1. **Explain the key features of Python that make it a popular choice for programming.**

a. It is widely used in industry.
b. It is widely used in data industry. For example, Data analytics, data scientists.
c. There are a lot of libraries where optimized and well written codes already exist. For example, Pandas, numpi and c-bond etc. There are 137000 libraries in python.
d. Python community is very active and keeps growing.
e. Very easy to use.
f. Versatile (Machine learning, Web development, Backed and front end development.
g. It can connect with other databases or other platforms as well.

2. **Describe the role of predefined keywords in Python and provide examples of how they are used in a program.**

These are the keywords or we can say code of instructions which are already present in the python language to give different type of outputs as per requirements.

For example, Print, type etc.

We can check the python keywords by using below command as shown in pic

```
help('keywords')
```

```
Here is a list of the Python keywords.  Enter any keyword to get more help.

False               class               from                or
None                continue            global              pass
True                def                 if                  raise
and                 del                 import              return
as                  elif                in                  try
assert              else                is                  while
async               except              lambda              with
await               finally             nonlocal            yield
break               for                 not
```

3. **Compare and contrast mutable and immutable objects in Python with examples.**

The word mutable itself indicates to change something.

In python we can change the elements present in the memory blocks. The elements whose elements we can change are known as ad mutable objects. For example, list.

The objects whose elements we can not change are known and immutable objects.

For example, string.

For mutable objects

```
employee = ["bharat", "ankush", "anuj", "suman"]

"""
like in above list ther are 4 objects and we can access their memory block by count form left to right (0,1,2,3)
For right to left we can count in negative sequences (-1,-2,-3,-4)
"""
# Now for example we want to access anuj in the list we can follow below syntax

employee[2] or employee[-3]
```

```
'anuj'
```

Now suppose we want to change “anuj” to “Rahul” then –

```
employee = ["bharat", "ankush", "anuj", "suman"]

"""
like in above list ther are 4 objects and we can access their memory block by count form left to right (0,1,2,3)
For right to left we can count in negative sequences (-1,-2,-3,-4)
"""
# Now for example we want to access anuj in the list we can follow below syntax

employee[2] = "Rohit"
```

```
employee
```

```
['bharat', 'ankush', 'Rohit', 'suman']
```

As we know that strings are immutable objects we can seen in below example

```
#We will see here that we cant modify string as it is immutable object.

name = "bharat" # here also we can access eatch element in the memory block lik ledt to right (0,1,2,3,4,5)

name[2]

name[2] = "h" # suppose we need to change a with h
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[37], line 7
      3 name = "bharat" # here also we can access eatch element in the memory block lik ledt to right (0,1,2,3,4,5)
      5 name[2]
----> 7 name[2] = "h"

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

4. **Discuss the different types of operators in Python and provide examples of how they are used.**

There are different types of operators we use in python language.

1. Arithmetic operators (+ - * / % **)

```
•[1]: a = 5
      b = 6
      print(a+b) # addition

      11

•[2]: a = 5
      b = 6
      print(a-b) #substraction

      -1

•[7]: 14 % 4
      # modulus operator which gives remander

 [7]: 2

•[8]: 14 / 2 # division

 [8]: 7.0
```

2. Comparison operators (== > < !* >= <=)

```
[9]: a = 5
     b = 7
     a == b

[9]: False

[10]: a >= b

[10]: False

[11]: a <= b

[11]: True
```

3. Logical operators (AND OR NOT)

```
[13]: a = 1
      b = 0

      a and b

[13]: 0

[14]: a or b

[14]: 1

[17]: not a

[17]: False

[18]: not b

[18]: True

[ ]:
```

4. Bitwise operators(AND& OR| XOR^ Left shift<< Right shift>>)

5. Membership operators(in not in)

```
[ ] #membership operator

    a = "PWSKILLS"

    "P" in a
```

```
True
```

```
[ ] "am" in "I am Ajay"
```

```
True
```

```
[ ] "data" in "PWSKILLS"
```

```
False
```

6. Identity operators(is is not)

```
[ ] #identity operator >> compare the memory location of two object
    a = 2
```

```
[ ] b = 3
```

```
[ ] a is b
```

```
False
```

```
[ ] b is a
```

```
False
```

```
[ ] a = "Pw Skills"
    b=a
```

```
[ ] b is a
```

```
True
```

7. Assignment operators (= += -= *= /=)

```
[ ] #Assignment operator

    a  =10
```

```
    a
    10
```

```
[ ] a + 5
    15
```

```
[ ] a
    10
```

```
[ ] a = a + 10
```

```
    a
    20
```

5. **Explain the concept of type casting in Python with examples.**

   It means that we can change the data type.

   For example, string to integer or string to float etc.

```
#string to integer
a = "2"
print(type(a))
print(type(int(a)))
```

```
<class 'str'>
<class 'int'>
```

```
[ ] #float to integer
    a = 3.4
    type(a)
    float
```

6. How do conditional statements work in Python? Illustrate with examples

Conditional statements are of 4 types

1. If
2. Else
3. Else if (elif)
4. Nested if else

```
[3]: username = (input("enter your username"))
     email = (input("enter your email id"))
     password = (input("enter your password"))
     if (username == ""):
         print("enter valid username")
     else:
         if ("@" not in email):
             print("enter a valid email")
         else:
             if len(password) < 2:
                 print("enter a valid vassword")
             else:
                 print("login succesfull")
```

```
enter your username Bharat
enter your email id bharat@gmail.com
enter your password 7845
login succesfull
```

7. Describe the different types of loops in Python and their use cases with examples.

There are two types of loop in python (while, for). We can also use different statements like control, break and continue to manage the loops accordingly. While loop with break

```
count = 19
while (count > 0):
    print(count)
    count = count -1
    if (count ==10):
        break
else:
    print("while loop executed succesfuly")
```

```
19
18
17
16
15
14
13
12
11
```

While loop with continue statement

```
f = 18
n = 1
while ( f > n):
     n = n+1
     if (n == 4):
           continue
     print(n)
```

```
2
3
5
6
7
8
9
10
11
12
13
14
15
16
17
18
```

For with break statement

```
]: a = [1, 2, 3,  4, "bharat"]
for b in a:
    print(b)
    if (b == 3):
        break
else:
    print("for loop executed successfully")
```

```
1
2
3
```

For with continue statement

```python
a = [1, 2, 3, "surjit", 4, "bharat"]
for s in a:
    if s =="surjit":
        continue
    print(s)


else:
    print("for loop executed successfully")
```

```
1
2
3
4
bharat
for loop executed successfully
```