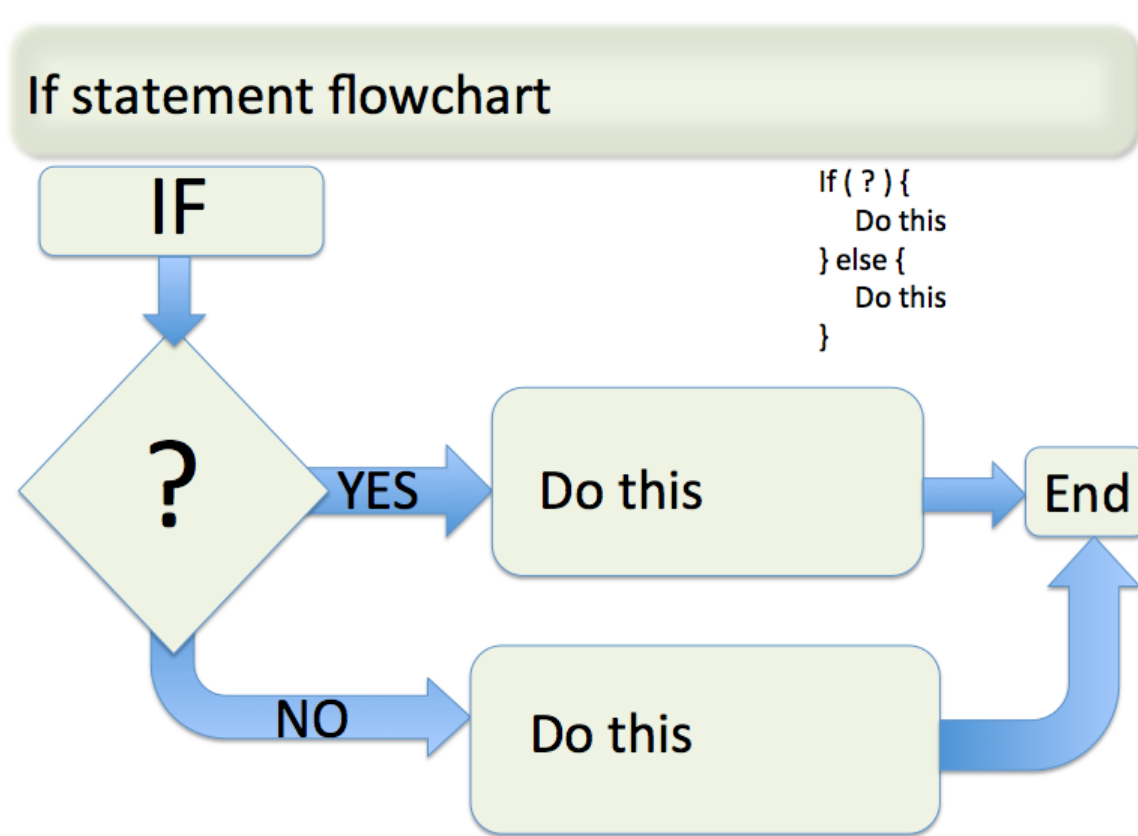


Chapter 2 Selection

Previously we learned about the **Sequence structure** where computers execute line by line. Then we learned about the **Repetition structure** where computers keep executing the same lines of code over and over again. Now let us learn about the **Selection structure** which allows us to choose the instructions we want the computer to do.

The Selection structure is something you do many times on a daily basis and you can think of it as a question with a yes or no answer (as humans we have lots of maybes). For example when you leave the house you ask yourself a question "Is it raining?" If the answer to that question is yes you wear a rain coat. If the answer is no you don't wear a raincoat. When we customize our websites we ask lots of questions.

Let us have our website display Good Morning or Good Afternoon. We can do this with the **if statement**. Let's look at the flowchart for an if statement.



In order to do this we need to know what time it is [<http://php.net/manual/en/function.date.php>]. We can do this with the PHP method date which gets us the server's current date and time (a big number counting how many seconds since 1/1/1970) formatted to how we need it. In our case we want the hour in 24 hour format which uses a capital G. To initialize a variable to the current hour it would look like this:

```
$now = date("G");
```

Our question is: Is \$now less than 12?

```
if ($now < 12) {
```

Our question always has to be expressed as a logical expression that has a true or false (yes or no) result. There are several operators [<http://php.net/manual/en/language.operators.comparison.php>] that we can use like < (less than) or > (greater than). When we want to see if something is equal to we use two equal signs ==. The biggest mistake people make is they use just one (it sounds normal \$x = 5 but that assigns 5 to the variable x whereas \$x == 5 asks the question is \$x equal to 5). Here is our if statement indented properly:

```
if ($now < 12) {  
    print '<h2>Good Morning</h2>';  
} else {  
    print '<h2>Good Afternoon</h2>';  
}
```

putting it all together (also indented properly) and we get this (try it and see which greeting you get):

```
<!DOCTYPE html>  
<html>  
    <head>  
        <title>Try Me</title>  
        <meta charset="utf-8">  
        <meta name="description" content="just a testing file">  
    </head>  
    <body>  
<?php  
$now = date("G");  
if ($now < 12) {  
    print '<h2>Good Morning</h2>';  
} else {  
    print '<h2>Good Afternoon</h2>';  
}  
?>  
    </body>  
</html>
```

Of course this means that after 6 pm (hour 18) it still says good afternoon. We can extend our if statement by making an **elseif statement** like this:

```
if ( $now < 12 ) {
    print ' <h2>Good Morning</h2>';
} elseif ( $now < 18 ) {
    print ' <h2>Good Afternoon</h2>';
} else {
    print ' <h2>Good Evening</h2>';
}
```

Can you see how that works? Try adding another elseif so you can add Good Night.

