu again! You may also have problems to get the size of the menu + the size of the contents exactly equal to the size of the title.

Set the width values as follow:

`width:20%` for the menu and `width:80%` for the contents.

$20\% + 80\% = 100\%$, so it should be equal to the width of the page! Actually this doesn't work because these percentages are calculated on the size of content-box (see the box model), which is

smaller than the total box size (which includes padding and border). We cannot set the padding to 0 because we don't want the text to touch the borders. So instead, we'll change the way the percentages are calculated by adding the following style:

```
* {  
  box-sizing: border-box;  
}
```

Now menu and contents should be aligned and have together exactly the same size as the title. Check that this keeps working when you change the size of the browser window.

Footer

Add a division for the footer:

```
<div class="footer">  
  <p>Copyright Webbsystem Course, 2020</p>  
</div>
```

and the following style:

```
.footer {  
  background-color: #9ab;  
  padding: 30px;  
  text-align: center;  
  color: white;  
}
```

Did you notice the problem? Try to increase the padding, for instance 100px. The padding does not apply in the top part of the box. This is because the previous divisions are floating. We need to “clear” the floating behavior after. Use this special style to fix the problem:

```
.central_part:after {  
  content: "";  
  display: table;  
  clear: both;  
}
```

Check that the padding is correctly applied.

Check that the whole page keeps in shape as you resize the browser's window.

Mobile version

Now we want the page to display differently on a mobile phone. Squeeze the window horizontally to simulate the smaller screen of a mobile phone: the contents are squeezed and the menu items are occluded. Use the following CSS code for responsive styling:

```
@media (max-width: 600px) {
  .blue_gray_title {
    color: red;
  }
  .contents {
    color: green;
  }
}
```

Check what happens when you squeeze the browser's window horizontally. Change the code so that, instead of changing the color, your page gets reorganized as follow:



Tips:

- there are several CSS properties to change.
- for automatically adjusting the height of a box to the text: `height: auto;`

Last detail

Congratulations, your page is almost finished. One last detail: it would be nice to remove the bullet points in front of the items of the menu. You can do that by changing the `list-style-type` property as follow:

```
list-style-type: none;
```

But which selector should you use?