if, elif, else Statements

`if` Statements in Python allows us to tell the computer to perform alternative actions based on a certain set of results.

Verbally, we can imagine we are telling the computer:

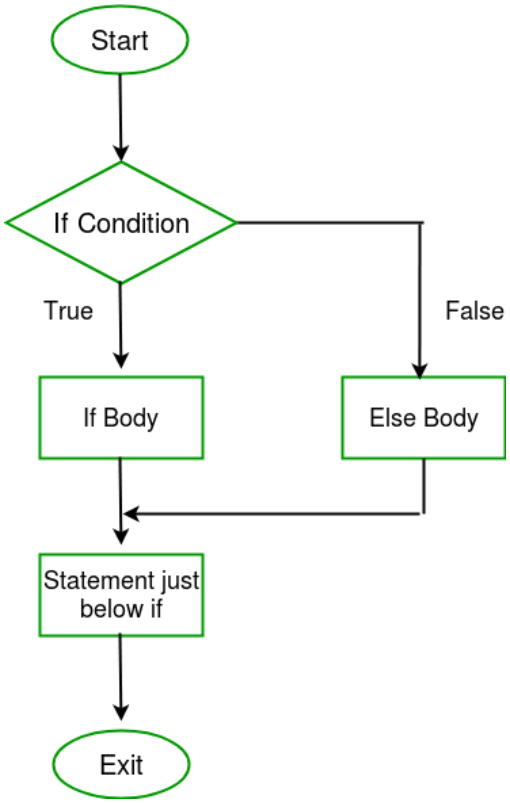
"Hey if this case happens, perform some action"

We can then expand the idea further with `elif` and `else` statements, which allow us to tell the computer:

"Hey if this case happens, perform some action. Else, if another case happens, perform some other action. Else, if *none* of the above cases happened, perform this action."

Let's go ahead and look at the syntax format for `if` statements to get a better idea of this:

```
if case1:
    perform action1
elif case2:
    perform action2
else:
    perform action3
```




```
## First Example

Let's see a quick example of this:
```

In [1]:

```
if True:
    print('It was true!')
```

```
It was true!
```

Let's add in some else logic:

In [2]:

```
x = False

if x:
    print('x was True!')
else:
    print('I will be printed in any case where x is not true')
```

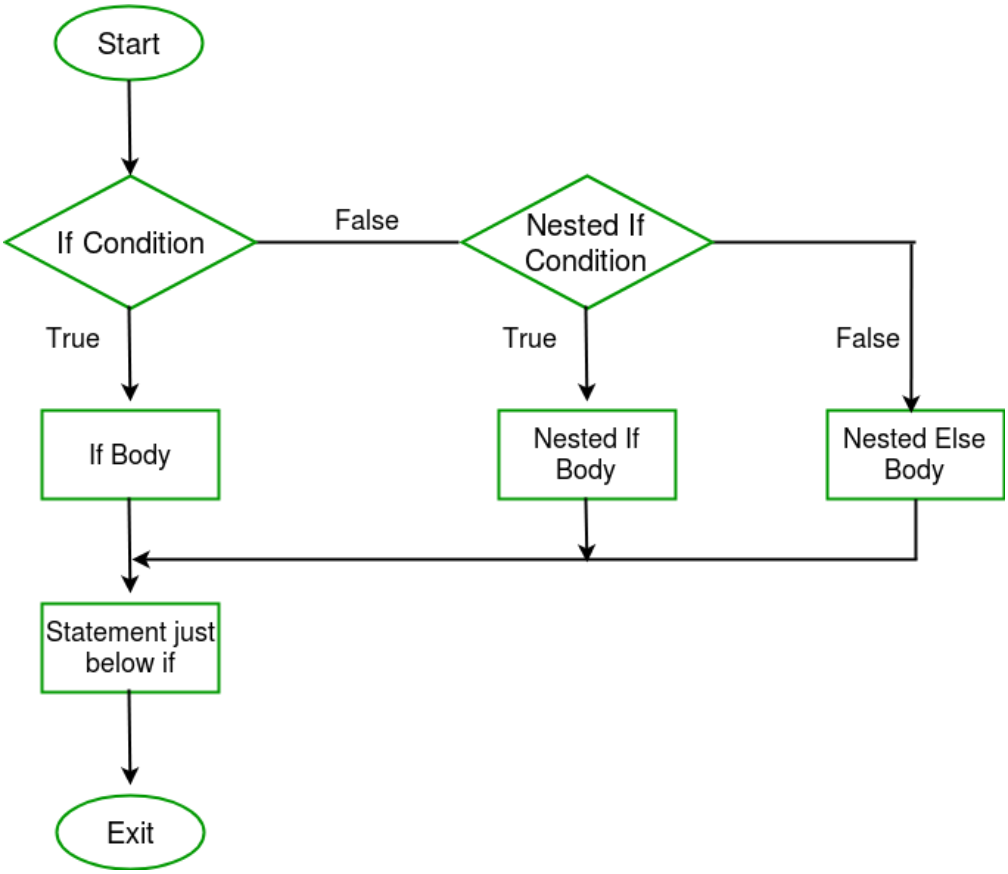
```
I will be printed in any case where x is not true
```

Multiple Branches

Let's get a fuller picture of how far `if`, `elif`, and `else` can take us!

We write this out in a nested structure. Take note of how the `if`, `elif`, and `else` line up in the code. This can help you see what `if` is related to what `elif` or `else` statements.

We'll reintroduce a comparison syntax for Python.



In [3]:

```
loc = 'Bank'

if loc == 'Bank':
    print('Welcome to the bank!')
else:
    print('Where are you?')
```

```
Welcome to the bank!
```

Note how the nested `if` statements are each checked until a True boolean causes the nested code below it to run. You should also note that you can put in as many `elif` statements as you want before you close off with an `else` .

Let's create two more simple examples for the `if` , `elif` , and `else` statements:

In [4]:

```
person = 'George'

if person == 'Sammy':
    print('Welcome Sammy!')
elif person =='George':
    print('Welcome George!')
else:
    print("Welcome, what's your name?")
```

```
Welcome George!
```

Indentation

It is important to keep a good understanding of how indentation works in Python to maintain the structure and order of your code. We will touch on this topic again when we start building out functions!