1- IF and ELSE

In programming, we often ask yes or no questions, and decide to do something based on the answer.

1.1 IF Statements

In [5]:
```
age = 13
if age > 20:
    Print ('you are too good!')
```

Python block

A block is the structure of code to separate part of the code from another part of the code.

When you change the indentation, you're generally creating new blocks.

In [4]:
```
age = 25
if age > 20:
    print('You are too good!')
    print('Why are you crying?')
    print('Why aren\'t you mowing a lawn or sorting papers?')
```

```
You are too good!
Why are you crying?
Why aren't you mowing a lawn or sorting papers?
```

1.2 Else Statement

In [6]:
```
age = 11
if age == 11:
    print("A  goat fell in the river!")
else:
    print("Ohh.  It's a scary condition.")
```

```
A  goat fell in the river!
```

In [7]:
```
age = 10
if age == 11:
    print("A  goat fell in the river!")
else:
    print("Ohh.  It's a scary condition.")
```

```
Ohh.  It's a scary condition.
```

1.3- IF AND ELIF (else-if) STATEMENTS

In [8]:
```
age = 12
if age == 10:
    print("What do you call to your brother?")
    print("Sunday!")
elif age == 11:
    print("What did the umair say to the uzair?")
    print("Brother! Brother!")
elif age == 12:
    print("What did ali say to amir?")
    print("Hi guys!")
elif age == 13:
    print("Why wasn't nawaz afraid of imran?")
    print("Because rather than eating corruption rupees.")
else:
    print("Huh? Great Theft")
```

```
What did ali say to amir?
Hi guys!
```

Combining Conditions

In [10]:
```
age = 10
if age == 10 or age == 11 or age == 12 or age == 13:
    print('What is in pharma industry? medicines!')
else:
    print('Cardiac Stimulamts')
```

```
What is in pharma industry? medicines!
```

In [11]:
```
age = 17
if age == 10 or age == 11 or age == 12 or age == 13:
    print('What is in pharma industry? medicines!')
else:
    print('Cardiac Stimulamts')
```

```
Cardiac Stimulamts
```

In [12]:
```
## Use greater than, less than operator
age=10
if age >= 10 and age <= 13:
    print('What is in drugs A remedy!')
else:
    print('Relax')
```

```
What is in drugs A remedy!
```

In [13]:
```
age=14
if age >= 10 and age <= 13:
    print('What is in drugs A remedy!')
else:
    print('Relax')
```

```
Relax
```

In [1]:
```
# Difference between string and number
age = 10
if age == 10:
    print("What's the best way to learn Islam?")
    print("From my point of view its not difficult")
```

```
What's the best way to learn Islam?
From my point of view its not difficult
```

In [2]:
```
age = '10'
if age == 10:
    print("What's the best way to learn Islam?")
    print("From my point of view its not difficult")
```

Converting variables into numbers and numbers into variables

In [5]:
```
# String into Number
age = '15'
converted_age = int(age)
```

In [6]:
```
# Number into string
age = 10
converted_age = str(age)
```

In [7]:
```
age = '10'
converted_age = int(age)
if converted_age == 10:
    print("What's the best way to learn Islam?")
    print("From my point of view its not difficult")
```

```
What's the best way to learn Islam?
From my point of view its not difficult
```

In [9]:
```
# use float numbers
age = '10.5'
converted_age = float(age)
if converted_age == 10.5:
    print("What's the best way to learn Islam?")
    print("From my point of view its not difficult")
```

```
What's the best way to learn Islam?
From my point of view its not difficult
```

2- Loops

2.1 For and While Loops

In [11]:
```
for x in range(0, 5):
    print('data')
```

```
data
data
data
data
data
```

In [13]:
```
# Using range and list together
print(list(range(10, 25)))
```

```
[10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]
```

In [14]:
```
for x in range(0, 5):
    print('data %s' %x)
```

```
data 0
data 1
data 2
data 3
data 4
```

In [16]:
```
x = 46
y = 85
while x < 50 and y < 90:
    x = x + 1
    y = y + 1
    print(x, y)
```

```
47 86
48 87
49 88
50 89
```

3- Code and Functions Modules

A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function. A function can return data as a result. Creating a Function In Python a function is defined using the def keyword*

In [18]:
```
def my_function():
     print("Hello from a function")
```

In [19]:
```
def my_function():
      print("Hello from a function")

my_function()
```

```
Hello from a function
```

In [34]:
```
def my_function(fname, lname):
      print(fname + " " + lname)

my_function("Emil", "rana")
```

```
Emil rana
```

In [21]:
```
## Using return
def variable_test ():
    first_variable = 10
    second_variable = 20
    return first_variable * second_variable
print(variable_test())
```

```
200
```

