In [1]:
```
#1 Explain the key features of Python that make it a popular choice for programming
```

```
* Python is a programming language.
* It is easy to understand and write.
* Widely used in industry.
* Lots of libraries.
* It is also used in Automation industry, frontend,backend and in analysing data.
```

In [2]:
```
#2 Describe the role of predefined keywords in Python and provide examples of how they are used in a progra
```

```
* Keywords are reserved keywords that have special meaning.
* Example- if,else,while,for,class,def,import,try,except,return,True,False,None etc.
* It cannot be used as an identifiers (i.e., variable names, function names).
* Keywords are case-sensitive.
```

In [3]:
```
#example of above predefined keyword
True and False
```

Out[3]: False

In [4]:
```
True or True
```

Out[4]: True

In [5]:
```
#3 Compare and contrast mutable and immutable objects in Python with examples
```

```
* In Python objects can be either changeable (modifiable) or unchangeable.
* Mutable objects- can be modified after creation.
 example - lists and dictionaries
* Immutable objects- cannot be changed after creation.
 example- tuples and strings
```

In [6]:
```
#example of mutable objects>>list
list= [13,21,'sanjay','pwskills']
list
```

Out[6]: [13, 21, 'sanjay', 'pwskills']

In [9]:
```
list[2]
```

Out[9]: 'sanjay'

In [11]:
```
list[2]='kumar'
list
```

Out[11]: [13, 21, 'kumar', 'pwskills']

In [12]:
```
#example of immutable objects>>string
str='sanjay'
str
```

Out[12]: 'sanjay'

In [14]:
```
str[3]
```

Out[14]: 'j'

In [16]:
```
str[3]=2
str
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-16-5a13b1f6739f> in <module>
----> 1 str[3]=2
      2 str

TypeError: 'str' object does not support item assignment
```

In [17]:
```
#4 Discuss the different types of operators in Python and provide examples of how they are used
```

```
* Operators are special symbols or keywords used to carry out specific actions on number of variables.
* Python supports various types of operators like-
 Arithmetic operator(+,-,*,/), Comparision operator(==,>=,<=), Logical operator(and,or,not),Assignment
operator(=,+=,-=,*=,/=)
* Operators have precedence and associativity rules that determine the order of evaluation in
expressions.
```

In [18]:
```
#examples of varrious operators are
#Assignment operator
a=3
b=-3
a-b
```

Out[18]: 6

In [19]:
```
#Logical operator
True or False
```

Out[19]: True

In [20]:
```
False and False
```

Out[20]: False

In [21]:
```
#Comparision operator
10>9
```

Out[21]: True

In [22]:
```
#Arithmetic operator
5+6-2
```

Out[22]: 9

In [23]:
```
#5 Explain the concept of type casting in Python with examples
```

```
* Type casting is also called as type conversion.
* It is the process of changing one data type to another in Python.
* Python provides inbuilt functions for type casting like
 int(), float(), str(), tuple(), list() etc
* Type casting is neccessary for performing arithmetic operations,data manipulations,input/output
operators in Python programs
```

In [24]:
```
#example of type casting
a='2'+3
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-24-b0c5abe477bc> in <module>
      1 #example of type casting
----> 2 a='2'+3

TypeError: can only concatenate str (not "int") to str
```

In [25]:
```
int('2')
```

Out[25]: 2

In [27]:
```
a= int('2')+3
a
```

Out[27]: 5

In [28]:
```
#6 How do conditional statements work in Python? Illustrate with examples
```

```
* Conditonals in Python are used to execute code based on the evaluationof one or more conditions.
* Python supports conditional statements such as
 if, elif(else if), else
* Conditionals can be nested to handle multiple conditions and control different branches of execution
based on the outcome of logical expression.
```

In [29]:
```
#example of conditionals statements
a=20
if a>50:
    print('This will not get executed')
elif a<50:
    print('I got it hurray')
else:
    print('The number is equal to 50')
```

```
I got it hurray
```

In [30]:
```
#7 Describe the different types of loops in Python and their use cases with examples
```

```
* In Python loops are employed to repeat a sequence of actions or code until a specific condition is
fulfilled.
* Two primary types of loops are
 for loops and while loops
* For loops are used for iterating over a sequence of elements.
* While loops are used for executing code until a specific condition becomes False.
```

In [31]:
```
#example of while loop
n=7
i=1
while i<n:
    print(i)
    i=i+1
```

```
1
2
3
4
5
6
```

In [32]:
```
for i in 'sanjay':
    print(i)
```

```
s
a
n
j
a
y
```

In [ ]:
```

```