Quiz 03-11: String, If-Statement, and For-Loop

1) String or not?

1. `a = "7"` → **string**
2. `b = 7` → **not a string**
3. `c = " "` → **string** (a string with one space)
4. `d = ""` → **string** (the empty string)

2) What does it print? (concatenation)

Output:

```
BlueBerry
Blue Berry
```

3) Index and slice practice

For `s = "pineapple"`:

1. `s[0]` is `p`
2. `s[4:9]` is `apple`
3. `s[:4]` is `pine`
4. `s[4:]` is `apple`

4) Fix the TypeError (string + number)

One correct fix:

```
print("Age: " + str(age))
```

5) List operations: what are the final values?

Step-by-step: - Start: `["cat", "dog"]` - Append `"fish"` → `["cat", "dog", "fish"]` - Change index 1 to `"hamster"` → `["cat", "hamster", "fish"]` - Pop index 0 removes `"cat"`

Answers: 1. `x` is: `"cat"` 2. `pets` is now: `["hamster", "fish"]`

6) Write code: swap first and last items (in-place)

One correct answer:

```
nums = [10, 20, 30, 40]

temp = nums[0]
nums[0] = nums[3]
nums[3] = temp

print(nums)  # [40, 20, 30, 10]
```

7) True or False (and / or / not)

1. `True`
2. `False`
3. `True`
4. `True`

8) Fill in the operator

1. `8 > 6` is `True`
2. `"hi" == "hi"` is `True`
3. `3 != 3` is `False`
4. `4 < 9` is `True`

9) What does it print?

Output:

```
A
Done
```

10) Write code: medal labels (if / elif / else)

One correct answer:

```
score = 82

if score >= 90:
    print("gold")
elif score >= 75:
    print("silver")
elif score >= 60:
    print("bronze")
else:
    print("try again")
```

For `score = 82`, it prints:

```
silver
```

---

11) Spot the mistake (`=` vs `==`)

Correct `if` line:

```
if n == 5:
```

Full fixed code:

```
n = 5
if n == 5:
    print("yes")
```

---

12) Write code: stay in or go out

One correct answer:

```
temp = 38
raining = False

if temp < 40 or raining == True:
    print("stay in")
else:
    print("go out")
```

With `temp = 38` and `raining = False`, it prints:

```
stay in
```

---

13) What numbers does `range` make?

1. `list(range(4))` → `[0, 1, 2, 3]`
2. `list(range(3, 12))` → `[3, 4, 5, 6, 7, 8, 9, 10, 11]`

3. `list(range(5, 26, 5))` → `[5, 10, 15, 20, 25]`

14) What does it print?

Output:

```
15
```

15) What does it print? (list traversal)

Output:

```
go!
stop!
go!
go!
```

16) What does it print? (`break`)

Output:

```
go
go
end
```

17) Write code: first even number (search + break)

One correct answer (uses a flag):

```
nums = [5, 9, 12, 7, 8]

found = False

for x in nums:
    if x % 2 == 0:
        print(x)
        found = True
        break

if found == False:
    print("no even")
```

Expected output:

```
12
```

18) Write code: build a list of squares (append + range)

One correct answer:

```
squares = []

for x in range(1, 6):
    squares.append(x * x)

print(squares)
```

Expected output:

```
[1, 4, 9, 16, 25]
```

19) Write code: count how many are > 10

One correct answer:

```
nums = [4, 12, 8, 20, 11]

count = 0
for x in nums:
    if x > 10:
        count = count + 1

print(count)
```

Expected output:

```
3
```

20) Last digit and “remove last digit”

For `n = 583` : 1. `last = 3` 2. `rest = 58`

21) What does it print? (peel digits)

Output:

```
5
0
4
```

22) Complete the code: sum of digits

Filled code:

```
n = 7316
s = 0

for _ in range(100):
    s = s + (n % 10)   # add last digit to s
    n = n // 10        # remove last digit from n
    if n == 0:         # stop when n == 0
        break

print(s)
```

Output:

```
17
```

23) Write code: print all factors of 24

One correct answer:

```
n = 24

for k in range(1, n + 1):
    if n % k == 0:
        print(k)
```

Output:

```
1
2
3
4
6
8
12
24
```