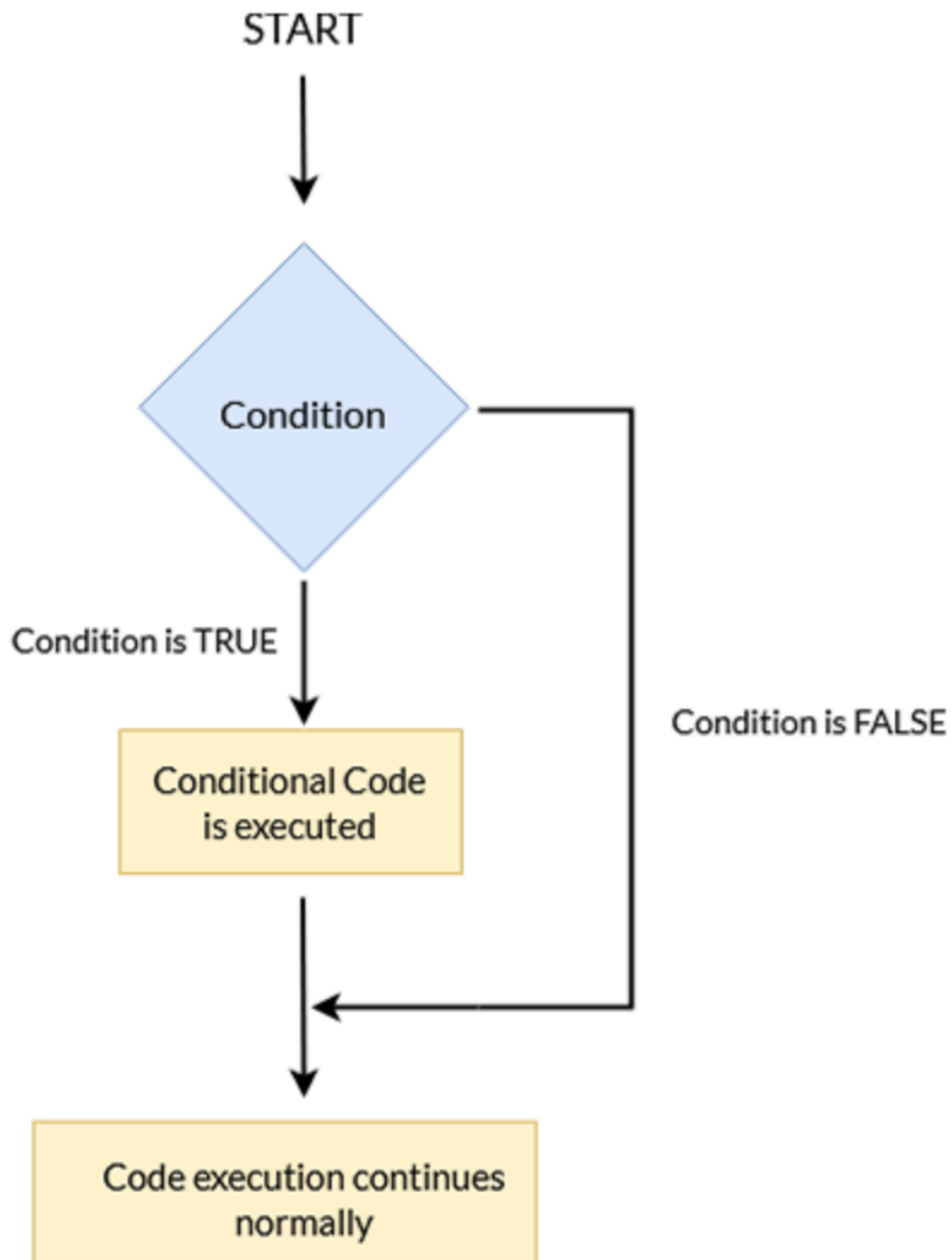


## PHP Control Structures - Written by Anthony Adams

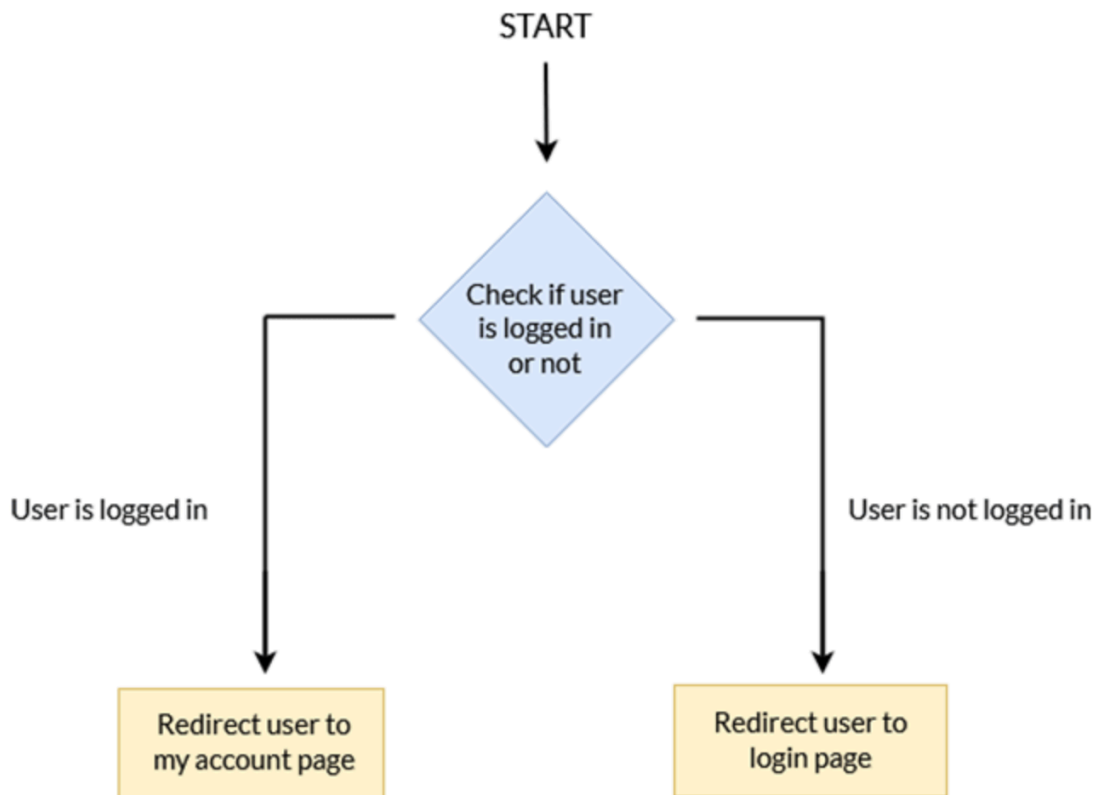
When we write code. We write it character by character, line by line. When a computer reads code, it reads it the same way. A control structure allows us to control and alter the flow of execution how we want it, rather than how the computer would naturally read and execute it.

Control structures are a core feature of PHP and allow the script to respond to different inputs or situations.



In the above diagram there is a condition and if that is met (true), the conditional code is executed.

If the condition is not met (false) the conditional code is bypassed and the rest of the code will be executed.



In the above example, the program will check if the user is logged in. If the user is in fact logged in they will be directed to a user account page. If not they will be directed to a login page.

PHP supports a number of different control structures:

- if
- else
- elseif
- switch
- while
- do-while
- for
- foreach

And more. Today we will look at a couple of these.

## If Statement

If statement allows us to execute a piece of code if the expression within it is true.

```
<?php
    $num = 15;
    if ($num > 10)
    {
```

```
        echo "The number is  
greater than 10";  
    }  
  
?>
```

In the above example:

We create a variable named \$num that contains the number of 15.

