Solution 30: Python Strings & Lists

---

1) String → list of characters (what does it print?)

```
['C', 'A', 'T']
```

---

2) List → string with `"".join(...)` (fill in the blanks)

One correct answer:

```
word = "".join(letters)
```

---

3) True / False (sequence facts)

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

---

4) `len()` practice (what does it print?)

`"Chelsea"` has 7 letters.

```
7
3
```

---

5) Indexing (what does it print?)

`word = "PYTHON"`

- `word[0]` is `P`
- `word[3]` is `H`

- `word[-1]` is the last character: `N`
- `word[-2]` is the second-to-last character: `O`

```
P
H
N
O
```

6) Last item (fill in the blanks)

Use index `-1` for the last item.

```
print(s[-1])
print(nums[-1])
```

7) Fix the bug (short answer)

1. Wrong.
2. `s` is `"hi"`, so it has length 2. Valid indices are `0` and `1`. Index `2` is out of range.
3. One correct line:

```
print(s[-1])
```

8) Slice a chunk (what does it print?)

`word = "ABCDEFG"`

- `word[1:4]` is indices 1,2,3 → `BCD`
- `word[:3]` is indices 0,1,2 → `ABC`
- `word[4:]` is indices 4,5,6 → `EFG`

```
BCD
ABC
EFG
```

9) Slice with step (what does it print?)

`nums = [0, 1, 2, 3, 4, 5, 6, 7, 8]`

- `nums[::2]` takes every 2nd item starting at 0
- `nums[1::3]` starts at index 1, then jumps by 3

```
[0, 2, 4, 6, 8]
[1, 4, 7]
```

10) Reverse with `[::-1]` (what does it print?)

```
racecar
```

11) Make slices (fill in the blanks)

One correct set of slices:

```
print(word[:4])
print(word[4:])
```

12) Update a list item (fill in the blanks)

`"blue"` is at index 2.

```
colors[2] = "purple"
print(colors)
```

Output:

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

13) Try to update a string (short answer)

1. No.
2. TypeError (strings are immutable, so you can’t assign to `s[0]`).
3. One correct way:

```
s = "blue"
new_s = "g" + s[1:]
print(new_s)
```

This prints:

```
glue
```

---

14) Count the letter `"a"` (fill in the blanks)

Fill-ins:

- `count = 0`
- `count = count + 1`

Full code:

```
word = "banana"

count = 0
for ch in word:
    if ch == "a":
        count = count + 1

print(count)
```

---

15) Every 2nd character (fill in the blanks)

Take every 2nd character starting at index 0:

```
print(s[::2])
```

Output:

```
HLO
```

---

16) Palindrome check using reverse (fill in the blanks)

```
print(s == s[::-1])
```

It prints:

```
True
```