Conditions in Python

Condition Statements

Comparison Operators

Comparison operations compare some value or operand and, based on a condition, they produce a Boolean. When comparing two values you can use these operators:

- equal: `==`
- not equal: `!=`
- greater than: `>`
- less than: `<`
- greater than or equal to: `>=`
- less than or equal to: `<=`

Let's assign `a` a value of 5. Use the equality operator denoted with two equal `==` signs to determine if two values are equal. The case below compares the variable `a` with 6.

Branching

Branching allows us to run different statements for different inputs. It is helpful to think of an **if statement** as a locked room, if the statement is **True** we can enter the room and your program will run some predefined tasks, but if the statement is **False** the program will ignore the task.

For example, consider the blue rectangle representing an ACDC concert. If the individual is older than 18, they can enter the ACDC concert. If they are 18 or younger than 18 they cannot enter the concert.

Use the condition statements learned before as the conditions need to be checked in the **if statement**. The syntax is as simple as `if condition statement :`, which contains a word `if`, any condition statement, and a colon at the end. Start your tasks which need to be executed under this condition in a new line with an indent. The lines of code after the colon and with an indent will only be executed when the **if statement** is **True**. The tasks will end when the line of code does not contain the indent.

In [1]:
```
# If statement example

Marks = 50
if Marks > 59:
    print(f"Result of {Marks}:", "Passed")
else:
    print(f"Result of {Marks}:", "Failed")
```

```
Result of 50: Failed
```

In [2]:
```
Marks = int(input("Enter Marks: "))
if Marks > 59:
    res = "Pass"
else:
    res = "Fail"

print(f"Result of {Marks}:", res)
```

```
Result of 80: Pass
```

In [3]:
```
Data = int(input("Enter Marks: "))
if Data % 2 == 0:
    res = "Even"
else:
    res = "Odd"

print(f"Result of {Data}:", res)
```

```
Result of 7: Odd
```

In [4]:
```
Marks = int(input("Enter Marks: "))
if Marks > 69:
    Grade = "B"
elif Marks > 59:
    Grade = "C"
else:
    Grade = "F"

print(f"Result of {Marks}: {Grade}")
```

```
Result of 65: C
```

Logical operators

Sometimes you want to check more than one condition at once. For example, you might want to check if one condition and another condition is **True**. Logical operators allow you to combine or modify conditions.

- `and`
- `or`
- `not`

These operators are summarized for two variables using the following truth tables:

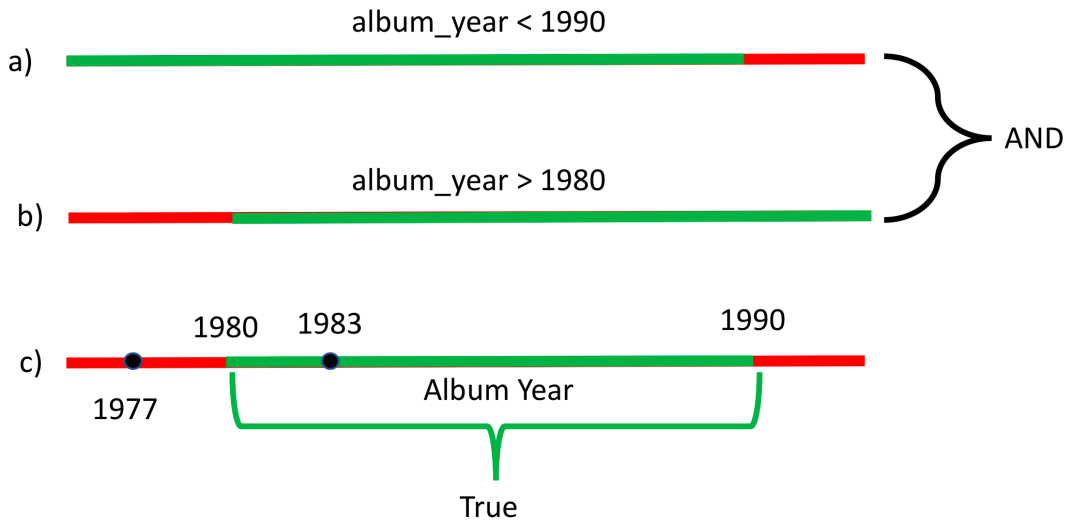
A	B	A & B
False	False	False
False	True	False
True	False	False
True	True	True

A	B	A or B
False	False	False
False	True	True
True	False	Ture
True	True	True

A	A!
False	True
True	False

The `and` statement is only **True** when both conditions are true. The `or` statement is true if one condition is **True**. The `not` statement outputs the opposite truth value.

Let's see how to determine if an album was released after 1979 (1979 is not included) and before 1990 (1990 is not included). The time periods between 1980 and 1989 satisfy this condition. This is demonstrated in the figure below. The green on lines **a** and **b** represents periods where the statement is **True**. The green on line **c** represents where both conditions are **True**, this corresponds to where the green regions overlap.



The block of code to perform this check is given by:

To determine if an album was released before 1980 (~ - 1979) or after 1989 (1990 - ~), an **or** statement can be used. Periods before 1980 (~ - 1979) or after 1989 (1990 - ~) satisfy this condition. This is demonstrated in the following figure, the color green in **a** and **b** represents periods where the statement is true. The color green in **c** represents where at least one of the conditions are true.

In [9]:
```
Marks = int(input("Enter Marks: "))
if Marks <= 59 and Marks >= 0:
    Grd = "F"
elif Marks <= 69 and Marks >= 60:
    Grd = "C"
elif Marks <= 79 and Marks >= 70:
    Grd = "B"
else:
    Grd = "N/A"

print(Grd)
```

```
N/A
```

The block of code to perform this check is given by:

The `not` statement checks if the statement is false:

In [10]:
```
a = 9
b = 2
c = a + b
print(f"Sum of {a} and {b} is:", c)
```

```
Sum of 9 and 2 is: 11
```

In [11]:
```
a = float(input("Enter 1st Number: "))
b = float(input("Enter 2nd Number: "))
c = a + b
print(f"Sum of {a} and {b} is:", c)
```

```
Sum of 6.0 and 8.0 is: 14.0
```

In [ ]:
```
a = float(input("Enter 1st Number: "))
b = float(input("Enter 2nd Number: "))
Op = (input("Enter an Operator e.g. +,-: "))
if Op == "+":
    c = a + b
elif Op == "-":
    c = a - b
elif Op == "*":
    c = a * b
else:
    c = "N/A"

print(f"{a} {Op} {b} is", c)
```

In [ ]:
```

```