

PHP

Controlling the actions

Last week hw disclosure

- Get it @ <http://istudy.tk/trip-planner/index.php.txt>
- For someone who didn't send hw
 - Grab this disclosure above, save it into PHP file in your www folder insides wamp installation folder.
 - Change background of body, run it by your web browser.
 - See the result and try to understand HTML tags and PHP code.
 - Add explanation by commenting on every line of PHP command - explain what that command is going to do
 - Save file, send me to codeenjoy@gmail.com you will get 1 point of bonus.

Condition statement

Condition statement

- Very often when you write code, you want to perform different actions for different decisions. You can use conditional statements in your code to do this.

In PHP we have the following conditional statements:

- **if** statement - use this statement to execute some code only if a specified condition is true
- **if...else** statement - use this statement to execute some code if a condition is true and another code if the condition is false
- **if...elseif....else** statement - use this statement to select one of several blocks of code to be executed
- **switch** statement - use this statement to select one of many blocks of code to be executed

if statement - use this statement to execute some code only if a specified condition is true

See format below:

```
if (condition)
    code to be executed if condition is true;
```

example 1

```
<html>
<body>

<?php
    $j = 5;
    $k = 10;
    if ($j < $k){
        print "Variable $j value is less than $k";
    }
?>

</body>
</html>
```

example 2

```
<html>
<body>
<?php
    // Get name of today as three characters
    $d=date("D");
    if($d=="Thu") print "Enjoy PHP<br />";
    if($d=="Fri") print "Have a nice weekend!";
    if($d=="Thu") { print "Cannot copy haha"; }
?>
</body>
</html>
```

if...else statement - use this statement to execute some code if a condition is true and another code if the condition is false

See format below:

```
if (condition)
    code to be executed if condition is true;
else
    code to be executed if condition is false;
```

example 1

```
<html>
<body>
<?php
    $j = 10;
    $k = 5;
    if($j < $k) {
        // This line will not be executed
        print "Variable $j value is less than $k";
    }
    else {
        // This line will be executed
        print "Variable $j value is greater than $k";
    }
?>
</body>
</html>
```

example 2

```
<html>
<body>
<?php
    $d=date("D");
    if ($d=="Fri")
        print "Have a nice weekend!";
    else
        print "Have a nice day!";
?>
</body>
</html>
```

If more than one line should be executed if a condition is true/false, the lines should be enclosed within curly braces:

```
<html>
<body>
<?php
    $d=date("D");
    if ($d=="Thu")
    {
        print "Hello!<br />";
        print "Enjoy PHP! ";
        print "See you next week";
    }
?>
</body>
</html>
```

if...elseif....else statement - use this statement to select one of several blocks of code to be executed

See format below:

```
if (condition)
    code to be executed if condition is true;
elseif (condition)
    code to be executed if condition is true;
else
    code to be executed if condition is false;
```

example 1

```
<html>
<body>
<?php
$d=date("D");
if ($d=="Fri")
    print "Have a nice weekend!";
elseif ($d=="Thu")
    print "Have a nice PHP day!";
else
    print "Have a nice day!";
?>
</body>
</html>
```

switch statement - use this statement to select one of many blocks of code to be executed

See format below:

```
switch(n)
{
    case label1:
        code to be executed if n=label1;
        break;
    case label2:
        code to be executed if n=label2;
        break;
    default:
        code to be executed if n is different from both label1 and label2;
}
```

example 1

```
<html>
<body>
<?php
$destination = "Tokyo";
print "Traveling to $destination<br />";
switch ($destination) {
    case "Egypt":
        print "Bring 15 bottles of SPF 50 Sunscreen<br />";
        break;
    case "Tokyo":
        print "Bring lots of money<br />";
        break;
    case "Amsterdam":
        print "Bring an open mind<br />";
        break;
}
?>
</body>
</html>
```

example 2

```
<html>
<body>
<?php
$destination = "Tokyo";
print "Traveling to $destination<br />";
switch ($destination) {
    case "Egypt":
        print "Bring 15 bottles of SPF 50 Sunscreen<br />";
        break;
    case "Tokyo":
        print "Bring lots of money<br />";
        //break; comment this out, what will happen?
    case "Amsterdam":
        print "Bring an open mind<br />";
        break;
}
?>
</body>
</html>
```

example 3

```
<html>
<body>
<?php
$destination = "Egypt";
print "Traveling to $destination<br />";
switch ($destination) {
    case "Egypt":
        print "Bring 15 bottles of SPF 50 Sunscreen<br />";
        break;
    case "Amsterdam":
        print "Bring an open mind<br />";
        break;
    default:
        echo "Bring lots of underwear!<br />";
        break;
}
?>
</body>
</html>
```

switch is similar to **if** but **switch** is speedy.

- With the use of the switch statement you can check for all these conditions at once, and the great thing is that it is actually more efficient programming to do this.
- The way the Switch statement works is it takes a single variable as input and then checks it against all the different cases you set up for that switch statement. Instead of having to check that variable one at a time, as it goes through a bunch of If Statements, the Switch statement only has to check one time.

looping

[Often when you write code, you want the same block of code to run over and over again in a row. Instead of adding several almost equal lines in a script we can use loops to perform a task like this.]

while loop statement

- loops through a block of code while a specified condition is true
- The while loop executes a block of code while a condition is true.
- See format as below:

```
while (condition)
{
    code to be executed;
}
```

example 1

```
<html>
<body>
<?php
$i=1;
while($i<=5)
{
    print "The number is " . $i . "<br />";
    $i++;
}
?>
</body>
</html>
```

example 1 explanation

- The example previously defines a loop that starts with $i=1$. It will then increment i with 1, and write some output. Then the condition is checked, and the loop will continue to run as long as i is less than, or equal to 5.

for loop statement

- The for loop is used when you know in advance how many times the script should run.
- See format as below:

```
for (init; condition; increment)
{
    code to be executed;
}
```

init: Mostly used to set a counter (but can be any code to be executed once at the beginning of the loop)

condition: Evaluated for each loop iteration. If it evaluates to TRUE, the loop continues. If it evaluates to FALSE, the loop ends.

increment: Mostly used to increment a counter (but can be any code to be executed at the end of the loop)

example 1

```
<html>
<body>
<?php
for ($i=1; $i<=5; $i++)
{
    print "The number is " . $i . "<br />";
}
?>
</body>
</html>
```

example 1 explanation

- The example previously defines a loop that starts with $i=1$. The loop will continue to run as long as i is less than, or equal to 5. i will increase by 1 each time the loop runs.

foreach loop statement

- The foreach loop is used to loop through arrays.
- For every loop iteration, the value of the current array element is assigned to \$value (and the array pointer is moved by one) - so on the next loop iteration, you'll be looking at the next array value.
- See format as below:

```
foreach ($array as $value)
{
    code to be executed;
}
```

example 1

```
<html>
<body>
<?php
$the_array = array("one","two","three");
foreach ($the_array as $value)
{
    echo $value . "<br />";
}
?>
</body>
</html>
```