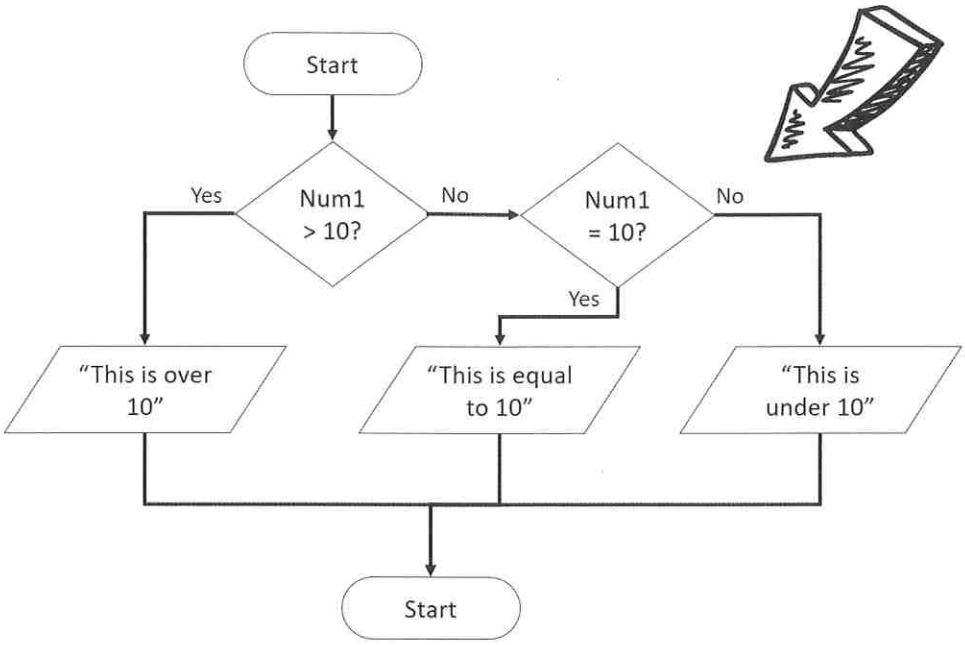
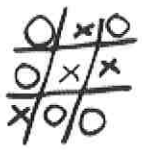
If Statements

Explanation

`If statements` allow your program to make a decision and change the route that is taken through the program.



Below is how the if statement for this flow chart would look in Python.



```
if num1 > 10:
    print("This is over 10")
elif num1 == 10:
    print("This is equal to 10")
else:
    print("This is under 10")
```

Indenting Lines of Code

Indenting is very important in Python as it shows the lines that are dependent on others, as shown in the example on the previous page. In order to indent text you can use your [**Tab**] key or you can press your [**space key**] five times. The [**backspace**] key will remove indents.

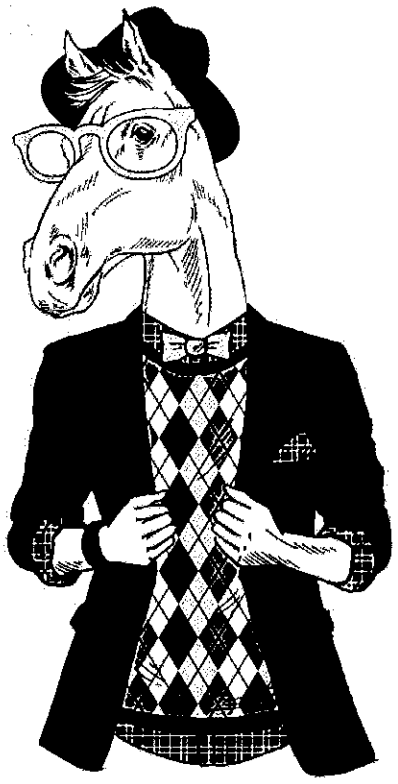
The first line of the if statement tests a condition, and if that condition is met (i.e. the first condition is true) then the lines of code directly below it are run. If it is not met (i.e. the first condition is false) it will test the second condition, if there is one, and so on. Below are examples of the different comparison and logical operators you can use in the condition line of your if statement.

Comparison Operators

Operator	Description
==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

Logical Operators

Operator	Description
and	Both conditions must be met
or	Either condition must be met



Example Code

Please note: In the examples shown, num is a variable entered by the user that has been stored as an integer.

```
if num > 10:
 print("This is over 10")
else:
 print("This is not over 10")
```

If num1 is over 10, it will display the message "This is over 10", otherwise it will display the message "This is under 10".

```
if num > 10:
 print("This is over 10")
elif num == 10:
 print("This is equal to 10")
else:
 print("This is under 10")
```

If num1 is over 10, it will display the message "This is over 10", otherwise it will check the next condition. If num1 is equal to 10, it will display the message "This is equal to 10". Otherwise, if neither of the first two conditions have been met, it will display the message "This is under 10".

```
if num >= 10:
 if num <= 20:
  print("This is between 10 and 20")
 else:
  print("This is over 20")
else:
 print("This is under 10")
```

If num1 is 10 or more then it will test another if statement to see if num1 is less than or equal to 20. If it is, it will display the message "This is between 10 and 20". If num1 is not less than or equal to 20 then it will display the message "This is over 20". If num1 is not over 10, it will display the message "This is under 10".

```
text = str.lower(text)
```

Changes the text to lower case. As Python is case sensitive, this changes the data input by the user into lower case so it is easier to check.

```
num = int(input("Enter a number between 10 and 20: "))
if num >= 10 and num <= 20:
 print("Thank you")
else:
 print("Out of range")
```

This uses **and** to test multiple conditions in the if statement. Both the conditions must be met to produce the output "Thank you".

```
num = int(input("Enter an EVEN number between 1 and 5: "))
if num == 2 or num == 4:
 print("Thank you")
else:
 print("Incorrect")
```

This uses **or** to test the conditions in the if statement. Just one condition must be met to display the output "Thank you".