

CSCI E-12 Assignment 4

150 points.

Due by Monday, May 5, 12 noon EDT

Submit via the "Assignments" area of the course site.

<http://cscie12.dce.harvard.edu/assignments>

Goals:

1. **Web Server Configuration.** Understand HTTP settings and use Apache configuration directives to customize HTTP responses
2. **Building a Page from Parts Using PHP include.** Build a small site for the U.S. Constitution using PHP "include"
3. **Video.** Include video content in a markup page, delivered by both HTTP download and RTMP streaming. Details of this part of the assignment will be given after the lecture on 4/21.

It is expected that all markup you create will validate (no errors; warnings OK) [note that for the PHP part of the assignment, the individual PHP documents will not validate, but the final markup delivered by the webserver should]. CSS that you write should also validate with no errors (warnings are OK). Validation will be checked with the W3C Markup and CSS validators.

Getting Started with this Assignment

- You will submit your work via a ZIP file to the dropbox, as well as publish your work on the course web hosting server (`morpheus.dce.harvard.edu`).
- **Working on the course web hosting server.** Use your obscurely named directory for this assignment as well. If you haven't already, please create an obscurely named folder in your `public_html` directory that will contain your work for the course. The reason for the obscure name is to prevent intentional or unintentional browsing via the web of your hard work by others. For example, the user John Harvard might create a folder named `VeRiTas_1636`.
- Download the assignment 4 "zip" file from the course web site and unzip it.
- For the "Apache Configuration" part of this assignment, it will be critical to work on the course web hosting server, `morpheus.dce.harvard.edu`.
- Keep the file names and directory structures the same. We rely on the file names being consistent when we grade.
- Edit "answers.html" to provide the requested information.
- When you are done with the assignment, zip the contents of the assignment folder and submit the zip file via the "Assignments" area on the course site.

Part 1. HTTP and Apache Configuration

40 points. To test your work in this part of the assignment, you will need to use the course web server, `morpheus.dce.harvard.edu`. You will create an `.htaccess` file to be placed in your `"public_html"` directory on `morpheus.dce.harvard.edu`. Through the configuration directives in the `.htaccess` file, you will accomplish the following:

1. Configure a custom error document response for HTTP response status of 404 ("Not Found").
2. Set expiration headers so that CSS, JavaScript, and image (JPEG, GIF, PNG) files are cached for 1 day.
3. Set an output filter so that XHTML, HTML, CSS, and JavaScript files are compressed before they are sent to the web browser.
4. Create a redirect or rewrite rule so that the URL (where your username is substituted in place of *jharvard*):

`http://morpheus.dce.harvard.edu/~jharvard/myfavoritebook`

redirects (HTTP 301 or 302) to a page that contains information about a favorite book of yours. The page that is redirected to should be an external URL (e.g. from WorldCat, Amazon, B&N, publisher site, etc.).

Use "Redirect" unless you want to take on the more difficult (but more flexible) "Rewrite" configuration.

You must also include your `.htaccess` file as part of your "zip" file submission (place in the `"apache_htaccess"` folder).

Part 2. Building a Page from Parts Using PHP Include

60 points. Build a small site for the U.S. Constitution using PHP “include”. This part of the assignment relies on the functionality provided by a web server configured to run PHP. You should use `morpheus.dce.harvard.edu` for this part of the assignment.

The goal is to take an existing site that does not use PHP include and re-construct it so that the pages are created through PHP include by combining different files. PHP include will be used so that each common page component can be separate out into its own file.

The starting files are in `usconstitution/original/`. Your files for this part should be in `usconstitution/php/`

An XHTML version of the United States Constitution is provided (8 XHTML documents). The pages use Blueprint CSS for page layout. In this exercise, you will re-construct this site using PHP so that the pages share a common “header”, “navigation”, and “footer” content. So in the end, you will create $8 + 3 = 11$ documents (as well as copying over the “css” and “images” folders):

```
usconstitution/php
|- index.php
|- ii.php
|- iii.php
|- iv.php
|- v.php
|- vi.php
|- vii.php
|- amendments.php
`- includes
    |- footer.php
    |- header.php
    `- navigation.php
```

You should keep the documents well-formed (i.e. start tags and corresponding end tags are in the same document, not split across different documents). Check that the final document delivered by the web server validates.

In the “footer”, use the **echo** command of PHP to produce the URI of the document (server name and path) and the last modified date.

The navigation for each page should show the current page differently than the others (as in the “original” site provided). You may use either of the two techniques to be discussed in class for this – a solely CSS-based technique, or a combination CSS and JavaScript technique for this.



