**PYTHON**

Here are **common Python interview questions** focused on **syntax, variables, data types, casting, strings, and booleans**, along with concise answers:

**1. What’s the correct way to write a comment in Python?**

*Answer:*

* *Single-line comment:*

*# This is a comment*

* *Multi-line (docstring style, though mainly for docs):*

*"""*

*This is a*

*multi-line comment*

*"""*

**2. How do you declare and initialize variables? Any naming rules?**

***Answer:***

*name = "Alice" # no declaration keyword*

*age = 30*

*pi = 3.14*

***Rules:***

* *Must start with a letter or underscore (\_), not a digit.*
* *Case-sensitive (age ≠ Age).*
* *No spaces; use underscores (first\_name).*

**3. What are Python’s built-in primitive data types?**

***Answer:***

* ***Numeric:*** *int, float, complex*
* ***Text:*** *str*
* ***Boolean:*** *bool (True or False)*
* ***NoneType:*** *None (the absence of a value)*

**4. How do you check a variable’s data type at runtime?**

***Answer:***

*x = 42*

*print(type(x)) # <class 'int'>*

*print(isinstance(x, int)) # True*

**5. What’s type casting/coercion? Give examples.**

***Answer:*** *Converting one data type to another using built-ins:*

*s = "123"*

*n = int(s) # '123' → 123*

*f = float("3.14") # → 3.14*

*b = bool(0) # → False (0 is Falsey)*

*txt = str(99) # → "99"*

**6. How does Python evaluate truthiness of values?**

***Answer:***

* ***Falsey*** *values: 0, 0.0, "" (empty string), None, False, empty collections*
* ***Truthy****: almost everything else*

*if "": print("won't print")*

*if 1: print("prints") # because 1 is truthy*

**7. How do you index and slice strings?**

***Answer:***

*s = "Hello, World!"*

*print(s[0]) # 'H'*

*print(s[-1]) # '!'*

*print(s[7:12]) # 'World'*

*print(s[:5]) # 'Hello'*

*print(s[::2]) # 'Hlo ol!'*

**8. How can you embed variables inside strings?**

**Answer:**

* ***f-strings*** *(Python 3.6+):*
* *name = "Alice"*
* *greeting = f"Hello, {name}!" # "Hello, Alice!"*
* ***str.format()****:*
* *"Hello, {}!".format(name)*

**9. What’s the difference between == and is?**

***Answer:***

* *== checks* ***value equality****.*
* *is checks* ***identity*** *(whether both refer to the same object in memory).*

*a = [1,2,3]*

*b = [1,2,3]*

*print(a == b) # True*

*print(a is b) # False*

**10. How do you convert a string containing both letters and numbers to an integer safely?**

**Answer:**  
*Use exception handling to catch invalid casts:*

*s = "123abc"*

*try:*

*n = int(s)*

*except ValueError:*

*print("Cannot convert to int")*

**11. What Boolean operators does Python support?**

***Answer:***

* ***Logical:*** *and, or, not*
* ***Comparison:*** *<, >, <=, >=, ==, !=*

*x, y = 5, 10*

*print(x < y and y < 20) # True*

*print(not (x == y)) # True*

**12. Are strings in Python mutable or immutable? What does that imply?**

***Answer:***

* ***Immutable****: once created, you cannot change their contents.*
* *Any “modification” creates a* ***new string****:*
* *s = "Hello"*
* *s = s + " World" # creates new string; original 'Hello' unchanged*

**Basic Problems**

* 1. **Variables and Operators**

**Q1. Swap two numbers without using a third variable.**

a = 5

b = 10

a, b = b, a

print("a =", a, "b =", b)

**Q2. Find the result of a complex expression:**

x = 10

y = 5

result = x\*\*2 + y\*3 - x//y

print(result) # 100 + 15 - 2 = 113

**2. Data Types and Type Casting**

**Q1. Convert a float to int and vice versa.**

a = 5.7

print(int(a)) # 5

b = 7

print(float(b)) # 7.0

**Q2. Check the type of multiple variables.**

a = 10

b = "hello"

c = [1, 2, 3]

print(type(a), type(b), type(c))

1. **Booleans and Conditionals**

**Q1. Write a program to check if a number is even or odd.**

num = 7

if num % 2 == 0:

print("Even")

else:

print("Odd")

**Q2. Check if a number is positive, negative or zero.**

n = -5

if n > 0:

print("Positive")

elif n < 0:

print("Negative")

else:

print("Zero")

1. **Strings**

**Q1. Count vowels in a string.**

s = "hello world"

vowels = 'aeiou'

count = sum(1 for char in s if char in vowels)

print("Vowels:", count)

**Q2. Reverse a string using slicing.**

s = "python"

print(s[::-1]) # nohtyp

1. **Functions**

**Q1. Write a function to find factorial.**

def factorial(n):

result = 1

for i in range(1, n+1):

result \*= i

return result

print(factorial(5)) # 120

**Q2. Create a function to check if a number is prime.**

def is\_prime(n):

if n <= 1:

return False

for i in range(2, int(n\*\*0.5)+1):

if n % i == 0:

return False

return True

print(is\_prime(7)) # True

1. **Loops (for, while)**

**Q1. Print Fibonacci series up to n terms.**

n = 7

a, b = 0, 1

for \_ in range(n):

print(a, end=" ")

a, b = b, a + b

**Q2. Find the sum of digits of a number using while loop.**

num = 1234

sum\_digits = 0

while num > 0:

sum\_digits += num % 10

num //= 10

print(sum\_digits) # 10

1. **Collections (List, Tuple, Set, Dictionary)**

**Q1. Remove duplicates from a list.**

lst = [1, 2, 2, 3, 4, 4, 5]

unique = list(set(lst))

print(unique)

**Q2. Count frequency of elements in a list using dictionary.**

lst = ['a', 'b', 'a', 'c', 'b']

freq = {}

for item in lst:

freq[item] = freq.get(item, 0) + 1

print(freq) # {'a': 2, 'b': 2, 'c': 1}

1. **Lambda Functions**

**Q1. Sort list of tuples by second element using lambda.**

pairs = [(1, 3), (2, 2), (4, 1)]

pairs.sort(key=lambda x: x[1])

print(pairs) # [(4, 1), (2, 2), (1, 3)]

**Q2. Double all numbers in a list using map and lambda.**

nums = [1, 2, 3, 4]

doubled = list(map(lambda x: x \* 2, nums))

print(doubled)