MYY106 - Introduction to Computer Science
8th Lab

Christmas mood

Before we start, let's make our desktop more... Christmas-like! The lab computers have the application `xsnow`. It is not in the `PATH`, so use the `whereis` command to see in which directory the `xsnow` executable is located and launch it with its absolute path.

Introduction

In this Lab we will create our own personal website. Our goal is to become familiar with the language `HTML`, which is used to develop web pages.

The personal web pages of the system users are located in the directory `~/public_html`. In order the web page to be accessible, the user's home directory must have `execute` permission for everyone. The `public_html` directory and its subdirectories must have `read` and `execute` permissions, while the files must have `read`.

Be careful: Do not give write permission to the contents of `public_html`, because this allows other users to modify them and makes your website vulnerable to attacks.

1. Create (if it does not exist) within your home directory the directory `public_html` that will host your website, and give the necessary permissions so that your website is accessible to everyone.

Implementation of the website

2. Inside the `public_html` directory, use the `wget` command to download the file `https://www.cse.uoi.gr/~myy106/files/index.html` (this is the file that the `web server` will read to display your website).

3. In a new tab in your browser, visit the address `https://www.cse.uoi.gr/~csyyxxxx` where `csyyxxxx` is your `username`.
   See how your website appears. This is the base on which you will form your content. If it doesn't load, check the permissions again.

Content configuration

Take a look at the code in the file `index.html`. You don't need to study it thoroughly. Just identify the `HTML` fields that you are familiar with from the lecture slides. Some pieces of code (such as lines 7 to 55 which contain css code) can be ignored for now. We will deal with these later.

4. Fill in the page title (line 5), your name (line 66) and email (line 70), so that your personal information appears.

In the past, we avoided displaying our email in its raw form on a website, so that automated bots could not collect it and use it to send spam or carry out phishing attacks. So to limit the unwanted messages, we used to write the email address in a descriptive form, such as `name [at] domain [dot] com`.

Today, however, Artificial Intelligence has evolved to such an extent that bots can easily recognize an email address even through images. Nevertheless, on many websites we continue to encounter emails presented in this descriptive form.

5. Find a small image on the internet that is representative of you (e.g. some kind of avatar, or a face, or if you want a real photo of yourself) and save it in the `public_html` directory. Modify the appropriate field in the `HTML` (line 63) to read your own image.

After each change, you should save the `.html` file and refresh the page (by pressing `F5` on your keyboard) to see the change in the browser. If the browser insists on showing you what it has stored in the `cache`, try `Ctrl+F5`.

Don't want to include your photo? No problem. It's quite common to use a standard clipart image, also known as an **avatar**. Try searching for images with the term `"user avatar"` and if you want add the term `"male"` or `"female"`.

Make sure the image you choose is free from copyright restrictions. When searching for images, under the search box, enable the **Tools** → **Licenses** → **Creative Commons Licenses** option to find images that you are allowed to use.

6. Your website displays various information, organized into the thematic sections `Courses`, `Projects`, `Skills` and `Interests`. In its current form, all of these sections are concentrated in the file `index.html`. We will change this structure: we will create separate files for each of the thematic sections, so that each file contains only the corresponding content.

   Click on this link and see the final result of the desired separation. https://youtu.be/8OTwe6FkDJ0.

   α′. First, study the format of the links in the file `index.html` (lines 79-82). These are not links to other pages, but internal links that lead to different points within the page. The fields `<a href="...">` contain the tags `[#mathimata, #ergasies, #deksiotites, #endiaferonta]`, while further down (lines 89, 111, 128 and 145) there are the corresponding fields `<a id="...">` containing the tags `[mathimata, ergasies, deksiotites, endiaferonta]`.

   Right now, if we click on the `Courses` link, we are taken to the part of the website where the label for `mathimata` is located. The same applies to our `ergasies`, `deksiotites` and `endiaferonta`.

   Also notice that the link we clicked on is included at the end of `url`.

   β′. Most websites we visit have a horizontal menu, the first item of which is usually "`Home`", which leads to the home page. To add the "`Home`" item to your own menu, simply insert an additional link before the others, which will display the word "Home" and will lead to the file `index.html` itself.

   γ′. In the `Home` link, add a small emoji-like icon before the word "Home", showing a house. You can find such icons, for example, at https://getemoji.com. Simply copy the emoji and paste into the `.html` file.

   δ′. Save the changes to the file `index.html`. Then create 4 more copies of the file in the same directory, and name them `courses.html`, `projects.html`, `skills.html`, and `interests.html`. You should have a total of 5 files that currently have exactly the same content.

   ε′. Open all the `.html` files, and remove from each of them the thematic sections that we do not want to appear in it. That is, when we load the file `courses.html`, we want only the thematic section `Courses` to appear below. In the file `projects.html`, only the thematic section `Projects` will appear. Similarly for `Skills` and `Interests`.

   στ′. In the file `index.html` a new thematic section named `Home` will appear below, which will contain a welcome text, e.g.

   Welcome to my web page!

   This site is currently under construction, but it will soon become a space where I share my journey as a student, including my courses, projects, skills, and interests.

   Check back soon for updates as I continue to learn, build, and grow!

   Make the text consist of separate paragraphs.

ζ΄. Update the links of all files, so that when we click on an item in the menu, the corresponding file is loaded. That is, the link `Courses` points to the file `courses.html` (the same for `Projects`, `Skills` and `Interests`).

Additional formatting

Now look at lines 7 through 55, which contain the `css` code for the page. From here we will change the appearance of the elements we see.

7. Change the colors of the page to match your preferences. See what colors are currently used by observing the values of the element properties (`color, background,` ...). These values are the `RGB` colors in hexadecimal format. You can also find various tools or suggestions for color combinations by visiting e.g. the website https://htmlcolorcodes.com . After choosing the colors you like, replace the `RGB` values in your own files.

8. Lines 22 to 47 define the style for the field `a`. Here we change the way the links are displayed.

   Uncomment lines 22 and 47 to enable the link styling. Refresh the page and notice how the links appear now. This will help you understand exactly what the `CSS` lines are changing. Take a moment to see exactly what effect each property has on the appearance of the page.

   The characters `"["` and `"]"` in lines 79-82 are no longer needed. Remove them from all files. Also, the appearance of all links has now changed, so the fields `<a id="...">` should be removed wherever they appear.

   

   Notice that in the link properties, we have chosen `rgb` colors with the same value for red, green and blue. These are the shades of gray.

   We chose a neutral shade to match whatever color scheme you have set for your page. Of course, if you want, you can also change the link colors.

9. Lines 49-53 define a new class named `largetext`. Activate it by removing the comments. In the code of each section there are the fields `<!--div class="largetext"-->` and `<!--/div-->`. Remove the comments to leave `<div class="largetext">` and `</div>`, and then refresh to see what is changed by `largetext`.

   Experiment with HTML and CSS properties to enhance the appearance of your page. For more information, you can search for examples or instructions on the World Wide Web. There are plenty of courses and tutorials. As ab example, we mention https://www.w3schools.com/

10. Format the content in the ”Courses”, ”Projects”, ”Skills” and ”Interests” sections so that the actual information you want to share is presented, creating some kind of a short CV of yourself.

    Add or remove sections and elements according to your needs, to bring your website into a form that is as representative of you as possible.
    Keep in mind that you are presenting yourself as a student of the Department. You do not need to include information about your personal life.

    Also keep in mind that your website is public and can be viewed by anyone. Share it with your people if you want (just give them the `url https://www.cse.uoi.gr/~csyyxxxx`).

Content Management Systems - CMS

When you are asked to create a modern website, you will not build it with plain `HTML` as we did here. You will use a CMS like e.g. **Wordpress** (https://wordpress.org), which provides a website management environment, and allows the average user to customize their content, using tools such as a text editor, a media manager, etc.

However, the current knowledge you have acquired about `HTML` and `css` will allow you to have greater control over the `CMS` that you will use in practice.

On your home computer you have administrator rights, so you can install any `CMS` without restrictions. In the Department's laboratories we cannot install `Wordpress`, because we do not have administrator rights.

However, there are other `CMS`, such as **Grav** (https://getgrav.org) or **CMSimple** (https://www.cmsimple.org), which do not require administrator rights, since they do not need access to a database. Such `CMS`, called `flat-file CMS`, can also be set up in your home directory in the labs, so that you have a fully functional page.

Submit answers: This time you didn't have to fill in anything in the answer file. Instead, you will submit something more impressive: your personal website. In the form field, paste the `url` of the page you created (`https://www.cse.uoi.gr/~csyyxxxx`).

https://forms.office.com/e/HNRXQJmv0n

Transferring Files from Home to the Lab

If you need to transfer files from your personal computer to the lab machines (for example, if you have implemented a `Python` program at home but you must use `turnin` from within the lab), you can do so through the terminal using the `scp` command:

```
scp file1 csyyxxxx@scylla.cse.uoi.gr:
```

This command copies the file `file1` to your home directory.
To copy it into a specific subdirectory, use:

```
scp file1 csyyxxxx@scylla.cse.uoi.gr:path/to/folder/
```

(replace `"path/to/folder/"` with the path to your own directory)