



Python Theoretical Assessment

1. What is the difference between a tuple and a list in Python?
 - a) Tuples are mutable and lists are immutable
 - b) Lists are mutable and tuples are immutable
 - c) Tuples are ordered and lists are unordered
 - d) Lists are ordered and tuples are unordered
2. How would you create a dictionary with keys and values in Python?
 - a) `dict = {1: "apple", 2: "banana", 3: "cherry"}`
 - b) `dict = {"apple": 1, "banana": 2, "cherry": 3}`
 - c) `dict = (1: "apple", 2: "banana", 3: "cherry")`
 - d) `dict = ["apple": 1, "banana": 2, "cherry": 3]`
3. What is the purpose of the "else" statement in a for loop?
 - a) To execute code after the for loop has finished executing
 - b) To exit the for loop if a certain condition is met
 - c) To execute code if the for loop does not encounter a "break" statement
 - d) To execute code if the for loop encounters a "continue" statement
4. How do you check the type of a variable in Python?
 - a) `type(x)`
 - b) `x.type()`
 - c) `x.getType()`
 - d) `Type(x)`
5. How do you check if a key exists in a dictionary in Python?
 - a) `key in dict`
 - b) `dict.contains(key)`
 - c) `dict[key]`
 - d) `dict.has_key(key)`



6. What is the purpose of the "try" and "except" statements in Python?
- a) To catch and handle errors that occur in the "try" block
 - b) To run specific code if an error occurs in the "try" block
 - c) To exit a loop if an error occurs in the "try" block
 - d) To raise an error if a specific condition is met in the "try" block
7. What is the correct syntax for a while loop in Python?
- a) while x < 5:
 - b) while (x < 5):
 - c) while x ==< 5:
 - d) while x <= 5:
8. How do you check if a list is empty in Python?
- a) if len(list) == 0:
 - b) if list.empty():
 - c) if list == null:
 - d) if not list:
9. How do you remove duplicates from a list in Python?
- a) list(set(x))
 - b) x.remove_duplicates()
 - c) list.unique(x)
 - d) x = list(dict.fromkeys(x))
10. How do you round a floating point number to 2 decimal places in Python?
- a) round(x, 2)
 - b) x.round(2)
 - c) x.toFixed(2)
 - d) x.roundDecimal(2)



Python Code Assessment

1. What is the output of the following code?

```
x = [1, 2, 3]
y = x
y[1] = 4
print(x)
```

- a) [1, 2, 3]
- b) [1, 4, 3]
- c) [4, 2, 3]
- d) [1, 2, 4]

2. What is the result of the following code?

```
def my_function(x):
    return x * 2

print(my_function(3))
```

- a) 6
- b) 3
- c) 9
- d) None

3. What is the result of the following code?

```
x = "Hello"
y = "World"
z = x + " " + y
print(z)
```

- a) "Hello"
- b) "World"
- c) "Hello World"
- d) "World Hello"



4. What is the output of the following code?

```
x = [1, 2, 3, 4, 5]
print(x[-2])
```

- a) 1
- b) 2
- c) 4
- d) 5

5. What is the output of the following code?

```
x = {"a": 1, "b": 2, "c": 3}
x.pop("b")
print(x)
```

- a) {"a": 1, "b": 2, "c": 3}
- b) {"a": 1, "c": 3}
- c) {"b": 2, "c": 3}
- d) {"a": 1, "b": 2}

6. What is the output of the following code?

```
x = [1, 2, 3, 4, 5]
y = x
y[1] = 9
print(x)
```

- a) [1, 2, 3, 4, 5]
- b) [1, 9, 3, 4, 5]
- c) [9, 2, 3, 4, 5]
- d) [1, 2, 9, 4, 5]

7. What is the output of the following code?

```
def fibonacci(n):
    if n <= 1:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)
print(fibonacci(6))
```



- a) 5
- b) 8
- c) 13
- d) 21

8. What is the output of the following code?

```
def reverse_string(s):  
    if len(s) == 0:  
        return s  
    else:  
        return reverse_string(s[1:]) + s[0]  
print(reverse_string("Hello"))
```

- a) "Hello"
- b) "olleH"
- c) "oHlle"
- d) "lleoH"

9. What is the output of the following code?

```
def find_largest(arr, n):  
    if n == 1:  
        return arr[0]  
    else:  
        return max(arr[n-1], find_largest(arr, n-1))  
arr = [1, 2, 3, 4, 5]  
print(find_largest(arr, len(arr)))
```

- a) 1
- b) 2
- c) 3
- d) 5



10. What is the output of the following code?

```
def binary_search(arr, l, r, x):  
    if r >= l:  
        mid = l + (r - l) // 2  
        if arr[mid] == x:  
            return mid  
        elif arr[mid] > x:  
            return binary_search(arr, l, mid-1, x)  
        else:  
            return binary_search(arr, mid+1, r, x)  
    else:  
        return -1  
arr = [1, 2, 3, 4, 5]  
x = 4  
result = binary_search(arr, 0, len(arr)-1, x)  
print(result)
```

- a) -1
- b) 0
- c) 1
- d) 3