Web Development and Web Design

by R.Hasko.

Draft 01.2019

Lab 2. Classic **Web 1.0** or “retro” web site**.**

1. Create new directory **Project2** inside **web_dev** at your laptop. All tasks must be synhronized with **web_dev/Project2** at your **GitHub Pages**.
2. Create **index.html** with links *(menu bar with sub menu)* to:

home
form
- **GET**
- **POST**
- **file upload**

cookies
- **enabled/disabled**
- **create/retrieve**
- **modify/delete**

sessions
- **start**
- **get/modify**
- **destroy**

date/time
- **today is**
- **timezone**
- **timestamp**

database
- **(reserved)**

phpinfo

Next create all ***.html** pages according structure of menu. In the future Projects we will use both server side and frontend solutions with different approaches but for now…

You have only static html pages at your **GitHub Pages** hosting:

<your_login>.github.io/web_dev/Project2

Create one single separated file **menu.html** with **structure of menu** and include it inside all your static html pages **without any serverside scripting**. You can use **iframe** as below or other solutions

index.html:
*<iframe src="menu.html" width="90%" height="50"></iframe>
*
...main content…

menu.html:
home
page2
...
last_page

Questions:

1. **frameset** or **iframe**
2. why we need **target="_parent"**
3. other ways for include **a.html** into **b.html**

3. Start Your journey to world of web servers. Install set of back-end apps (for example, **Apache, PHP, MySQL**). You can use own web server from your OS, for example, **IIS/Windows** instead of **Apache,** with PHP too.

What Do I Need? To start using PHP, you can:

- Find a web host with PHP and MySQL support
- Install a web server on your own PC, and then install PHP and MySQL

Use a Web Host With PHP Support.

If your server has activated support for PHP you do not need to do anything.
Just create some **.php** files, place them in your web directory, and the server will automatically parse them for you.
You do not need to compile anything or install any extra tools.
Because PHP is free, most web hosts offer PHP support.

Set Up PHP on Your Own PC. You must:

- install a web server, for example **Apache**
- install **PHP**
- install a database, such as **MySQL**

The official PHP website (PHP.net) has installation instructions for PHP: http://php.net/manual/en/install.php. Look at XAMP, WAMP or OpenServer etc.

4. Start your local server and check using default web browser.
   (look at the http://localhost:80 or http://127.0.0.1)

   Questions:
   4. difference between **localhost** and **127.0.0.1**
   5. difference between **local** and **global IP**
   6. why we write **:80** after localhost
   7. difference between **file:///** and **http://**

5. Create subdirectory **Project2** inside root directory for web access
6. Create appropriate ***.php** pages according your menu system. For example, **form_GET.html** with **answer.php.**
   You can look at the detailed explanation with samples at:

   Form (GET, POST) https://www.w3schools.com/php/php_forms.asp
   File upload *https://www.w3schools.com/php/php_file_upload.asp*
   Cookies *https://www.w3schools.com/php/php_cookies.asp*
   Sessions *https://www.w3schools.com/php/php_sessions.asp*
   date/time *https://www.w3schools.com/php/php_date.asp*
   phpinfo *http://php.net/manual/en/function.phpinfo.php*

7. Add **README.md** into **Project2** with next text:

   Course “Web Development and Web Design”
   # Project 2
   Author: <your_first_name> <your_second_name>

   and check public accessibility from desktop and mobile browsers to **<your_login>.github.io/web_dev/Project2**

Web 1.0

Web 1.0 is the term used to refer to the first stage of development on the World Wide Web that was characterized by simple static websites.

The term Web 1.0 didn't appear until the term Web 2.0 was coined in 1999 by Darci DiNucci. During that time, the web was undergoing a major transformation. Most websites in the 1990s had originally been built with static HTML pages, and a few simple styles embedded in the HTML markup. In the late 1990s and early 2000s, interactive website features redefined what could be accomplished in a web browser and marked a major point of evolution in the world of web development.

During the stage known as Web 1.0, websites were marked by the following typical Web 1.0 characteristics:

- **Static pages:** Pages didn't offer interactive features that changed based on website visitor behavior. At that point websites were largely informational.
- **Website content stored in files:** Virtually every modern website makes use of a database to store the majority of website content. During Web 1.0 this was not the case and most website content was stored directly in the website files, not in a separate database.
- **Combination of content and layout:** Good web design practice today dictates the separation of webpage markup and styling. Virtually every modern website makes use of external style sheets to determine the look and layout of webpages. During Web 1.0 most styling was built into the page markup itself, often by misusing HTML elements such as tables.
- **Proprietary HTML tags:** During Web 1.0 browsers attempted to stand out by offering support for proprietary tags, creating significant incompatibility problems between websites that used these tags and site visitors using unsuported browsers.

- **Guestbooks:** Website visitor comments were usually added to a *Guestbook* page rather than attached directly to content pages.
- **E-mailing of forms:** Web hosting servers during the Web 1.0 phase rarely offered support for server-side scripting, which is required to use the web server to submit a form. As a result, during Web 1.0, when the *Submit* button was clicked on most forms the website visitor's e-mail client would launch, and the visitor would have to e-mail their form to an e-mail address provided by the website.

The transition from Web 1.0 to 2.0 took place over time as servers were upgraded, average connection speeds increased, and developers learned new skills and techniques. The transition began in the last year or two of the 1990s and Web 2.0 features had made strong headway by 2006, although there are still vestiges of Web 1.0 with us today in quiet corners of the web.

Server side or Back-End (PHP)

PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.

Example.php

```
<!DOCTYPE html>
<html>
<body>
<?php
echo "My first PHP script!";
?>
</body>
</html>
```

Versions of PHP. You can use PHP 5 or PHP 7.
PHP 7 is the newest version of PHP. PHP 7 is much faster than the previous popular stable release! PHP 7 has improved Error Handling, supports stricter Type Declarations and supports new operators, methods and properties.

What is PHP?

- PHP is an acronym for "PHP: Hypertext Preprocessor"
- PHP is a widely-used, open source scripting language
- PHP scripts are executed on the server
- PHP is free to download and use

PHP is an amazing and popular language!

It is powerful enough to be at the core of the biggest blogging system on the web (**WordPress**)! It is deep enough to run the largest social network (**Facebook**)!
It is also easy enough to be a beginner's first server side language!

What is a PHP File?

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code are executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension ".php"

What Can PHP Do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can be used to control user-access
- PHP can encrypt data

With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

Why PHP?

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is free. Download it from the official PHP resource: www.php.net
- PHP is easy to learn and runs efficiently on the server side

Read more:

PHP 5 Tutorial https://www.w3schools.com/php/default.asp
PHP 7 Tutorial https://www.w3schools.com/php7/default.asp
The smart PHP IDE from Jetbrains https://www.jetbrains.com/phpstorm/
Book https://education.fsu.edu/wp-content/uploads/2015/04/Learning-PHP-MySQL-JavaScript-and-CSS-2nd-Edition-1.pdf