Quiz 06–11: Lists and For-Loops

1) Build a list (mixed types)

One correct answer:

```
items = [5, "hi", 0]
```

2) Index and change

```
nums = [10, 20, 30, 40]
nums[1] = nums[1] + 5
nums[3] = 99
print(nums)
```

Output:

```
[10, 25, 30, 99]
```

3) `append` and `pop`

```
letters = ["A", "B"]
letters.append("C")
x = letters.pop()
letters.append("D")
print(x)
print(letters)
```

Output:

```
C
['A', 'B', 'D']
```

4) Insert and remove

One correct answer:

```
colors = ["red", "green", "blue"]

colors.insert(1, "yellow")   # insert
colors.remove("blue")        # remove
print(colors)                # print
```

Output:

```
['red', 'yellow', 'green']
```

5) What does `range(stop)` produce?

```
nums = list(range(6))
print(nums)
```

Output:

```
[0, 1, 2, 3, 4, 5]
```

6) What does `range(start, stop)` produce?

```
nums = list(range(3, 9))
print(nums)
```

Output:

```
[3, 4, 5, 6, 7, 8]
```

7) Step forward

```
nums = list(range(2, 13, 3))
print(nums)
```

Output:

```
[2, 5, 8, 11]
```

8) Step backward (negative step)

```
nums = list(range(10, 2, -2))
print(nums)
```

Output:

```
[10, 8, 6, 4]
```

9) Count positives

One correct answer:

```
nums = [-2, 5, 0, 7, -1, 3]

count = 0

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

print(count)
```

Output:

```
3
```

10) Sum of even numbers in a list

One correct answer:

```
nums = [4, 1, 6, 9, 2, 8]

total = 0

for x in nums:
    if x % 2 == 0:
        total += x

print(total)
```

Output:

```
20
```

11) Build a new list (not in-place)

One correct answer:

```
words = ["a", "hi", "robot", "ok"]

long_words = []

for w in words:
    if len(w) >= 3:
        long_words.append(w)

print(long_words)
```

Output:

```
['robot']
```

12) Reverse a list (not in-place)

One correct answer:

```
nums = [3, 1, 4, 1, 5]

n = len(nums)
rev = []

for i in range(n - 1, -1, -1):
    rev.append(nums[i])

print(rev)
```

Output:

```
[5, 1, 4, 1, 3]
```

13) First multiple of 7

One correct answer:

```
nums = [5, 11, 13, 21, 8, 28]

for x in nums:
    if x % 7 == 0:
        print(x)
        break
```

Output (the first multiple of 7):

```
21
```

14) Search a range with `break`

One correct answer:

```
for x in range(1, 51):
    if x % 6 == 0 and x % 8 == 0:
        print(x)
        break
```

Output:

```
24
```

15) Stop when the running sum gets big

One correct answer:

```
nums = [2, 4, 6, 8, 10]

total = 0  # running sum

for x in nums:
    total = total + x
    if total >= 13:
        print(total)
        break
```

Output:

```
20
```

16) Digits: last digit and “remove last digit”

Let `n = 5089`:

- `n % 10` is **9**
- `n // 10` is **508**

17) Sum of digits

One correct answer:

```
n = 32046

total = 0

for _ in range(1000):
    x = n % 10        # get the last digit
    n = n // 10       # remove the last digit from n
    total = total + x # add digit to total

    if n == 0:
        break

print(total)
```

Output:

```
15
```

18) Count how many digits

One correct answer:

```
n = 900120

count = 0

for _ in range(1000):
    if n == 0:
        break

    n = n // 10          # remove the last digit from n
    count = count + 1    # increase the digit count

print(count)
```

Output:

```
6
```

19) Factors in a range

One correct answer:

```
n = 36

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

Output:

```
1
2
3
4
6
9
12
18
36
```

20) Prime check

One correct answer:

```
n = 29
is_prime = True

for d in range(2, n):
    if n % d == 0:
        is_prime = False
        break

if is_prime:
    print("prime")
else:
    print("not prime")
```

Output:

```
prime
```

21) All pairs (store in a list)

One correct answer:

```
letters = ["A", "B"]
nums = [1, 2, 3]

pairs = []

for l in letters:
    for n in nums:
        pairs.append(l + str(n))

print(pairs)
```

Output:

```
['A1', 'A2', 'A3', 'B1', 'B2', 'B3']
```

22) Mini table (nested loops + spacing)

One correct answer:

```
for r in range(3):
    line = ""
    for c in range(4):
        line = line + "#"
    print(line)
```

Output:

```
####
####
####
```

23) Multiplication grid (nested loops + `range`)

One correct answer:

```
for n in range(1, 6):
    row = list(range(n, n * 5 + 1, n))
    print(row)
```

Output:

```
[1, 2, 3, 4, 5]
[2, 4, 6, 8, 10]
[3, 6, 9, 12, 15]
[4, 8, 12, 16, 20]
[5, 10, 15, 20, 25]
```