PHP -IF STATEMENTS

PHP Conditional Statements

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

In PHP we have the following conditional statements:

- if statement - executes some code if one condition is true
- if...else statement - executes some code if a condition is true and another code if that condition is false
- if...elseif...else statement - executes different codes for more than two conditions
- switch statement - selects one of many blocks of code to be executed

PHP - The if Statement

The if statement executes some code if one condition is true.

Syntax

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

Example

Output "Have a good day!" if 5 is larger than 3:

```
if (5 > 3) {
  echo "Have a good day!";
```

```
}
```

We can also use variables in the if statement:

```
$t = 14;


if ($t < 20) {

  echo "Have a good day!";

}
```

Comparison Operators

If statements usually contain conditions that compare two values.

```
$t = 14;


if ($t == 14) {

  echo "Have a good day!";

}
```

Logical Operators

To check more than one condition, we can use logical operators, like the && operator:

```php
$a = 200;

$b = 33;

$c = 500;


if ($a > $b && $c < $a ) {

  echo "Both conditions are true";

}
```

PHP - The if...else Statement

The if...else statement executes some code if a condition is true and another code if that condition is false.

Syntax

```php
if (condition) {

  // code to be executed if condition is true;

} else {

  // code to be executed if condition is false;

}
```

Example

Output "Have a good day!" if the current time is less than 20, and "Have a good night!" otherwise:

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

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

PHP - The if...elseif...else Statement

Short Hand If

To write shorter code, you can write if statements on one line.

Example

One-line if statement:

```php
$a = 5;


if ($a < 10) $b = "Hello";


echo $b
```

Short Hand If...Else

if...else statements can also be written in one line, but the syntax is a bit different.

Example

One-line if...else statement:

```php
$a = 13;
```

```php
$b = $a < 10 ? "Hello" : "Good Bye";
```

```php
echo $b;
```

Nested If

You can have if statements inside if statements, this is called *nested* if statements.

Example

An if inside an if:

```php
$a = 13;
```

```php
if ($a > 10) {
  echo "Above 10";
  if ($a > 20) {
    echo " and also above 20";
  } else {
    echo " but not above 20";
  }
}
```

Certainly! Here are a few random examples of if statements in C:

Example 1: Checking if a Number is Positive, Negative, or Zero

```c
#include <stdio.h>

int main() {
    int num;

    printf("Enter an integer: ");
    scanf("%d", &num);

    if (num > 0) {
        printf("%d is positive.\n", num);
    } else if (num < 0) {
        printf("%d is negative.\n", num);
    } else {
        printf("The number is zero.\n");
    }

    return 0;
}
```

Example 2: Checking if a Year is a Leap Year

```c
#include <stdio.h>

int main() {
```

```c
    int year;
    printf("Enter a year: ");
    scanf("%d", &year);

    if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
        printf("%d is a leap year.\n", year);
    } else {
        printf("%d is not a leap year.\n", year);
    }

    return 0;
}
```

Example 3: Determining the Largest among Three Numbers

```c
#include <stdio.h>

int main() {
    int num1, num2, num3;

    printf("Enter three numbers: ");
    scanf("%d %d %d", &num1, &num2, &num3);

    if (num1 >= num2 && num1 >= num3) {
        printf("%d is the largest.\n", num1);
    } else if (num2 >= num1 && num2 >= num3) {
```

```c
        printf("%d is the largest.\n", num2);
    } else {
        printf("%d is the largest.\n", num3);
    }


    return 0;
}
```

These examples showcase different scenarios where if statements are used to make decisions based on conditions. Feel free to modify and experiment with these examples to further understand how if statements work in C.