CONTROL STRUCTURES

Control Structures

- Control structures are blocks of code that dictate the flow of execution based on conditions or repetitions.

- Importance: They allow developers to execute certain parts of code based on logical conditions, making the application dynamic and responsive.

- Types of Control Structures:

✓Sequential: Default mode where statements are executed line by line.

✓Conditional: Execute code based on true/false conditions.

✓Looping: Repeat code multiple times.

Control Structures

- **Sequential**: Code runs in the order it's written.

- **Conditional**: It includes:

✓If statements

✓if-else statements

✓switch statements.

- **Looping**: It includes:

✓While loops

✓Do-while loops

✓For loops

✓Foreach loops

Why Control Structures?

- Role in Decision Making: Allow the program to make decisions (e.g. ‘if’, ‘if-else’, and ‘switch’).
- Role in Repetitive Tasks: Handle repetitive actions efficiently (e.g., ‘for’ loops).
- Examples: Real-world analogies, like traffic lights (conditions) or daily routines (loops).

Conditional Statements

- **Purpose**: Control which blocks of code are executed based on specific conditions.

- **Importance**: They make programs dynamic by allowing decisions to be made.

- **Conditional**: It includes:

✓If statements

✓if-else statements

✓switch statements.

Conditional Statements(contd.)

- There are several statements in PHP that you can use to make decisions:

- The if statement
- The if...else statement
- The if...elseif....else statement
- The switch...case statement

The if Statement

- The if statement is used to execute a block of code only if the specified condition evaluates to true.
- This is the simplest PHP's conditional statements and can be written like:

```
<?php
$d = date("D");
if($d == "Tue"){
    echo "It's Tuesday!";
}
?>
```

The if Statement(contd.)

```
<?php
$t = date("H");

if ($t < "20") {
  echo "Have a good day!";
}
?>
```

The if Statement(contd.)

```
<?php
$num=12;
if($num<100){
echo "$num is less than 100";
}
?>
```

OUTPUT:

```
12 is less than 100
```

The if...else Statement

- The if...else statement allows you to execute one block of code if the specified condition is evaluates to true and another block of code if it is evaluates to false.

```
<?php
$d = date("D");
if($d == "Fri"){
    echo "Have a nice weekend!";
} else{
    echo "Have a nice day!";
}
?>
```

The if...else Statement(contd.)

```
<?php
$t = date("H");

if ($t < "20") {
  echo "Have a good day!";
} else {
  echo "Have a good night!";
}
?>
```

The if...else Statement(contd.)

```
<?php
$num=13;
if($num%2==0){
echo "$num is even number";
}else{
echo "$num is odd number";
}
?>
```

OUTPUT:

```
13 is odd number
```

The if...elseif...else Statement

- The if...elseif...else a special statement that is used to combine multiple if...else statements.

```
<?php
$d = date("D");
if($d == "Fri"){
    echo "Have a nice weekend!";
} elseif($d == "Sun"){
    echo "Have a nice Sunday!";
} else{
    echo "Have a nice day!";
}
?>
```

The if...elseif...else Statement(contd.)

```
<?php
$t = date("H");

if ($t < "10") {
  echo "Have a good morning!";
} elseif ($t < "20") {
  echo "Have a good day!";
} else {
  echo "Have a good night!";
}
?>
```

The if...elseif...else Statement(contd.)

```
<?php
    $marks=69;
    if ($marks<33){
        echo "fail";
    }
    else if ($marks>=34 && $marks<50) {
        echo "D grade";
    }
    else if ($marks>=50 && $marks<65) {
       echo "C grade";
    }
    else if ($marks>=65 && $marks<80) {
        echo "B grade";
    }
    else if ($marks>=80 && $marks<90) {
        echo "A grade";
    }
    else if ($marks>=90 && $marks<100) {
        echo "A+ grade";
    }
   else {
        echo "Invalid input";
    }
?>
```

OUTPUT:

B grade

The Ternary Operator

- The ternary operator provides a shorthand way of writing the if...else statements.

- The ternary operator is represented by the question mark (?) symbol and it takes three operands: a condition to check, a result for true, and a result for false.

```
<?php
$age = 15;
echo ($age < 18) ? 'Child' : 'Adult';
?>
```

OUTPUT:

```
Child
```

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.

```
<?php
    $age = 23;
    $nationality = "Indian";
    //applying conditions on nationality and age
    if ($nationality == "Indian")
    {
        if ($age >= 18) {
            echo "Eligible to give vote";
        }
        else {
            echo "Not eligible to give vote";
        }
    }
?>
```

OUTPUT:

```
Eligible to give vote
```

PHP switch Statement

- The switch statement is used to perform different actions based on different conditions.

- Use the switch statement to select one of many blocks of code to be executed.

PHP switch Statement(contd.)

```
<?php
$num=20;
switch($num){
case 10:
echo("number is equals to 10");
break;
case 20:
echo("number is equal to 20");
break;
case 30:
echo("number is equal to 30");
break;
default:
echo("number is not equal to 10, 20 or 30");
}
?>
```

OUTPUT:

```
number is equal to 20
```

PHP switch statement with character

```
<?php
$ch = 'k';
switch($ch){
case 'a':
echo 'It is a vowel';
break;
case 'e':
echo 'It is a vowel';
break;
case 'i':
echo 'It is a vowel';
break;
case 'o':
echo 'It is a vowel';
break;
case 'u':
echo 'It is a vowel';
break;
default:
echo 'It is a consonant';
}
?>
```

OUTPUT:

```
It is a consonant
```

PHP switch statement with String

```php
<?php
    $ch = "B.Tech";
    switch ($ch)
    {
        case "BCA":
            echo "BCA is 3 years course";
            break;
        case "Bsc":
            echo "Bsc is 3 years course";
            break;
        case "B.Tech":
            echo "B.Tech is 4 years course";
            break;
        case "B.Arch":
            echo "B.Arch is 5 years course";
            break;
        default:
            echo "Wrong Choice";
            break;
    }
?>
```

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.

```
<?php
    $ch = 'c';
    switch ($ch)
    {
        case 'a':
            echo "Choice a";
            break;
        case 'b':
            echo "Choice b";
            break;
        case 'c':
            echo "Choice c";
            echo "</br>";
        case 'd':
            echo "Choice d";
            echo "</br>";
        default:
            echo "case a, b, c, and d is not found";
    }
?>
```

```
OUTPUT:
Choice c
Choice d
case a, b, c, and d is not found
```

PHP nested switch statement

```
<?php
$author = "Stephen King";
$book ="The silence of the lambs";
switch($author)
{
case "JK Rowling":
{
    switch($book)
    {
    case "Harry Potter1";
    echo "Harry Potter1, The price is 300$";
    break;
    case "Harry Potter2";
    echo "Harry Potter2, The price is 200$";
    break;
    default:
    echo "Author found but not the book";
    }
}
break;
case "Stephen King":
{
    switch($book)
    {
    case "Hannibal";
    echo "Hannibal, The price is 500$";
    break;
    case "The silence of the lambs";
    echo "The silence of the lambs, The price is 700$";
    break;
    default:
    echo "Author found but not the book";
    }
}
break;
default:
echo "Author not found";
}
?>
```

OUTPUT:

```
The silence of the lambs, The price is 700$
```