Another use is to have PHP print a class for the current page you are on. For example, if we are on the about page we would like our HTML navigation to look like this with a class attribute printed for the about page so that we can style the CurrentPage menu item differently (good for usability in letting the user know which page they are currently on). This is what we want:

```
<nav>
  <ol>
    <li>
      <a href="index.php">Home</a>
    </li>
    <li class="CurrentPage">
      <a href="about-us.php">About Us</a>
    </li>
  </ol>
</nav>
```

Our first step is to look at what we want PHP to customize which is `class="CurrentPage">`. In order to do this we need PHP to tell us what page we are on. There is built-in PHP **Global variable** (meaning variables available to all pages) that has the path and filename in it. For example, the URL:

`http://rerickso.w3.uvm.edu/cs008/lab1.php`

We can pull out the /cs008/lab1.php with the PHP global variable:

```
$_SERVER[ 'PHP_SELF' ]
```

We always want to sanitize (clean, remove bad things) the data (information) that comes from the user. Everything in the `$_SERVER` variable always needs to be cleaned. We do this

with another PHP function called `htmlspecialchars`. This function will replace certain characters that have meaning in HTML to less dangerous characters. For example, it will take a `<` symbol (used at the beginning of every HTML tag) and replace it with `<`; which means it will no longer be interpreted as an HTML opening tag. Here is how we assign this to a variable:

```
$phpSelf = htmlspecialchars($_SERVER['PHP_SELF'], ENT_QUOTES, "UTF-8");
```

Now in our example `$phpSelf` would have `/cs008/lab1.php` and we only want `lab1`. There is another PHP method we can use called `pathinfo` which puts `$phpSelf` into an array, breaking it up into pieces. Let's type into `tryme.php` and see what it looks like. I am going to use `print_r` to print out the `pathinfo` array.

```
<?php
$phpSelf = htmlspecialchars($_SERVER['PHP_SELF'], ENT_QUOTES, 'UTF-8');

$pathParts = pathinfo($phpSelf);

print '<pre>';
print_r($pathParts);
print '</pre>';
?>
```

This will print out the array. This array is not an indexed array but an associative array which means it has names to index the values and not numbers.

```
Array
(
    [dirname] => /cs008
    [basename] => lab1.php
    [extension] => php
    [filename] => lab1
)
```

We can see that to get `lab1` we need to ask for the `filename`. To assign the `filename` to a variable it would look like this (I print it out just so I can see it, after I see it I will delete the print line):

```
$currentPage = $pathParts['filename'];
print "<p>Current Page is: " . $currentPage;
```

Ok that was a lot of work just to make one small customization. Yes but that is the way programming is. Our goal is to have PHP print out `class="CurrentPage"` in our nav. Looking at our HTML:

```
<li class="CurrentPage">
```

We add PHP to the place we want to replace:

```
<li <?php print 'class="CurrentPage"'; ?> >
```

This just prints it out but is our first step as we having PHP print out the class. Test this to make sure it works and then we add our if statement.

```
<li  
<?php  
if ($currentPage == 'about-us') {  
    print 'class="CurrentPage"';  
}  
?>  
>
```

There is no else statement in this case as we do nothing if they are not equal. We do need to do this for each nav item so my little sample will look like this:

```
<?php  
$phpSelf = htmlentities($_SERVER['PHP_SELF'], ENT_QUOTES, 'UTF-8');  
$path_parts = pathinfo($phpSelf);  
  
$currentPage = $pathParts['filename'];  
?>
```

```
<!-- DOCTYPE, html, head body etc all go here. Removed for simplicity -->
```

```
<nav>  
    <ol>  
        <li <?php  
            if ($currentPage == 'index') {  
                print 'class="CurrentPage"';  
            }  
            ?>  
            >  
            <a href="index.php">Home</a>  
        </li>  
        <li <?php  
            if ($currentPage == 'about-us') {  
                print 'class="CurrentPage"';  
            }  
            ?>  
            >  
            <a href="about-us.php">About Us</a>  
        </li>  
    </ol>  
</nav>
```

That is a lot for a little customization! I am not expecting you to be able to do this without having my sample as this is PHP fast track, which means I will be giving you templates you need to follow. Generally you need to edit a few parts of the code like, what is the name of the file. The rest is just what it is as that is how you do it. Are you starting to see how we use PHP to customize our web site by printing out custom HTML code for us?

