PHP If Else

PHP if else statement is used to test condition. There are various ways to use if statement in PHP.

* [if](https://www.javatpoint.com/php-if-else#if)
* [if-else](https://www.javatpoint.com/php-if-else#if-else)
* [if-else-if](https://www.javatpoint.com/php-if-else#if-else-if)
* [nested if](https://www.javatpoint.com/php-if-else#nested-if)

PHP If Statement

PHP if statement allows conditional execution of code. It is executed if condition is true.

If statement is used to executes the block of code exist inside the if statement only if the specified condition is true.

Syntax:-

1. **if**(condition){
2. //code to be executed
3. }

**Example**

1. <?php
2. $num=12;
3. **if**($num<100){
4. echo "$num is less than 100";
5. }
6. ?>
7. **Output:**
8. 12 is less than 100

## PHP If-else Statement

PHP if-else statement is executed whether condition is true or false.

If-else statement is slightly different from if statement. It executes one block of code if the specified condition is **true** and another block of code if the condition is **false**.

**Syntax**

1. **if**(condition){
2. //code to be executed if true
3. }**else**{
4. //code to be executed if false
5. }

 Examples :-

1. <?php
2. $num=12;
3. **if**($num%2==0){
4. echo "$num is even number";
5. }**else**{
6. echo "$num is odd number";
7. }
8. ?>

**Output:**

1. 12 is even number

## PHP If-else-if Statement

The PHP if-else-if is a special statement used to combine multiple if?.else statements. So, we can check multiple conditions using this statement.

**Syntax**

1. **if** (condition1){
2. //code to be executed if condition1 is true
3. } **elseif** (condition2){
4. //code to be executed if condition2 is true
5. } **elseif** (condition3){
6. //code to be executed if condition3 is true
7. ....
8. }  **else**{
9. //code to be executed if all given conditions are false
10. }

**Example**

1. <?php
2. $marks=69;
3. **if** ($marks<33){
4. echo "fail";
5. }
6. **else** **if** ($marks>=34 && $marks<50) {
7. echo "D grade";
8. }
9. **else** **if** ($marks>=50 && $marks<65) {
10. echo "C grade";
11. }
12. **else** **if** ($marks>=65 && $marks<80) {
13. echo "B grade";
14. }
15. **else** **if** ($marks>=80 && $marks<90) {
16. echo "A grade";
17. }
18. **else** **if** ($marks>=90 && $marks<100) {
19. echo "A+ grade";
20. }
21. **else** {
22. echo "Invalid input";
23. }
24. ?>

**Output:**

B Grade

## PHP nested if Statement

The nested if statement contains the if block inside another if block. The inner if statement executes only when specified condition in outer if statement is **true**.

**Syntax**

1. **if** (condition) {
2. //code to be executed if condition is true
3. **if** (condition) {
4. //code to be executed if condition is true
5. }
6. }

**Example**

1. <?php
2. $age = 23;
3. $nationality = "Indian";
4. //applying conditions on nationality and age
5. **if** ($nationality == "Indian")
6. {
7. **if** ($age >= 18) {
8. echo "Eligible to give vote";
9. }
10. **else** {
11. echo "Not eligible to give vote";
12. }
13. }
14. ?>

**Output:**

Eligible to give vote

# PHP Switch

PHP switch statement is used to execute one statement from multiple conditions. It works like PHP if-else-if statement.

### **Syntax**

1. **switch**(expression){
2. **case** value1:
3. //code to be executed
4. **break**;
5. **case** value2:
6. //code to be executed
7. **break**;
8. ......
9. **default**:
10. code to be executed **if** all cases are not matched;
11. }

### **Important points to be noticed about switch case:**

1. The **default** is an optional statement. Even it is not important, that default must always be the last statement.
2. There can be only one **default** in a switch statement. More than one default may lead to a **Fatal** error.
3. Each case can have a **break** statement, which is used to terminate the sequence of statement.
4. The **break** statement is optional to use in switch. If break is not used, all the statements will execute after finding matched case value.
5. PHP allows you to use number, character, string, as well as functions in switch expression.
6. Nesting of switch statements is allowed, but it makes the program more complex and less readable.
7. You can use semicolon (;) instead of colon (:). It will not generate any error.

