Logical Operators in Python

Logical operators are used in “conditional” statements when we need more than one condition to be met.

- and - checks whether two or more conditions are true
- or - checks whether ‘at least’ one condition is true
- not - check when a condition is not true

The ‘and’ logical operator is useful to find results within a particular range.

```
temp = 50

if temp > 19 and temp < 26:
    print("The temperature is very comfortable.")
elif temp < 0 or temp > 40:
    print("Stay indoors, it is a bad day outside")
else:
    print(f"{temp} degrees Celsius is uncomfortable.")
```

In the above example: ***if temp > 19 and temp < 26:*** the entire statement is only true when both conditions are met. For: ***elif temp < 0 or temp > 40:*** the entire statement is deemed true if either of the two conditions are true.

Exercise

Re-write this code asking the user to provide the temperature in Celsius. Extend the conditional code to check whether the temperature is over 55^{o}C and state “humans can no longer dissipate heat beyond 55degC – find a cooler place!”

When using **Boolean** conditionals, we can drop the '== True' part, it is assumed.

```
sunny = True

if sunny:    # <-- if sunny == True
    print("It is a sunny day.")
else:
    print("It is not a sunny day.")
```

We use the '**not**' logical operator for Boolean conditionals, which flip True to False.

```
sunny = True

if not sunny:    # <-- if sunny == False
    print("It is not a sunny day.")
else:
    print("It is a sunny day.")
```

Exercise:

Extend this code to first ask the user for input: "Is it a sunny day outside (Y/N)?". Convert their answer to a Boolean so that anything entered other than Y will return False.

Solution:

```
sunny = input("Is it sunny outside (Y/N)?: ")
if sunny == "Y":
    sunny = True
else:
    sunny = False

if sunny:    # <-- if sunny == True
    print("It is a sunny day.")
else:
    print("It is not a sunny day.")
```