**Python Question & Answers**

**### Python Interview Questions and Answers**

**\*\*1. How do you open a file for reading and writing in Python?\*\***

Use the `open()` function with the desired mode.

Example:

```python

file = open('data.txt', 'r+')  # Opens for reading and writing

```

**\*\*2. What is the difference between `"r"`, `"w"`, `"r+"`, and `"x+"` modes in file operations?\*\***

- `"r"`: Read-only, file must exist.

- `"w"`: Write-only, creates file or truncates if exists.

- `"r+"`: Read and write, file must exist.

- `"x+"`: Create and open for reading/writing, fails if file exists.

**\*\*3. How can you handle the case when a file does not exist while opening it?\*\***

Use a `try-except` block:

```python

try:

    file = open('data.txt', 'r')

except FileNotFoundError:

    print("File not found.")

```

**\*\*4. How do you write a new line to a file and then read the updated content?\*\***

```python

with open('data.txt', 'a+') as file:

    file.write('\nNew line')

    file.seek(0)

    print(file.read())

```

**\*\*5. What is the purpose of `file.seek(0)` in file handling?\*\***

It moves the file pointer to the beginning of the file, allowing you to read from the start.

**\*\*6. What are first-class functions in Python?\*\***

Functions are treated as objects; they can be passed as arguments, returned from other functions, and assigned to variables.

**\*\*7. How can you pass a function as an argument to another function?\*\***

```python

def greet(name):

    return f"Hello, {name}"

def call\_func(func, value):

    return func(value)

print(call\_func(greet, "Alice"))

```

**\*\*8. How do you define and call a function with multiple arguments?\*\***

```python

def add(a, b):

    return a + b

result = add(2, 3)

```

**\*\*9. What is a nested function? Give an example.\*\***

A function defined inside another function.

```python

def outer():

    def inner():

        print("Inner function")

    inner()

```

**\*\*10. What is the difference between a `for` loop and a `while` loop in Python?\*\***

- `for` loop: Iterates over a sequence (like list, string).

- `while` loop: Repeats as long as a condition is true.

**\*\*11. How do you avoid infinite loops in a `while` loop?\*\***

Ensure the loop condition will eventually become false.

**\*\*12. How does the `break` statement work in loops?\*\***

It immediately exits the loop.

**\*\*13. How do you handle exceptions in Python?\*\***

Using `try-except` blocks.

**\*\*14. What is the use of `try-except` blocks?\*\***

To catch and handle errors without stopping the program.

**\*\*15. How do you catch specific exceptions like `FileNotFoundError`?\*\***

```python

try:

    open('file.txt')

except FileNotFoundError:

    print("File not found.")

```

**\*\*16. How can you use a dictionary to manage student names and grades?\*\***

```python

grades = {'Alice': 90, 'Bob': 85}

```

**\*\*17. How do you update a value in a dictionary?\*\***

```python

grades['Alice'] = 95

```

**\*\*18. How do you iterate over all items in a dictionary?\*\***

```python

for name, grade in grades.items():

    print(name, grade)

```

**\*\*19. How do you import a module from a subdirectory in Python?\*\***

Use `from subdir import module` if there’s an `\_\_init\_\_.py` file.

**\*\*20. What is the purpose of the `\_\_pycache\_\_` folder?\*\***

It stores compiled bytecode of Python files for faster loading.

**\*\*21. How do you take user input and use it in your program?\*\***

```python

name = input("Enter your name: ")

print("Hello,", name)

```

**\*\*22. How do you print formatted strings in Python?\*\***

```python

name = "Alice"

print(f"Hello, {name}")

```

**\*\*23. What is the difference between `read()`, `readline()`, and `readlines()` methods for files?\*\***

- `read()`: Reads entire file as a string.

- `readline()`: Reads one line at a time.

- `readlines()`: Reads all lines into a list.