### **PHP Switch Example**

1. <?php
2. $num=20;
3. **switch**($num){
4. **case** 10:
5. echo("number is equals to 10");
6. **break**;
7. **case** 20:
8. echo("number is equal to 20");
9. **break**;
10. **case** 30:
11. echo("number is equal to 30");
12. **break**;
13. **default**:
14. echo("number is not equal to 10, 20 or 30");
15. }
16. ?>

**Output:**

number is equal to 20

## PHP switch statement with character

**Program to check Vowel and consonant**

We will pass a character in switch expression to check whether it is vowel or constant. If the passed character is A, E, I, O, or U, it will be vowel otherwise consonant.

1. <?php
2. $ch = 'U';
3. **switch** ($ch)
4. {
5. **case** 'a':
6. echo "Given character is vowel";
7. **break**;
8. **case** 'e':
9. echo "Given character is vowel";
10. **break**;
11. **case** 'i':
12. echo "Given character is vowel";
13. **break**;
14. **case** 'o':
15. echo "Given character is vowel";
16. **break**;
17. **case** 'u':
18. echo "Given character is vowel";
19. **break**;
20. **case** 'A':
21. echo "Given character is vowel";
22. **break**;
23. **case** 'E':
24. echo "Given character is vowel";
25. **break**;
26. **case** 'I':
27. echo "Given character is vowel";
28. **break**;
29. **case** 'O':
30. echo "Given character is vowel";
31. **break**;
32. **case** 'U':
33. echo "Given character is vowel";
34. **break**;
35. **default**:
36. echo "Given character is consonant";
37. **break**;
38. }
39. ?>

**Output:**

Given character is vowel

## PHP switch statement with String

PHP allows to pass string in switch expression. Let's see the below example of course duration by passing string in switch case statement.

1. <?php
2. $ch = "B.Tech";
3. **switch** ($ch)
4. {
5. **case** "BCA":
6. echo "BCA is 3 years course";
7. **break**;
8. **case** "Bsc":
9. echo "Bsc is 3 years course";
10. **break**;
11. **case** "B.Tech":
12. echo "B.Tech is 4 years course";
13. **break**;
14. **case** "B.Arch":
15. echo "B.Arch is 5 years course";
16. **break**;
17. **default**:
18. echo "Wrong Choice";
19. **break**;
20. }
21. ?>

**Output:**

B.Tech is 4 years course

## PHP switch statement is fall-through

PHP switch statement is fall-through. It means it will execute all statements after getting the first match, if break statement is not found.

1. <?php
2. $ch = 'c';
3. **switch** ($ch)
4. {
5. **case** 'a':
6. echo "Choice a";
7. **break**;
8. **case** 'b':
9. echo "Choice b";
10. **break**;
11. **case** 'c':
12. echo "Choice c";
13. echo "</br>";
14. **case** 'd':
15. echo "Choice d";
16. echo "</br>";
17. **default**:
18. echo "case a, b, c, and d is not found";
19. }
20. ?>

**Output:**

Choice c

Choice d

case a, b, c, and d is not found

## PHP nested switch statement

Nested switch statement means switch statement inside another switch statement. Sometimes it leads to confusion.

