COMP10001 Foundations of Computing
Semester 1, 2021

Tutorial Questions: Week 5

— VERSION: 1476, DATE: MARCH 26, 2021 —

Discussion

1. What is a "method"? How do methods differ from functions? How are they the same?

   Now try Exercise 1

2. What is a "tuple"? What does it store?
3. What is a "list"? How is it different to a tuple?
4. How do we add and remove items from a list?
5. How can lists and tuples be indexed or sliced?

   Now try Exercise 2

6. What is "iteration" in programming? Why do we need it?
7. What are the two main types of loop in python? How do we write them?
8. What do we mean by the "loop variable" in a `for` loop?
9. What are the differences between the two main types of loops? In which situations are they used?
10. Is it always possible to convert a `while` loop into a `for` loop and vice versa? How do we do it?

    Now try Exercises 3 – 6

Exercises

1. Evaluate the following method calls given the assignment `s = "Computing is FUN!"` Think about the input and output of each method. You're not expected to know all methods for all types: if you haven't seen some of these before, your best guess based on the name will probably be right!

   (a) `s.isupper()`
   (b) `s.upper()`
   (c) `s.endswith("FUN!")`
   (d) `s.count('i')`
   (e) `s.strip('!')`
   (f) `s.replace('i', '!')`

2. Evaluate the following given the assignment `lst = [2, ("green", "eggs", "ham"), False]`

   (a) `lst[2]`
   (b) `lst[1][-2]`
   (c) `lst[1][-2][:3]`
   (d) `lst.append(5); print(lst)`
   (e) `lst.pop(2); print(lst)`

3. What is the output of the following snippets of code containing loops?

   (a)
```
i = 2
while i < 8:
    print(f"The square of {i} is {i * i}")
    i = i + 2
```

(b)
```
for ingredient in ("corn", "pear", "cream", "fish"):
    if ingredient.startswith('c'):
        print(ingredient, "is delicious!")
    else:
        print(ingredient, "is not!")
```

(c)
```
i = 0
colours = ("pink", "red", "blue", "gold", "red", "red", "grey")
while i < len(colours):
    if colours[i] == "red":
        print("Found red at index", i)
    i += 1
```

(d)
```
MIN_WORD_LEN = 4
long_words = 0
text = "Once upon a time there lived a princess"
for word in text.split():
    if len(word) > MIN_WORD_LEN:
        print(word, "is too long!")
        long_words += 1
print(long_words, "words were too long")
```

4. Rewrite the loops in Questions 3a and 3b, converting `for` loops to `while` loops and vice versa.

5. Consider the following `while` loop and two conversions to `for` loops. Are the two `for` loops equivalent? Why might you choose one over the other?

```
count = 0
items = ('eggs', 'spam', 'more eggs')
while count < len(items):
    print(f"need to buy {items[count]}")
    count += 1
```

```
items = ('eggs', 'spam', 'more eggs')
for count in range(len(items)):
  print(f"need to buy {items[count]}")
```

```
items = ('eggs', 'spam', 'more eggs')
for item in items:
  print(f"need to buy {item}")
```

6. Do the following code snippets do the same thing? What are some advantages and disadvantages of each snippet? What if we needed a hundred different types of tool?

```
print("We need some saws")
print("We need some hammers")
print("We need some cogs")
print("We need some nails")
```

```
def get_str(part):
    return f"We need some {part}"

print(get_str("saws"))
print(get_str("hammers"))
print(get_str("cogs"))
print(get_str("nails"))
```

```
def get_str(part):
    return f"We need some {part}"

parts = ("saws", "hammers", "cogs", "nails")

for part in parts:
    print(get_str(part))
```

Problems

1. Write a function which takes a positive integer input `n` and prints the thirteen times tables from `1 * 13` until `n * 13`.

2. Write a function which converts a temperature between degrees Celsius and Fahrenheit. It should take a float, the temperature to convert, and a string, either `'c'` or `'f'` indicating a conversion from degrees Celsius and Fahrenheit respectively. The formulae for conversion are below.

$$C = \frac{F - 32}{1.8} \qquad F = C \times 1.8 + 32$$

3. Write a function which takes a tuple of strings and returns a list containing only the strings which contain at least one exclamation mark or asterisk symbol.