

Unit-II

Lecture: 4

Controlling Program Flow (PHP Conditional events and loops)

PHP lets programmers evaluate different conditions during the course of a program and take decisions based on whether these conditions evaluate to true or false. These conditions, and the actions associated with them, are expressed by means of a programming construct called a conditional statement.

- **The if Statement:**

- ✓ simplest conditional statement.
- ✓ if X happens, do Y.

e.g. below example contains a conditional statement that checks if the value of the \$number variable is less than 0 and prints a notification message if so.

```
1 <?php
2 $number = -13;
3 if ($number <0) {
4     echo "number is negative";
5 }
6 ?>
```

- ✓ The condition is always enclosed in the parentheses.
- ✓ If the condition evaluates to true, the code within the curly braces is executed, if it evaluates to false, the code within the curly braces is skipped.

- **The if-else Statement:**

- ✓ An improved version of the if construct that allows to define an alternative set of actions that the program should take when the condition specified evaluates to false.
- ✓ i.e. if X happens do Y; otherwise do Z.

e.g. in below example, an if-else statement is used to account for two possible outcomes: a number less than zero and all other numbers.

```

1  <?php
2  $number = 3;
3  if ($number <0) {
4      echo "number is negative";
5  }
6  else {
7      echo "number is either positive or zero";
8  }
9  ?>
10

```

- **The if-elseif-else Statement:**

- ✓ The if-elseif-else statement lets you chain together multiple if-else statements, thus allowing the programmer to define actions for more than just two possible outcomes.

- ✓ Syntax:

```

if (condition) {
    code to be executed if this condition is true;
} elseif (condition) {
    code to be executed if first condition is false and this condition is true;
} else {
    code to be executed if all conditions are false;
}

```

- ✓ As soon as one of the conditional statements evaluates to true, PHP will execute the corresponding code, skip the remaining conditional tests and jump straight to the lines following the entire if-elseif-else block, So, *even if more than one of the conditional tests evaluates to TRUE, PHP will only execute the code corresponding to first true test.*

- **The switch-case Statement:**

- ✓ An alternative to the if-elseif-else statement is the switch-case statement, which does almost the same thing: it tests a variable against a series of values until it finds a match, and then executes the code corresponding to that match.

Syntax:

```
switch (n) {  
    case label1:  
        code to be executed if n=label1;  
        break;  
    case label2:  
        code to be executed if n=label2;  
        break;  
    case label3:  
        code to be executed if n=label3;  
        break;  
    ...  
    default:  
        code to be executed if n is different from all labels;  
}
```

- ✓ First, a single expression n (most often a variable), is evaluated once. The value of the expression is then compared with the values for each case in the structure.
- ✓ If there is a match, the block of code associated with that case is executed.

Difference between if-elseif-else and switch-case:

- Once PHP finds a **case statement that evaluates to true, it executes not only the code the code corresponding to that case statement, but also the code for all subsequent case statements.**
- If this is not what you want, **add a break statement to the end of each case block to tell PHP to break out** of the switch-case statement block once it executes the code corresponding to the first true case.