United States Constitution - Amendments

Article I
Article II
Article III
Article IV
Article V
Article VI
Article VII
Amendments

Proposed by Congress, and ratified by the legislatures of the several states, pursuant to the Fifth Article of the original Constitution.

Article I

Congress shall make no law respecting an establishment of religion, or prohibiting the free exercise thereof; or abridging the freedom of speech, or of the press; or the right of the people peaceably to assemble, and to petition the Government for a redress of grievances.

Article II

A well regulated Militia, being necessary to the security of a free State, the right of the people to keep and bear Arms, shall not be infringed.

Article III

No Soldier shall, in time of peace be quartered in any house, without the consent of the Owner, nor in time of war, but in a manner to be prescribed by law.

Article IV

The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated, and no Warrants shall issue, but upon probable cause, supported by Oath or affirmation, and particularly describing the place to be searched, and the persons or things to be seized.

Article V

No person shall be held to answer for a capital, or otherwise infamous crime, unless on a presentment or indictment of a Grand Jury, except in cases arising in the land or naval forces, or in the Militia, when in actual service in time of War or public danger; nor shall any person be subject for the same offence to be twice put in jeopardy of life or limb; nor shall be compelled in any criminal case to be a witness against himself, nor be

Part 3. Video

50 points. Author two markup documents, each of which includes a video. One video is delivered via HTTP download, and the other via RTMP and the Amazon Cloudfront CDN.

- Produce two markup pages, each containing the “Publishing a page on the course web server” video – one will use the HTTP download method, the other will use RTMP streaming. The markup pages can be very “plain” – the main point here is to show you can include video in a page using Javascript and a video player.
 - Files:
 - `video/video_http.html`
 - `video/video_rtmp.html`
- Use **either** JW Player or Flowplayer (both are widely used HTML5+Flash-based video players). You get to choose which one – you do not need to use both. I recommend using the JW Player, as it is easier to configure; however, you are welcome to use the open source Flowplayer if you wish. If there is another player you wish to use (see <http://praegnanz.de/html5video/>), you are free to do so.
- Use Javascript to include the video in the page (i.e. do not use “embed” or “object” tags). You should use the Javascript libraries that come with the video player you are using.

Video information

- Title: “Publishing a page on the course web server”
- Dimensions: height: 600px; width: 800px
- HTTP download URL:
https://s3.amazonaws.com/csciel2/screencasts/publishing_on_web_server.mp4
- Streaming information:
 - Application path:
`rtmp://s35uyvud967x16.cloudfront.net/cfx/st/`
 - Stream identifier:
`mp4:screencasts/publishing_on_web_server.mp4`
- For your interest only (nothing “actionable” here): I am using Amazon S3 to deliver the HTTP download video and Amazon Cloudfront to deliver the RTMP streaming video. See: <http://www.learningapi.com/streamingmedia-articles/amazon-cloudfront-streaming/> for details. Amazon S3 and Cloudfront are an easy and inexpensive way to host files to download and stream videos.

Notes for JW Player

- Home page: <http://www.longtailvideo.com/jw-player/>
- Download page: <http://www.longtailvideo.com/jw-player/download/>
You should use the free version using the
“Free Download JW Player for Non-Commercial Use” button
- Documentation: <http://www.longtailvideo.com/support/jw-player/>
- RTMP streaming information:
<http://www.longtailvideo.com/support/jw-player/28854/using-rtmp-streaming>
- For RTMP streaming, the application path is combined with the stream identifier:
`rtmp://s35uyvud967x16.cloudfront.net/cfx/st/mp4:screencasts/publishing_on_web_server.mp4`

Notes for Flowplayer 5.4.6

- Home page: <http://flowplayer.org/>
- Use “Flowplayer HTML5”, not “Flowplayer Flash”
- You can either download flowplayer, or use versions on the flowplayer CDN
 - Download page: <http://flowplayer.org/download/>
ZIP: <http://releases.flowplayer.org/5.4.6/flowplayer-5.4.6.zip>
 - CDN versions: See “2. <head/> setup” section at
<https://flowplayer.org/docs/setup.html#head-setup>
- Documentation: <http://flowplayer.org/docs/>
- RTMP streaming information:
<https://flowplayer.org/docs/setup.html#server-side>
- Note that for RTMP streaming, there is a `data-rtmp` attribute of the containing `div`, whose value is the “Application path” of the video. There is also a `src` attribute for the `source` element within the `video` element, whose value is the “Stream identifier” of the video:
 - `data-rtmp="rtmp://s35uyvud967x16.cloudfront.net/cfx/st/"`
 - `<source src="mp4:screencasts/publishing_on_web_server.mp4" ... />`

