

py101_lesson_3_easy_1

December 4, 2024

1 Lesson 3 Practice Problems: Easy 1

1.1 Question 1

Will the code below raise an error?

A. Yes, because index 6 doesn't exist.

```
[1]: numbers = [1, 2, 3]
numbers[6] = 5
```

```
-----
IndexError                                                 Traceback (most recent call last)
Cell In[1], line 2
      1 numbers = [1, 2, 3]
----> 2 numbers[6] = 5

IndexError: list assignment index out of range
```

1.2 Question 2

How can you determine whether a given string ends with an exclamation mark (!)? Write some code that prints True or False depending on whether the string ends with an exclamation mark.

```
[3]: def exclaims(str):
        return True if str[-1] == "!" else False

str1 = "Come over here!"    # True
str2 = "What's up, Doc?"    # False
print(exclaims(str1))
print(exclaims(str2))
```

```
True
False
```

The suggested solution uses the string method `endswith()`, which is handy. I should remember it - it has a start and end value, which should be useful for file extensions

1.3 Question 3

Starting with the string `famous_words = "seven years ago..."` show two different ways to create a new string with “Four score and” prepended to the front of the string.

```
[1]: famous_words = "seven years ago..."
early_famous_words = "Four score and"

newstring_1 = early_famous_words + " " + famous_words
newstring_2 = f'{early_famous_words} {famous_words}'

print(newstring_1)
print(newstring_2)
```

Four score and seven years ago...
Four score and seven years ago...

1.4 Question 4

Using the following string, print a string that contains the same value, but using all lowercase letters except for the first character, which should be capitalized.

```
[7]: munsters_description = "the Munsters are CREEPY and Spooky."
print(munsters_description.capitalize())
```

The munsters are creepy and spooky.

1.5 Question 5

Starting with the string: `munsters_description = "The Munsters are creepy and spooky."` print the string with the case of all letters swapped: `"tHE mUNSTERS ARE CREEPY AND SPOOKY."` That is, lowercase letters are converted to uppercase, and uppercase letters are converted to lowercase”

```
[10]: munsters_description = "The Munsters are creepy and spooky."
print(munsters_description.swapcase())
```

tHE mUNSTERS ARE CREEPY AND SPOOKY.

1.6 Question 6

Determine whether the name Dino appears in the strings below – check each string separately:

```
str1 = "Few things in life are as important as house training your pet dinosaur."
str2 = "Fred and Wilma have a pet dinosaur named Dino."
```

```
[12]: str1 = "Few things in life are as important as house training your pet dinosaur.
        "
str2 = "Fred and Wilma have a pet dinosaur named Dino."

print("Dino" in str1)
```

```
print("Dino" in str2)
```

False
True

1.7 Question 7

How can we add the family pet, “Dino”, to the following list

```
flintstones = ["Fred", "Barney", "Wilma", "Betty", "Bammam", "Pebbles"]
```

```
[14]: flintstones = ["Fred", "Barney", "Wilma", "Betty", "Bammam", "Pebbles"]  
  
flintstones.append("Dino")  
  
print(flintstones)
```

```
['Fred', 'Barney', 'Wilma', 'Betty', 'Bammam', 'Pebbles', 'Dino']
```

1.8 Question 8

In the previous problem, our first answer added ‘Dino’ to the list like this: “`flintstone.append(“Dino”)`”. How can we add multiple items to our list (e.g., ‘Dino’ and ‘Hoppy’)? Replace the call to append with another method invocation.

```
[17]: flintstones = ["Fred", "Barney", "Wilma", "Betty", "Bammam", "Pebbles"]  
new_members = ["Dino", "Hoppy"]  
flintstones.extend(new_members)  
  
print(flintstones)
```

```
['Fred', 'Barney', 'Wilma', 'Betty', 'Bammam', 'Pebbles', 'Dino', 'Hoppy']
```

1.9 Question 9

Print a new version of the sentence given by advice that ends just before the word house. Don’t worry about spaces or punctuation: remove everything starting from the beginning of house to the end of the sentence.

```
[18]: advice = "Few things in life are as important as house training your pet  
↳dinosaur."  
print(advice[0:39])
```

```
Few things in life are as important as
```

The suggested solution uses `.split()`, which is better since it can call a substring and doesn’t rely on me counting letters like a goof.

1.10 Question 10

Print the following string with the word important replaced by urgent:

```
[20]: advice = "Few things in life are as important as house training your pet  
↳dinosaur."  
  
print(advice.replace("important", "urgent"))
```

Few things in life are as urgent as house training your pet dinosaur.

```
[ ]:
```