Keep in mind my nav sample does nothing for the look unless you add rules to your CSS file! Lets say you have no background color for your nav. For the current page we could just add a background color (there are lots of things you can do but that is for you to play around with).

```
.CurrentPage{
    background-color: #D9D9D9;
}
```

Here is my sample which makes it easy to see which page you are on:



Here are the other rules I used:

```
nav ol li{
    display: inline-block;
    list-style-type: none;
    padding: 0.5em;
    text-align: center;
    width: 5em;
}

nav li:hover{
    background-color: lightcyan;
}
```

Self Test Questions
topics:

- if statements
- phpSelf
- htmlentities

1. Why is this not a good computer conditional question?

Are you single or married?

2. What is incorrect about this if statement?

```
if($numberOfClasses = 5){  
    print "You have a normal course load";  
}
```

3. Why will this not say "good night" at 1 am?

```
if($currentHour < 12){  
    print "Good Morning";  
} elseif ($currentHour < 18){  
    print "Good Afternoon";  
} elseif ($currentHour < 24){  
    print "Good Evening";  
} elseif ($currentHour < 2){  
    print "Good Night";  
} else{  
    print "I'm Sleeping";  
}
```

Challenge Questions

1. Age Classifier

Write some php code that determines whether a person is an infant, a child, a teenager or an adult depending on the 'age' variable you set. Print their classification on screen. Use the following guidelines:

- If the person is 1 year old or less, he or she is an infant
- If the person is older than 1 year, but younger than 13 years, he or she is a child.
- If the person is at least 13 years old, but less than 20 years old, he or she is a teenager.
- If the person is at least 20 years old, he or she is an adult.

2. Mass and Weight

Scientists measure an object's mass in kilograms and its weight in newtons. If you know the amount of mass of an object in kilograms, you can calculate its weight in newtons with the following formula:

$$\text{weight} = \text{mass} \times 9.8$$

Write some php code that calculates weight based off mass stored in a variable. If the object weighs more than 500 newtons, display a message indicating that it is too heavy. If the object weighs less than 100 newtons, display a message indicating that it is too light.

3. Magic Dates

The date June 10, 1960 is special because when it is written in the following format, the month times the day equals the year:

6/10/60

Write some php code that determines whether the month times the day equals the year when given variables that hold a month, a day, and a two digit year. If so, the page should display a message saying the date is magic. Otherwise, it should display a message saying the date is not magic.

4. Color Mixer

The colors red, blue and yellow are known as the primary colors because they cannot be made by mixing other colors. When you mix two primary colors, you get a secondary color, as shown here:

When you mix red and blue, you get purple.

When you mix red and yellow, you get orange.

When you mix blue and yellow, you get green.

Write some php code that determines the mixed color based on two variables and displays it on the page.

5. Body Mass Index

Write some php code that calculates and displays a person's body mass index. The BMI is often used to determine whether a person is overweight or underweight for his or her height. A person's BMI is calculated with the following formula:

$$BMI = weight \times 703 / height^2$$

where *weight* is measured in pounds and *height* is measured in inches. Store height and weight as variables. The page should display the BMI and a message indicating whether the person has optimal weight, is underweight, or is overweight. A person's weight is considered to be optimal if his or her BMI is between 18.5 and 25. If the BMI is less than 18.5, the person is considered to be underweight. If the BMI value is greater than 25, the person is considered to be overweight.

Answers to Challenge Questions

1. Answer:

```
<?php
$age = 18;

if($age <= 1){ print "Infant";}
else if($age >1 and $age <13){ print "Child";}
else if($age >13 and $age < 20){ print "Teenager";}
else if($age > 20){ print "Adult"; }
else { print "Undefined age!";}

?>
```

2. Answer:

```
<?php
$mass = 10;

$weight = mass * 9.8;
if($weight > 500){ print "Object too heavy!";}
if($weight < 100){ print "Object too light!";}

?>
```

3. Answer:

```
<?php
$day = 6;
```

```

$month = 10;
$year = 60;

if($day * $month == $year){
    print "The date is magic!";
} else {
    print "The date is not magic.";
}

?>

```

4. Answer:

```

<?php
    $color1 = "red";
    $color2 = "blue";

    if(($color1 == "red" and $color2 == "blue")
        or ($color1 == "blue" and $color2 == "red")){
        print "purple";
    }
    if(($color1 == "red" and $color2 == "yellow")
        or ($color1 == "yellow" and $color2 == "red")){
        print "orange";
    }
    if(($color1 == "yellow" and $color2 == "blue")
        or ($color1 == "blue" and $color2 == "yellow")){
        print "green";
    }

?>

```

5. Answer:

```

<?php
    $weight = 150;
    $height = 72;

    $bmi = $weight * 703 / ($height * $height);
    print $bmi;
    if($bmi < 18.5) { print "Underweight"; }
    else if($bmi >=18.5 and $bmi <25){ print "Optimal"; }
    else if ($bmi > 25){ print "Overweight"; }

?>

```