We create an if statement, stating that if the variable named \$num is greater than the number 10

Echo out the conditional string "The number is greater than 10";

We could add to this which brings us to the next part.

## Else Statement

In the above example we will get a message telling us the variable number is greater than 10. But what if the number is lower than 10? In the above case nothing will happen.

We could simply add an else statement.

```
<?php
    $num = 9;

    if ($num > 10)
    {
        echo "The number is
greater than 10";
    }

    else
    {
        echo "The number is
less than 10";
    }

?>
```

Now if we change the number of the variable to

less than 10 we will get the second message.

## Elseif Statement

In the above example, if the number is greater than 10 we get a message saying so. If the number is less than 10 we also get a message saying so. But what if the number is the same as 10?

In its current state we would get a message saying that the number is less than 10 even though it was the same. Which is clearly not true. We can fix this using an else if statement.

```
<?php
    //The variable
    $num = 10;

    //If statement - If the
number is greater than 10...
    if ($num > 10)
    {
```

```
        //The conditional
output
        echo "The number is
greater than 10";
    }

    //Elseif statement if the
variable of $num is equal to 10
    elseif ($num == 10)
    {

        //The conditional
output
        echo "The number is the
same as 10";
    }

    //Else statement
    else
    {

        //The conditional
output
        echo "The number is
less than 10";
    }
```

?>

## Switch Statement

There is a totally different way we can write this above example. We can do this using switch a switch statement.

```
<?php

//The variable we want to check
$number = 20;

//Switch statement with the
variable we want to check loaded in
the ()
switch ($number) {

    //In the case that the
variable is greater than 20
    case $number > 20:
        //echo out this
conditional
        echo "The number is
greater then 20";
```

```
        //Break
        break;

        //In the case that the
variable is equal to 20
        case $number == 20:
            //Echo out this
conditional
            echo "The number is the
same as 20";
            //Break
            break;

        //In the case that the
variable is less than 20
        case $number < 20:
            //Echo out this
conditional
            echo "The number is
less than 20";
            //Break
            break;
    }
?>
```

We can use this for other forms of data as well:

```
<?php

    $food = "oxtail";

    //Switch statement with the
variable we want to check loaded in
the ()
    switch ($food) {

        //We can have multiple
cases in case there is different
ways to spell it
        case "Ox Tail":
        case "ox tail":
        case "Oxtail":
        case "oxtail":
            //echo out this
conditional
            echo "Ox Tail is my
fave dish and the best part of the
cow 🐮 ";
            //Break
            break;

        //Same thing here with the
```

```
different spelling
    case "Curry Goat":
    case "curry goat":
        //Echo out this
conditional
        echo "Curry Goat is a
stewed dish which often contains
potato and possibly chick peas🍲";
        //Break
        break;

        //Same thing here with the
different spelling
    case "Patty":
    case "patty":
        //Echo out this
conditional
        echo "Patties are also
a staple all over the world 🌍";
        //Break
        break;

        //A default answer
default:
        //Echo out a default .
There is no need for a break after
```

```
seeing as it's the last one.  
        echo "It's ok I  
guess.";  
    }  
?>
```

In the above example we want to have different things pop up depending on the different food we enter. We can also add multiple cases (In the case above for if its spelled with caps or all lower case and so on).

## Links

We can pass a link through in any string. Just like we can pass through any HTML within a string. It will render HTML.

```
<?php  
  
    //This is just a link being  
passed through in a string. But note  
that it uses both single and double  
quotes.  
    $text = 'my site is <a
```

```
href="http://google.com">Here</a>';  
  
    echo $text  
  
?>
```

Here is a more complex way of passing links

```
<?php  
    $link_name = 'Humber  
College';  
    $link_url = 'https://  
humber.ca';  
    $link_description = 'Humber  
College - Technology';  
?>  
  
    <h2>Here is the link: <a  
href="<?php echo $link_url ?>"><?  
php echo $link_name ?> </a></h2>  
    <p><?php echo  
$link_description ?></p>
```