Conditional Statements and Loops in PHP

Overview

- IF-Else Statements
- Nested IF-Else
- Switch Case
- While Loop
- For Loop
- Do-While Loop
- Goto Statement
- Break and Continue

Conditional Statements and Loops in PHP

- Conditional statements and loops are fundamental constructs in PHP that allow you to control the flow of your program. They help you make decisions, repeat tasks, and handle different scenarios dynamically.

IF-Else Statements

- The **if-else** statement is used to execute a block of code based on whether a condition is true or false. It is one of the most commonly used conditional statements in programming.
- **Syntax**

```
if (condition)
{
// Code to execute if condition is true
}
else
{
// Code to execute if condition is false
}
```

IF-Else Example

```
$age = 18;
if ($age >= 18)
{
   echo "You are an adult.";
}
else
{
   echo "You are a minor.";
}
```

Nested IF-Else

- Nested if-else statements allow you to check multiple conditions within another if or else block. This is useful for handling more complex decision-making scenarios.

Syntax:

```
if (condition1)
{
   if (condition2)
     {
      // Code if both conditions are true
     }
     else {
      // Code if condition1 is true but condition2 is false
     }
} else {
   // Code if condition1 is false
}
```

Nested IF-Else Example

```
$age = 20;
$hasLicense = true;
if ($age >= 18)
{
   if ($hasLicense)
     {
      echo "You can drive.";
     }
     else
     {
      echo "You need a license to drive.";
     }
} else {
   echo "You are too young to drive.";
}
```

Switch Case

- The switch statement is used to perform different actions based on the value of a variable. It is a cleaner alternative to multiple if-else statements when dealing with many conditions.
- **Syntax**

```
switch (variable) {
   case value1:
      // Code to execute
      break;
   case value2:
      // Code to execute
      break;
   default:
      // Code if no case matches
}
```

Switch Case Example

```
$day = "Monday";
switch ($day)
{
   case "Monday":  echo "Today is Monday.";
      break;
   case "Tuesday": echo "Today is Tuesday.";
      break;
   default: echo "Today is neither Monday nor
   Tuesday.";
}
```

While Loop

- The while loop repeatedly executes a block of code as long as the specified condition is true. It is useful when you don’t know in advance how many times the loop should run.

- **Syntax**

```
while (condition)
{
   // Code to execute
}
```

While Loop Example

```
$i = 1;
while ($i <= 5)
{
    echo "The number is: $i <br>";
    $i++;
}
```

For Loop

- The for loop is used when you know how many times you want to execute a block of code. It combines initialization, condition checking, and iteration in one line.

- **Syntax**

```
for (init; condition; increment)
{
   // Code to execute
}
```

For Loop Example

```
for ($i = 1; $i <= 5; $i++)
{
    echo "The number is: $i <br>";
}
```

Do-While Loop

- The do-while loop is similar to the while loop, but it guarantees that the block of code is executed at least once, even if the condition is false.
- **Syntax**

```
do
{
   // Code to execute
} while (condition);
```

Do-While Loop Example

```
$i = 1;
do
{
    echo "The number is: $i <br>";
    $i++;
} while ($i <= 5);
```

Goto Statement

- The goto statement allows you to jump to a specific label in your code. It is rarely used because it can make code harder to read and maintain.
- **Syntax**

```
goto label;
// Code to skip
label:
// Code to execute
```

Goto Statement Example

```
goto myLabel;
echo "This will not be executed.";
myLabel:
echo "This will be executed.";
```

Break Statement

- The break statement is used to exit a loop or a switch statement prematurely. It stops the execution of the loop or switch immediately.
- **Example**

```
for ($i = 1; $i <= 10; $i++) {
    if ($i == 5) {
        break;
    }
    echo "The number is: $i <br>";
}
```

Continue Statement

- The continue statement skips the rest of the current iteration of a loop and moves to the next iteration. It is useful for skipping specific values or conditions.
- **Example**

```
for ($i = 1; $i <= 5; $i++)
{
   if ($i == 3) {
      continue;
   }
   echo "The number is: $i <br>";
}
```

Questions ?