

Outline: Chapter 9 demos and Practice quiz 4

- [Announcements](#)
- [Today's class](#)
- [Python string methods](#)
- [Python list methods](#)
- [Review 'while' loop and modules](#)
- [Practice Quiz 4](#)

Announcements

1. [PA 1 Flexdoko](#)

- Reading quiz in canvas - Done
- Part A due Friday 10/25 (30 pts)
- Part B due Friday 11/01 (70 pts) - Extended
 - PA2 will be posted next Wednesday 10/30
 - Complete 1-2 functions this Weekeend

2. Actual quiz 4 Next Thursday 10/31, in King248

- Covers chapters 7 and 8: [while loop](#) and [modules](#)
- New way: all written format
- Make sure you review the practice quiz 4 and solution right before the actual quiz

Today's class

1. (25 mins) In-class practices: Strings and Lists
 - Python string methods review, practices
 - Python list methods review, practices
 - Tips to use string/list methods in Python
2. (20 mins) Review [while](#) loop and modules
3. (30 mins) Practice quiz 4

Python string methods

Commonly used Python string methods.

Methods	Returns ?
find(x)	Returns the index of the first occurrence of the substring x or -1 if not found.
count(x)	Returns the number of times x occurs in the string.
isdigit()	Returns True if all characters are the numbers 0-9.
islower()	Returns True if all cased characters are lowercase letters.
isupper()	Returns True if all cased characters are uppercase letters.
startswith(x)	Returns True if the string starts with x .

Methods	Returns ?
<code>endswith(x)</code>	Returns <code>True</code> if the string ends with <code>x</code> .

Find more String methods: [A summary of all string methods](#)

Tips to use String methods in Python

1. Understand the Method's Return Type

- String methods like `replace()`, `upper()`, and `lower()` return new strings and do not modify the original string (strings in Python are immutable).
- `find(x)` returns index or -1

2. Check Case Sensitivity Before Using Case Methods

- use `islower()` or `isupper()` to check if the case conversion is necessary
- use `lower()` or `upper()` to standardize the case and avoid mismatches due to case differences

Practice 1

Define a Python function named `sumDigits(s)` returns the sum of the digits 0-9 that appear in the string `s`, ignoring all other characters. Return 0 if there are no digits in the string.

```
# Example
>>> sumDigits("aa1bc2d3")
6
>>> sumDigits("jmu")
0
```

Practice 2

Define a Python function named `end_second(first, second)` that returns `True` if the second string appears at the end of the first string, ignoring case differences.

```
# Example
>>> end_second("csJMU", "jmu")
True
```

Python list methods

Commonly used Python list methods

Adding elements

- `append(x)`
- `insert(i, x)`

Removing elements

- `remove(x)` VS. `pop(i)`

Advanced

- `index(x)`: Return index of first item in list with value `x`.
 - Python list has no `find(x)` method. The `find(x)` method is specific to strings
- `count(x)`: Count the number of times value `x` is in list.

Python function/methods that convert strings into lists.

- The `list()` function can be used to convert a string into a list of individual characters.
- The `split()` list method splits a string into a list of substrings based on a specified delimiter (default is whitespace).
 - The `join()` string method performs the inverse operation of `split()` by joining a list of strings together to create a single string.
 - `ss = "/".join(["10", "23", "2024"])`

More practices: Codingbat [List-2](#)

- Complete it after the class if you are looking for more practices

Review while loop and modules

Chapter 7: While Loops

- `while` loop 3 elements: initialization, condition, and updates
- `break` and `continue`
- The `random` module
 - `randint()`, `random()`, `seed()`
 - How to use them?
- Review `while` loop: [examples and demos](#)

Practice

Define a Python function `get_evens()`, which prompt the user to enter integer values using the keyboard until they enter `Done`, then return a list of all even numbers.

Solution without using `break` or `continue`

```
def get_evens():
    evens = []
    v = input("input an int: ")
    while v != "Done":
        number = int(v)
        if number % 2 == 0:
            evens.append(number)
```

```
v = input("input an int: ")  
return evens
```

Solution using `break` and/or `continue`

```
# TBA after the class
```

Chapter 8: Modules

- How to use `import`
- `if __name__ == "__main__"`
- `Pytest`
 - PA2 requires Pytest

Practice Quiz 4

- (5 mins) Preparation: Please put away your devices, take out your pencils, and write your names on the quiz sheets.
 - Wait for further instructions before starting.
- (25 mins) Quiz: Complete the quiz on your own, without using any external materials.