

# cis20.1-fall2007-sklar, assignment III

## instructions

- This is the assignment for unit III. It is worth 14 points.
- **It is due on Monday November 26.**
- You don't need to email me anything! I will assume that you have followed the naming instructions below, so I will look on your class server account for your **index.html** which should contain links as described in the last section (below).

## 1 setting up php

1. Log into the new class web server using **telnet** and the username/password that I'll give you in the lab. The IP address of the web server is: 146.245.250.181
2. Create a directory called `public_html`:  
unix-prompt\$ `mkdir public_html`
3. Set its permissions to be owner everything, group and world readable and executable:  
unix-prompt\$ `chmod 755 public_html`
4. Create a simple test HTML file called `index.html` in your `public_html` directory, e.g., something like this:

```
<html>
<body>
hello and welcome to prof sklar's cis20.1 page!
</body>
</html>
```

The editors **emacs** and **vi** are both installed on the class web server. You can also edit files on your workstation or laptop and then ftp them to the web server. A "quick and dirty" emacs handout will be available in class.

5. Make sure that your simple test HTML file works by opening a browser window and going to:  
`http://146.245.250.181/~your-username/index.html`  
**If this part doesn't work, ask me for help! You CAN'T get any further with the lab if this part does not succeed!**  
(1 point)

6. Create a simple test PHP file called `p1.php` in your `public_html` directory, e.g.:

```
<?php print "hello\n"; phpinfo(); ?>
```

7. Make sure that your simple test HTML file works by opening a browser window and going to:  
`http://146.245.250.181/~your-username/p1.php`  
**If this part doesn't work, ask me for help! You CAN'T get any further with the lab if this part does not succeed!**
8. Add a link in your `index.html` file labeled "hello" that invokes `p1.php` when it is clicked.  
(1 point)

Note that your files must have an extension of `.php` in order for the browser to recognize that there are PHP commands inside it that need to be interpreted.

## 2 exploring php

Note that all the PHP examples from class have been posted on the syllabus portion of the web page, under unit III (<http://www.sci.brooklyn.cuny.edu/~sklar/cis20.1>). They are all now linked to my page on the new class server, and you can try them out from the web page. You can also download the source code and try them out yourself on your page on the class server.

1. Copy the example files `bigform.html` and `bigform.php` from the class web page onto your class server account:

- html form is here: <http://146.245.250.181/~sklar/examples/bigform.html>
- php source code is here: <http://146.245.250.181/~sklar/examples/bigform.txt>

Note that the source code file is named `.txt` instead of `.php` — that is so you can look at the source code without it being executed by the browser. There is also a file there called `bigform.php`, which is what is invoked when you click on the **submit** button on the HTML form page; `bigform.txt` is just a copy of `bigform.php` with the extension changed so you can look at the code.

2. Make sure they work by entering data in every field on the form and then clicking **submit**. Verify that all your fields are echoed correctly.
3. Add a link in your `index.html` file labeled “bigform” that goes to your `bigform.html` when it is clicked. (1 point)
4. Now copy the example files `myface.html` and `myface.php` from the class web page onto your class server account:

- html form is here: <http://146.245.250.181/~sklar/examples/myface.html>
- php source code is here: <http://146.245.250.181/~sklar/examples/myface.txt>

5. In your `public_html` directory, create a subdirectory called `users` and set its file permissions to everything/everybody:

```
unix-prompt$ mkdir users
unix-prompt$ chmod 777 users
```

Note that this isn't necessarily a great idea in terms of security, but really the only bad thing that can happen is that people might post inappropriate content. In which case, I'll delete it.

6. Make sure they work by entering data in the **name** field on the form and then clicking **submit**. Verify that it works correctly by clicking on the link displayed. A “hello” message should appear with your name on it.
7. Add a link in your `index.html` file labeled “myface” that goes to your `myface.html` when it is clicked. (1 point)

### 3 creating your myface :-) web site

Okay, so up to here, you've earned 4 points by basically following what I've given you and making sure you have your account set up correctly. The rest of the assignment (*10 more points*) is where you really do something of your own and put all the pieces together. The idea here is: **first**, create a form (like the **bigform** example, above) in which you ask a user a bunch of questions to get information about them; and then **second**, create a web page for the user based on the information you've collected in the form (like the **myface** example, above).

**Be creative!** *This is your big opportunity to demonstrate what you have learned about PHP and interface design!*

1. Modify `myface.html` to collect information from a user that you can transform into something interesting to display.

*(5 points this part)*

You must use at least 5 different types of input widgets on the HTML form, not including the submit button; i.e., select at least 5 from:

```
<input type="checkbox" ... >
<input type="file" ... >
<input type="password" ... >
<input type="radio" ... >
<input type="text" ... >
<select ... ><option> ... </select>
<select ... multiple><option> ... </select>
<textarea ... >
```

You will earn *1 point* for each type of widget you use and handle correctly, which includes passing data to the PHP file in the next step.

2. Modify `myface.php` to generate a web page—the user's "myface" page—using the information gathered from the form.

*(5 points this part)*

You will earn *1 point* for each piece of input you read from the form correctly and display creatively.

*If you display everything simply as plain text, without changing any properties (i.e., by using default colors and fonts), then you will not earn any points here. :-)*

If you need help with HTML, there are a number of links on the class web page under on-line resources for unit III that point to information and/or tutorials on basic and advanced HTML, style sheets, web safe colors, and HTML forms.

This doesn't need to be complicated. You could, for example:

- (a) use `<input type="text" ... >` to get the user's name and then display their name in a non-default color and non-default font
- (b) use `<select ... multiple>` to get a list of the user's favorite baseball teams and then display them using an ordered (`<ol>`) or unordered (`<ul>`) list
- (c) display a few pictures on your form and use `<input type="radio" ... >` to ask the user to indicate which image most closely resembles themselves; then display the selected image on their "myface" page
- (d) use `<textarea ... >` to get the user to enter directions for how they get to campus; then on the "myface" page, substitute icons from <http://www.mta.info> for the name of each subway line the user enters

I won't give you all the ideas...

Have fun and BE CREATIVE!