📘 20 Python Basics Interview Questions

Author: PyShaala
Date: August 14, 2025
YouTube Channel: PyShaala

Welcome to this special notebook covering **top 20 Python Basics interview questions**.

🟢Q1. What is Python and why is it popular?

Explanation: Python is a high-level, interpreted programming language known for its readability and simplicity. It's popular due to its versatility (web development, data science, automation, etc.), large community, and extensive libraries.

💡 **Interview Hack:** When asked about Python's popularity, mention its key features like readability, versatility, and large ecosystem.

🟢Q2. What are some key features of Python?

Explanation: Key features include:

- Easy to learn and having syntax like english language.
- Dynamically typed.
- Interpreted language.
- Large standard library.
- Supports multiple programming paradigms (procedural, object-oriented, functional).
- Platform Independent.

💡 **Interview Hack:** Focus on readability and the extensive libraries as major selling points.

🟢Q3. Is Python a compiled or interpreted language? Explain.

Explanation: Python is typically considered an interpreted language. This means the code is executed line by line by an interpreter without a separate compilation step into machine code beforehand. However, Python does compile source code into bytecode (`.pyc` files), which is then executed by the Python Virtual Machine (PVM).

🟢Q4. Explain the difference between interpreted and compiled languages, and where Python fits in.

Explanation: Interpreted languages execute code line by line using an interpreter, while compiled languages are translated into machine code before execution.

Python is generally considered an interpreted language, although it does involve a compilation step to bytecode.

🟢Q5. What is PEP 8? Why is it important?

Explanation: PEP 8 is the official Python Enhancement Proposal that provides style guidelines for Python code. It covers formatting conventions (indentation, line length, naming conventions, etc.) to make code more readable and consistent across projects and developers. It's important because consistent code is easier to understand, maintain, and collaborate on.

💡 **Interview Hack:** Mention readability and consistency as the main benefits of following PEP 8.

In [ ]:

```python
# Example of PEP 8 compliant code (snake_case for variables)
my_variable_name = 10

print("PEP 8 promotes readable code.")
```

```
PEP 8 promotes readable code.
```

🟢 Q6. What are Python's data types?

Common ones: int, float, str, bool, list, tuple, set, dict.

💡 **Interview Hack:** List them quickly to show familiarity.

In [2]:

```
data = [42, 3.14, "hi", True, [1,2], (3,4), {5,6}, {"a":1}]
for x in data:
    print(f'{type(x)}: {x}')
```

```
<class 'int'>: 42
<class 'float'>: 3.14
<class 'str'>: hi
<class 'bool'>: True
<class 'list'>: [1, 2]
<class 'tuple'>: (3, 4)
<class 'set'>: {5, 6}
<class 'dict'>: {'a': 1}
```

🟢 Q7. What is dynamic typing in Python?

You don't declare variable types; Python infers them at runtime.

💡 **Interview Hack:** Emphasize flexibility but also note potential runtime errors.

In [3]:

```
x = 5
print(type(x))
x = "hello"
print(type(x))
```

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

🟢 Q8. How do you write comments in Python?

- Single-line: `#`
- Multi-line: triple quotes (but not officially a comment)

💡 **Interview Hack:** Mention docstrings for documentation.

In [4]:

```
# This is a comment
""" This is
a multi-line docstring """
```

Out[4]:

```
' This is \na multi-line docstring '
```

🟢 Q9. How do you take user input in Python?

Use `input()` which returns a string.

💡 **Interview Hack:** Always show casting to int/float if needed.

In [5]:

```
name = input("Enter name: ")
print(name)
```

```
Pyshaala
```

🟢 Q10. How do you check the type of a variable?

Use `type()`.

💡 **Interview Hack:** Also mention `isinstance()` for inheritance-aware checks.

In [6]:
```
print(type(42))
print(isinstance(42.02, int))
```

```
<class 'int'>
False
```

🟢 Q11. What is indentation in Python and why is it important?

Indentation defines code blocks in Python instead of braces `{}`.

💡 **Interview Hack:** Mention 4 spaces as the convention.

In [7]:
```
if False:
    print("Indented!")
    print('second line')
print('third line')
```

```
third line
```

🟢 Q12. How do you declare and use variables in Python?

Just assign with `=`.

💡 **Interview Hack:** Mention multiple assignment.

In [8]:
```
x = 7

a, b = 5, 10
print(a, b)  # Output: 5 10
```

```
5 10
```

🟢 Q13. Explain mutable vs immutable types in Python.

- Mutable: list, dict, set
- Immutable: int, float, str, tuple

💡 **Interview Hack:** Give quick examples of both.

In [9]:

```
x = [1,2]
x.append(3)
print(x)
my_str = 'Pyshaala'
my_str[8] = 'P'
```

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

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[9], line 5
      3 print(x)
      4 my_str = 'Pyshaala'
----> 5 my_str[8] = 'P'

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

🟢 Q14. How do you convert between data types?

Use casting functions: `int()`, `float()`, `str()`.

💡 **Interview Hack:** Show list-to-set conversion.

In [ ]:

```
print(int(3.7))
print(set([1,1,2]))
```

🟢 Q15. What are Python keywords?

Reserved words with special meaning.

💡 **Interview Hack:** Show quick check using `keyword` module.

In [ ]:

```
import keyword

print(keyword.kwlist)
print(len(keyword.kwlist))
print(keyword.iskeyword("for"))  # Output: True
```

🟢 Q16. What is `None` in Python?

`None` is a special constant representing absence of value.

💡 **Interview Hack:** It's not `0` or empty string.

In [10]:

```
x = None
print(x is None)
```

```
True
```

🟢 Q17. Explain `id()` function.

Returns unique identifier (memory address) of an object.

💡 **Interview Hack:** Good for checking mutability behavior.

In [11]:

```
a = 8
b = 5
print(id(a) == id(b))
print(id(a))
print(id(b))
```

```
False
140705789236232
140705789236136
```

🟢 Q18. What is the difference between `=` and `==`?

- `=` assigns a value.
- `==` checks equality.

💡 **Interview Hack:** Show both in code to avoid confusion.

In [ ]:

```
a = 5
print(a == 5)
```

🟢 Q19. What is pass in Python?

The `pass` keyword represents a null operation in Python.

It is generally used for the purpose of filling up empty blocks of code which may execute during runtime but has yet to be written.

In [ ]:

```
if a == None:
    pass
```

🟢 Q20. What is docstring in Python?

- Documentation string or docstring is a multiline string used to document a specific code segment.

- The docstring should describe what the function or method does

In [ ]:

```
def my_empty_func():
   '''Empty Function for showing doc
   string and pass example'''
   # do nothing
   pass
# print(my_empty_func.__doc__)
help(my_empty_func)
```