1. <?php
2. $car = "Hyundai";
3. $model = "Tucson";
4. **switch**( $car )
5. {
6. **case** "Honda":
7. **switch**( $model )
8. {
9. **case** "Amaze":
10. echo "Honda Amaze price is 5.93 - 9.79 Lakh.";
11. **break**;
12. **case** "City":
13. echo "Honda City price is 9.91 - 14.31 Lakh.";
14. **break**;
15. }
16. **break**;
17. **case** "Renault":
18. **switch**( $model )
19. {
20. **case** "Duster":
21. echo "Renault Duster price is 9.15 - 14.83 L.";
22. **break**;
23. **case** "Kwid":
24. echo "Renault Kwid price is 3.15 - 5.44 L.";
25. **break**;
26. }
27. **break**;
28. **case** "Hyundai":
29. **switch**( $model )
30. {
31. **case** "Creta":
32. echo "Hyundai Creta price is 11.42 - 18.73 L.";
33. **break**;
34. **case** "Tucson":
35. echo "Hyundai Tucson price is 22.39 - 32.07 L.";
36. **break**;
37. **case** "Xcent":
38. echo "Hyundai Xcent price is 6.5 - 10.05 L.";
39. **break**;
40. }
41. **break**;
42. }
43. ?>

**Output:**

Hyundai Tucson price is 22.39 - 32.07 L.

# PHP For Loop

PHP for loop can be used to traverse set of code for the specified number of times.

It should be used if the number of iterations is known otherwise use while loop. This means for loop is used when you already know how many times you want to execute a block of code.

It allows users to put all the loop related statements in one place. See in the syntax given below:

### **Syntax**

1. **for**(initialization; condition; increment/decrement){
2. //code to be executed
3. }

### **Parameters**

The php for loop is similar to the java/C/C++ for loop. The parameters of for loop have the following meanings:

**initialization** - Initialize the loop counter value. The initial value of the for loop is done only once. This parameter is optional.

**condition** - Evaluate each iteration value. The loop continuously executes until the condition is false. If TRUE, the loop execution continues, otherwise the execution of the loop ends.

**Increment/decrement** - It increments or decrements the value of the variable.

### **Example**

1. <?php
2. **for**($n=1;$n<=10;$n++){
3. echo "$n<br/>";
4. }
5. ?>
6. **Output:**
7. 1
8. 2
9. 3
10. 4
11. 5
12. 6
13. 7
14. 8
15. 9
16. 10

### **Example**

All three parameters are optional, but semicolon (;) is must to pass in for loop. If we don't pass parameters, it will execute infinite.

1. <?php
2. $i = 1;
3. //infinite loop
4. **for** (;;) {
5. echo $i++;
6. echo "</br>";
7. }
8. ?>

**Output:**

1

2

3

4

.

.

.

### **Example**

Below is the example of printing numbers from 1 to 9 in four different ways using for loop.

1. <?php
2. /\* example 1 \*/
4. **for** ($i = 1; $i <= 9; $i++) {
5. echo $i;
6. }
7. echo "</br>";
9. /\* example 2 \*/
11. **for** ($i = 1; ; $i++) {
12. **if** ($i > 9) {
13. **break**;
14. }
15. echo $i;
16. }
17. echo "</br>";
19. /\* example 3 \*/
21. $i = 1;
22. **for** (; ; ) {
23. **if** ($i > 9) {
24. **break**;
25. }
26. echo $i;
27. $i++;
28. }
29. echo "</br>";
31. /\* example 4 \*/
33. **for** ($i = 1; $j = 0; $i <= 9; $j += $i, print $i, $i++);
34. ?>

**Output:**

**123456789 (4 times same out put)**

## PHP Nested For Loop

We can use for loop inside for loop in PHP, it is known as nested for loop. The inner for loop executes only when the outer for loop condition is found **true**.

In case of inner or nested for loop, nested for loop is executed fully for one outer for loop. If outer for loop is to be executed for 3 times and inner for loop for 3 times, inner for loop will be executed 9 times (3 times for 1st outer loop, 3 times for 2nd outer loop and 3 times for 3rd outer loop).

**Example**

1. <?php
2. **for**($i=1;$i<=3;$i++){
3. **for**($j=1;$j<=3;$j++){
4. echo "$i   $j<br/>";
5. }
6. }
7. ?>

**Output:**

1 1

1 2

1 3

2 1

2 2

2 3

3 1

3 2

3 3

123456789123456789

123456789

123456789

123456789

123456789

123456789

123456789

123456789

123456789

123456789

123456789