

# 1) EXPLAIN KEY FEATURES OF PYTHON THAT MAKES IT POPULAR CHOICE FOR PROGRAMMING

**SIMPLICITY** python is easy to learn and use with a clear syntax that resembles english its also an object oriented language versatility in python is general purpose language that can be used for widely variety application other features of python are open source, standard library and database support

#2) describe the role of predefined keywords in python and provide examples and how they are used  
**ANSWER--** in python predefined keywords are reserved words that defines the syntax and structure of language  
 #examples

```
#print keyword
print("hello world")
```

```
hello world
```

```
#example
print("pw skills")
print("ajay gupta")
```

```
pw skills
ajay gupta
```

#3) compare and contrast mutable and immutable objects in python with example

**ANSWER-** in python the main difference between mutable and immutable objects is that mutable objects can be changed after they are created while immutable object cannot, mutable objects can be modified while immutable object cannot. use mutable object when you need to change the object state or content and use immutable objects for values that should not be modified

#EXAMPLES OF MUTABLE OBJECTS IN PYTHON

```
my_list=[1,2,3]
my_list.append(4)
print(my_list)
#here i understood that i can modify or change my list container as per my choice it shows that it is mutable
```

```
[1, 2, 3, 4]
```

#EXAMPLE OF IMMUTABLE OBJECT IN PYTHON

```
my_tuple=(1,2,3)
my_tuple[0]=4
print(my_tuple)
#here this code gave an error as the object cannot be modified
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-4-6cc72d1021bf> in <cell line: 3>()
      1 #EXAMPLE OF IMMUTABLE OBJECT IN PYTHON
      2 my_tuple=(1,2,3)
----> 3 my_tuple[0]=4
      4 print(my_tuple)
```

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

Next steps: [Explain error](#)

 **Generate** create a dataframe with 2 columns and 10 rows



[Close](#)

#4) DISCUSS DIFFERENT TYPES OF OPERATORS IN PYTHON AND THEIR USES WITH EXAMPLES

Double-click (or enter) to edit

**ANSWER-** there are several types of operators python the purpose of operators in python is to perform operations on variables and other different values operators can be symbol or keywords or a combination of symbols used for performing variety of tasks below are some examples of operators

```
#example of arithmetic operator
a=5
b=6
print(a+b)
```

 11

```
#example of assignment operators
x=5
x
```

 5

```
#example of logical operator
a=True
b=False
print(a and b)
```

 False

```
print(a or b)
```

 True

```
#example of identity operator
a=[1,2,3]
b=[1,2,3]
print(a is b)
```

 False

```
#example of bitwise operator
a=5
b=3
result=a|b
```

```
result=a&b
```

```
#5)EXPLAIN THE CONCEPT OF TYPE CASTING WITH EXAMPLES IN PYTHON
```

type casting in python refers to the conversation of one data type to another this can be useful when you need to perform operation on different types of data or when you want to ensure that data is in expected format there are two types of casting 1)implicit type casting,explicit type casting


```
#example of type casting implicit
a=5
b=2.0
result=a+b#implicit conversation of a to float
print(result)
```

 7.0

```
#example of explicit and implicit casting
a=10
b=2.5
result=a+b# implicit converts a to float
print(result)
#explicit type casting
string_number="100"
integer_number=int(string_number)
print(integer_number)
```

 12.5  
100

```
#HOW CONDITIONAL STATEMENT WORK IN PYTHON ILLUSTRATE WITH EXAMPLES
#if condition,elif condition,else:
age=18
if age>=18:
    print("you are eligible to vote")
else:
    print("you are not eligible to vote")
```

 you are eligible to vote

```
#conditional statement in python allow you to execute certain blocks of code based on wether a condition is true or false the primary c
```

#6)describe different types of loops in python and their used cases with examples

python provides several types of loops to execute a block of code repeatedly the two primary types are for loops and while loops here an overview of each along with common use cases

```
#example of loops
#for loop
fruits=["apple","banana","cherry"]
for fruit in fruits:
    print(fruit)
```

```
↵ apple
    banana
    cherry
```

```
for i in range(5):
    print(i)
```

```
↵ 0
   1
   2
   3
   4
```

Start coding or [generate](#) with AI.

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.