Klicker Questions

March 19, 2024

[1]:

```
def concatenate(list1: list, list2: list) -> list:
    for x in list2:
        list1.append(x)
```

[2]:

```
x = [1,2,3]
y = [4,5,6]
concatenate(x,y)
print(x)
print(y)
```

```
[1, 2, 3, 4, 5, 6]
[4, 5, 6]
```

[1]:

```
def repeat(text: str) -> str:
    return text * REPEAT_FACTOR
```

[4]:

```
REPEAT_FACTOR = 5
repeat("Hello")
```

[4]:

```
'HelloHelloHelloHelloHello'
```

[3]:

```
def reverse(lst: list) -> list:
    return lst[::-1]
```

[4]:

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

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

[7]:

```
import random
def random_answer(question: str) -> str:
    answers = [ "It is certain", "Ask again later",
                  "My sources say no"]
    return random.choice(answers)
```

[8]:

```
random_answer("Is this a pure function?")
```

[8]:

```
'My sources say no'
```

What is the order of the mobile phones in x?

[9]:

```
phones = [
    {"make": "Nokia", "model": 216, "color": "Black"},
    {"make": "Mi Max", "model": "2", "color": "Gold"},
    {"make": "Samsung", "model": 7, "color": "Blue"},
]

x = sorted(phones, key=lambda x: x['color'])
```

[10]:

```
x
```

[10]:

```
[{'make': 'Nokia', 'model': 216, 'color': 'Black'},
 {'make': 'Samsung', 'model': 7, 'color': 'Blue'},
 {'make': 'Mi Max', 'model': '2', 'color': 'Gold'}]
```

What will the following code print?

[11]:

```
from typing import Iterable, Callable
```

[12]:

```
class Splitter:
    def __init__(self, delimiter: str = " "):
        self.delimiter = delimiter

    def __call__(self, a) -> Iterable[str]:
        return a.split(self.delimiter)

splitter = Splitter("a")
print(splitter("Banana bread"))
```

```
['B', 'n', 'n', ' bre', 'd']
```

What will the following code print?

[13]:

```
def mydecorator(func: Callable) -> Callable:
    def inner(*args) -> str:
        return func(*args).upper()
    return inner

@mydecorator
def myfunc(s: str) -> str:
    rev = s[::-1]
    print(rev)
    return rev
```

```python
print(myfunc("Test"))
```

```
tseT
TSET
```

Will the following code work?

[14]:

```python
class Word:

    def __init__(self, text: str):
        self.text = text

    def __iter__(self):
        for char in self.text:
            return char

for c in Word("Banana"):
    print(c)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[14], line 10
      7         for char in self.text:
      8             return char
---> 10 for c in Word("Banana"):
     11     print(c)

TypeError: iter() returned non-iterator of type 'str'
```

What will the following code print?

[15]:

```python
def foo() -> str:
    x = "world"
    y = "!?!?"
    yield "Hello"
    for i in x:
        yield i
    yield y * 2

print(list(foo()))
```

```
['Hello', 'w', 'o', 'r', 'l', 'd', '!?!?!?!?']
```

[ ]:

```python
```