Q1) Explain the key features of Python that make it a popular choice for programming. Ans: 1) Easy to Learn and Read : Python syntax is easy . Its code are almost like plain english. 2) InterprQeted Language: Python is an interpreted language, meaning code is executed line by line , hence debugging is easy. 3) Versatile and Powerful: Python can be used for a wide range of applications. 4) Large Standard Library: Provide pre-written code for common task. 5) Open Source: Python is free to use.
Q2) Describe the role of predefined keyword in python and provide examples of how they are used in a program. Ans: Keywords are predifined words that hold a special meaning and have purpose. example: if , elif , else ,True , False , None , Continue etc.

In [1]:
```
#example
x = 10
if x > 5:
    print("x is greater than 5")
elif x == 5:
    print("x is equal to 5")
else:
    print("x is less than 5")
```

```
x is greater than 5
```

Q3) compare and contrast mutable and immutable objects in python with example. Ans: Mutable objects are those whoes values can be changed after the creation. ex. List , Sets , Dictionry Immutable objects are those whose values can not be modified after creation

In [2]:
```
#example
my_list = [2,3.3,5,"Ajay","Vijay"]
my_list[2]
```

Out[2]:
```
5
```

In [3]:
```
my_list[2] = 100
my_list
```

Out[3]:
```
[2, 3.3, 100, 'Ajay', 'Vijay']
```

In [4]:
```
#immutable example
my_touple = (5,2,7)
my_touple
```

Out[4]:
```
(5, 2, 7)
```

Q4) Discuss the different types of operator in python and provide example of how they are used. Ans: Operators are special symbols that perform operations on variables and values. 1. Arithmetic Operators: Addition (+) , Subtraction (-) , Multiplication (*) , Division (/), Floor Division (//) , Modulus (%) , Exponentiation (**) 2. Comparison Operators :Equal to (==) , Not equal to (!=),Greater than (>) , Less than (<) , Greater than or equal to (>=) , Less than or equal to (<=) 3. Logical Operators: AND , OR , NOT 4. Assignment Operators: (=) , (+=) , 5. Bitwise Operators

In [5]:
```
a = 5
b = 2
a + b
```

Out[5]:
```
7
```

In [6]:
```
a / b
```

Out[6]:
```
2.5
```

In [7]:
```
a > b
```

Out[7]:
```
True
```

In [8]:
```
a == b
```

Out[8]:
```
False
```

In [9]:
```
 a > 2 and b >3
```

Out[9]:
```
False
```

In [10]:
```
a > 2 or b > 3
```

Out[10]:
```
True
```

In [11]:
```
c = a + b
```

In [12]:
```
c
```

Out[12]:
```
7
```

Q5) Explain the concept of type casting in python with example

Ans: The process of changing the data types of values/ object is known as type casting.

In [13]:
```
#ex
a ="5"
b = 2
```

```
a+b
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[13], line 4
      2 a ="5"
      3 b = 2
----> 4 a+b

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

In [14]:
```
#type casting
a = "5"
b = 2
int(a)+b
```

Out[14]:
```
7
```

Q6) How do conditional statement work in python? Illustrate with examples. Ans: Conditional statements in Python allows to execute different blocks of code based on whether a condition is true or false.

In [15]:
```
#example
name = input(" Enter your name: ")
email = input("Enter your email: ")
password = input(" Enter Password: ")
if name == "":
    print("Please enter valid name")
else:
    if "@" not in email:
        print("Please enter valid email")
    else:
        if len(password)<6:
            print("Please enter valid password")
        else:
            print("Registration is successfull!")
```

```
 Enter your name: Chetana
Enter your email: chetana@gmail.com
 Enter Password: 123465
Registration is successfull!
```

Q7) Describe the different types of loops in python and their use cases with example. Ans: In Python, loops are used to execute a block of code repeatedly based on a condition. Python supports three main types of loops: for Loop while Loop nested Loop 1. for Loop The for loop is used to iterate over a sequence (like a list, tuple, string, or range). It is commonly used when you know in advance how many times you want to execute a block of code. 2. While loop The while loop repeatedly executes a block of code as long as a given condition is true. It is used when the number of iterations is not known in advance and is determined by some condition 3.Nested loops are loops inside other loops. Both for and while loops can be nested inside each other.

In [16]:
```
#example of for loop
fruits = ["Apple","Mango","Banana","Orange"]
for fruit in fruits:
    print(fruit)
```

```
Apple
Mango
Banana
Orange
```

In [19]:
```
#example of while loop
count = 0
while count < 5:
    print(count)
    count += 1
```

```
0
1
2
3
4
```

In [20]:
```
# Nested loop
rows = 4
columns = 6

# Outer loop for rows
for i in range(rows):
    # Inner loop for columns
    for j in range(columns):
        print("*", end=" ")  # Print a star followed by a space
    print()  # Newline after each row
```

```
* * * * * *
* * * * * *
* * * * * *
* * * * * *
```

In [ ]:
```

```