Q.1.Explain the key features of Python that makes it a popular choice for programming

- Easy to read
- Easy to Understand
- Large number of Libraries
- Scalability
- versatility
- Portability
- Open Source
- Community Above features makes python different from the other programming languages

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

Predefined key words are those which have a specific meaning for the compiler and hence have a fixed usage.

```
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
```

```
help("def")
help("if")
help("true")
help("while")
```

```
                    ["else" ":" suite]

    This repeatedly tests the expression and, if it is true, executes the
    first suite; if the expression is false (which may be the first time
    it is tested) the suite of the "else" clause, if present, is executed
    and the loop terminates.

    A "break" statement executed in the first suite terminates the loop
    without executing the "else" clause's suite.  A "continue" statement
    executed in the first suite skips the rest of the suite and goes back
    to testing the expression.

    Related help topics: break, continue, if, TRUTHVALUE
```

Q.3.Compare and contrast mutable and immutable objects with examples In Python, objects are categorized into two main types based on their mutability: mutable and

In Python, objects are categorized into two main types based on their mutability: **mutable and immutable objects.**

Mutable Objects Mutable objects are those that can be modified after their creation. This means that the content of the object can change, but the object itself remains the same in terms of its identity. eg. Lists,Dictionaries,etc.

Immutable Objects Immutable objects, on the other hand, cannot be modified after their creation. eg. Strings,Integers,etc.

```
l = ["baingan", "aaloo", 3, 100, 2+4j, True, 100.0001]
print(l)
l[0]
l[0] = "b"
print(l)
type(l) #mutable
```

```
['baingan', 'aaloo', 3, 100, (2+4j), True, 100.0001]
['b', 'aaloo', 3, 100, (2+4j), True, 100.0001]
list
```

```
a = "pwskills"
a[0]
a[0] = "s" #immutable
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-14-0ddbe0ac7c9a> in <cell line: 3>()
      1 a = "pwskills"
      2 a[0]
----> 3 a[0] = "s"

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

Next steps: Explain error

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

There are Seven types of operators:

```
1.Arthematic

 Arithmetic operators are used to perform basic mathematical operations.

2.Comparision

 Comparison operators are used to compare two values and return a boolean result.

3.Logical
   AND,OR,NOT
4.Bitwise
   same as logical but in o ans 1s

5.Assignment
   Assigning arthematic operators
```

5. Explain the the comncept of type casting in python with examples

Type casting in Python refers to the conversion of one data type into another. This is useful when we need to perform operations that require specific data types, or when we want to ensure consistency in data types. Type casting can be done in two ways: implicit and explicit.

IMPLICIT: Implicit type casting, also known as coercion, is automatically performed by Python when an operation involves mixed data types. Python converts one data type to another type without human interference

Explicit Explicit type casting, also known as type conversion, is done manually by the programmer using predefined functions. This type of casting is necessary when we need to convert between incompatible type

```
K = 99
# Integer
N = 99.99 # Float
 # Addition of integer and float
result = K + N
print(type(K))
print(type(N))
print(result)
print(type(result))
```

```
<class 'int'>
<class 'float'>
198.99
<class 'float'>
```

```
M="2"
print(type(M))
N=int(M)
print(type(N))
print(N)
```

```
<class 'str'>
<class 'int'>
2
```

6.How do conditional statements work in Python. Illustrate with examples. conditional statements are: if condition if else if elif else nested if else

```
a = 200
if a > 100: #to execute the if block of code if the condition is true
    print("a is greater than 100.")
```

```
a is greater than 100.
```

```
#if-else

is_ds_course = False
if is_ds_course:
    print("I am studing in pwskills and watching the lecture")
else:
    print("I will watch a movie")
```

```
I will watch a movie
```

```
#if -elif -else
a = 100
if a > 100:
    print("no is greater than 100")
elif a < 100:
    print("The no is less than 100")
else:
    print("equals to 100")
```

```
equals to 100
```

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

loops allows us to repeatedly execute a block of code loops are of teo types: for loop and while loop

while loops : repeatedly executes a block of code as long as condition is True

for loops : iterate over a sequence of elements

```

```

```
n = 7
i = 1
while i<n:
    print(i)
    i = i+1
else:
    print("This will be executed when the while loop is run succeessfully without any break")
```

```
1
2
3
4
5
6
This will be executed when the while loop is run succeessfully without any break
```

```
l = [1, 2, 3, 4, "MANI", "SHANKAR"]
for i in l:
    print(i)
else:
    print("This will executed when loop ends without a break")
```

```
1
2
3
4
MANI
SHANKAR
This will executed when loop ends without a break
```

```
Start coding or generate with AI.
```