In [24]:
```
# Variable Outside the functions
another_variable = 40
def variable_test2():
    first_variable = 10
    second_variable = 20
    return first_variable * second_variable * another_variable
print(variable_test2())
```

```
8000
```

Module

A module can define functions, classes, and variables. A module can also include runnable code. Grouping related code into a module

In [25]:
```
def add(x, y):
    return (x+y)

def subtract(x, y):
    return (x-y)
```

In [28]:
```
import time
print(time.asctime())
```

```
Wed Aug  2 23:36:53 2023
```

In [29]:
```
age=11
if age >= 10 and age <= 13:
    print('What is Migraine? A headache!')
else:
    print('Very Sad')
```

```
What is Migraine? A headache!
```

5- Python Classes and Objects

Python is an object oriented programming language. Almost everything in Python is an object, with its properties and methods. A Class is like an object constructor, or a "blueprint" for creating objects.Objects are a way of organizing code in a program and breaking things down to make it easier to think about complex ideas.

In [31]:
```
## Create a class
class MyClass:
      x = 5
```

In [33]:
```
## Create an Object
p1 = MyClass()
print(p1.x)
```

```
5
```

In [35]:
```
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age

p1 = Person("Uzair", 36)

print(p1.name)
print(p1.age)
```

```
Uzair
36
```

In [37]:
```
# Creating a class
class Things:
    pass
class Inanimate(Things):
    pass
class Animate(Things):
    pass
class Sidewalks(Inanimate):
    pass
class Animals(Animate):
    pass
class Mammals(Animals):
    pass
class Giraffes(Mammals):
    pass
```

In [40]:
```
# Adding class characteristics as functions
class Animals(Animate):
    def breathe(self):
        pass
    def move(self):
        pass
    def eat_food(self):
        pass
```

6- Python Build in Function

In [41]:
```
abs(313)
```

Out[41]:
```
313
```

In [42]:
```
abs(313.5)
```

Out[42]:
```
313.5
```

In [44]:
```
abs(12.43)
```

Out[44]:
```
12.43
```

In [46]:
```
#random Integer
integer = -24
print('Absolute value of -24 is:', abs(integer))
#random floating number
floating = -33.33
print('Absolute value of -33.33 is:', abs(floating))
```

```
Absolute value of -24 is: 24
Absolute value of -33.33 is: 33.33
```

In [47]:
```
## Bool method
test = []
print(test,'is',bool(test))
```

```
[] is False
```

In [48]:
```
test = [0]
print(test,'is',bool(test))
```

```
[0] is True
```

In [49]:
```
test = 0.0
print(test,'is',bool(test))
```

```
0.0 is False
```

In [50]:
```
test = None
print(test,'is',bool(test))
```

```
None is False
```

In [51]:
```
test = True
print(test,'is',bool(test))
```

```
True is True
```

In [52]:
```
test = 'Easy string'
print(test,'is',bool(test))
```

```
Easy string is True
```

In [55]:
```
## Dir method
number = [1, 2, 3]
print(dir(number))
print('\nReturn Value from empty dir()')
print(dir())
```

```
['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__do
c__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__gt__', '__hash__', '__iadd__',
'__imul__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__',
'__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmul__', '__setattr__', '__setitem_
_', '__sizeof__', '__str__', '__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'ins
ert', 'pop', 'remove', 'reverse', 'sort']

Return Value from empty dir()
['Animals', 'Animate', 'Giraffes', 'In', 'Inanimate', 'Mammals', 'MyClass', 'Out', 'Person', 'Sidewalks', 'T
hings', '_', '_4', '_41', '_42', '_44', '__', '___', '__builtin__', '__builtins__', '__doc__', '__loader__',
'__name__', '__package__', '__spec__', '_dh', '_i', '_i1', '_i10', '_i11', '_i12', '_i13', '_i14', '_i15',
'_i16', '_i17', '_i18', '_i19', '_i2', '_i20', '_i21', '_i22', '_i23', '_i24', '_i25', '_i26', '_i27', '_i2
8', '_i29', '_i3', '_i30', '_i31', '_i32', '_i33', '_i34', '_i35', '_i36', '_i37', '_i38', '_i39', '_i4', '_
i40', '_i41', '_i42', '_i43', '_i44', '_i45', '_i46', '_i47', '_i48', '_i49', '_i5', '_i50', '_i51', '_i52',
'_i53', '_i54', '_i55', '_i6', '_i7', '_i8', '_i9', '_ih', '_ii', '_iii', '_oh', 'add', 'age', 'another_vari
able', 'converted_age', 'exit', 'floating', 'get_ipython', 'integer', 'my_function', 'number', 'p1', 'quit',
'subtract', 'test', 'time', 'variable_test', 'variable_test2', 'x', 'y']
```

In [ ]: