

EEE110 Computer Programming

Strings & Dictionaries and Sets

Dr Kasim Zor

April 4, 2023

Strings



Question 1. Extracting Characters from a String - Part 1/2 (15 minutes)

At a university, each student is assigned a system login name, which the student uses to log into the campus computer system. As part of your internship with the university's Information Technology department, you have been asked to write the code that generates system login names for students. You will use the following algorithm to generate a login name:

- Get the first three characters of the student's first name. (If the first name is less than three characters in length, use the entire first name.)
- Get the first three characters of the student's last name. (If the last name is less than three characters in length, use the entire last name.)

Strings



Question 1. Extracting Characters from a String - Part 2/2 (15 minutes)

- Get the last three characters of the student's ID number. (If the ID number is less than three characters in length, use the entire ID number.)
- Concatenate the three sets of characters to generate the login name.

For example, if a student's name is Amanda Spencer, and her ID number is ENG6721, her login name would be AmaSpe721. You decide to write a function named `get_login_name` that accepts a student's first name, last name, and ID number as arguments, and returns the student's login name as a string. You will save the function in a module named `login.py`. This module can then be imported into any Python program that needs to generate a login name.

Strings



Solution 1 - Part 1/2

```
def get_login_name(first, last, idnumber):  
    set1 = first[0 : 3]  
  
    set2 = last[0 : 3]  
  
    set3 = idnumber[-3 :]  
  
    login_name = set1 + set2 + set3  
  
    return login_name
```

Solution 1 - Part 2/2

```
def main():
    first = 'Kasım' #input('Enter your first name: ')
    last = 'Zor' #input('Enter your last name: ')
    idnumber = '2004514018' #input('Enter your student ID number: ')

    print('Your system login name is: ')
    print(get_login_name(first, last, idnumber))

main()

## Your system login name is:
## KasZor018
```

Question 2. Validating the Characters in a Password - Part 1/2
(15 minutes)

At the university, passwords for the campus computer system must meet the following requirements:

- The password must be at least seven characters long.
- It must contain at least one uppercase letter.
- It must contain at least one lowercase letter.
- It must contain at least one numeric digit.

When a student sets up his or her password, the password must be validated to ensure it meets these requirements. You have been asked to write the code that performs this validation. You decide to write a function named `valid_password` that accepts the password as an argument and returns either `true` or `false`, to indicate whether it is valid.

Question 2. Validating the Characters in a Password - Part 2/2
(15 minutes)

Here is the algorithm for the function, in pseudocode:

```
valid_password function:
Set the correct_length variable to false
Set the has_uppercase variable to false
Set the has_lowercase variable to false
Set the has_digit variable to false
If the password's length is seven characters or greater:
    Set the correct_length variable to true
    for each character in the password:
        if the character is an uppercase letter:
            Set the has_uppercase variable to true
        if the character is a lowercase letter:
            Set the has_lowercase variable to true
        if the character is a digit:
            Set the has_digit variable to true
If correct_length and has_uppercase and has_lowercase and has_digit:
    Set the is_valid variable to true
else:
    Set the is_valid variable to false
Return the is_valid variable
```

Solution 2 - Part 1/2

```
def valid_password(password):
    correct_length = has_uppercase = has_lowercase = \
    = has_digit = False
    if len(password) >= 7:
        correct_length = True
        for ch in password:
            if ch.isupper(): has_uppercase = True
            if ch.islower(): has_lowercase = True
            if ch.isdigit(): has_digit = True
        if correct_length and has_uppercase and \
        has_lowercase and has_digit: is_valid = True
        else: is_valid = False
    return is_valid
```

Solution 2 - Part 2/2

```
def main():
    password = 'kaszor018' #input('Enter your password: ')

    while not valid_password(password):
        print('That password is not valid.')
        password = 'KasZor018' #input('Enter your password: ')

    print('That is a valid password.')

main()

## That password is not valid.
## That is a valid password.
```

Question 3. Storing Names and Birthdays in a Dictionary - Part 1/2 (30 minutes)

In this section, we look at a program that keeps your friends' names and birthdays in a dictionary. Each entry in the dictionary uses a friend's name as the key, and that friend's birthday as the value. You can use the program to look up your friends' birthdays by entering their names. The program displays a menu that allows the user to make one of the following choices:

- 1 Look up a birthday
- 2 Add a new birthday
- 3 Change a birthday
- 4 Delete a birthday
- 5 Quit the program

Question 3. Storing Names and Birthdays in a Dictionary - Part 2/2 (30 minutes)

The program initially starts with an empty dictionary, so you have to choose item 2 from the menu to add a new entry. Once you have added a few entries, you can choose item 1 to look up a specific person's birthday, item 3 to change an existing birthday in the dictionary, item 4 to delete a birthday from the dictionary, or item 5 to quit the program.

Solution 3 - Part 1/4

```
LOOK_UP = 1
ADD = 2
CHANGE = 3
DELETE = 4
QUIT = 5
def main():
    birthdays = {}
    choice = 0
    while choice != QUIT:
        choice = get_menu_choice()
        if choice == LOOK_UP: look_up(birthdays)
        elif choice == ADD: add(birthdays)
        elif choice == CHANGE: change(birthdays)
        elif choice == DELETE: delete(birthdays)
```

Solution 3 - Part 2/4

```
def get_menu_choice():
    print()
    print('Friends and Their Birthdays')
    print('-----')
    print('1. Look up a birthday')
    print('2. Add a new birthday')
    print('3. Change a birthday')
    print('4. Delete a birthday')
    print('5. Quit the program')
    print()
    choice = 2 #int(input('Enter your choice: '))
    while choice < LOOK_UP or choice > QUIT:
        choice = int(input('Enter a valid choice: '))
    return choice
```

Solution 3 - Part 3/4

```
def look_up(birthdays):
    name = input('Enter a name: ')
    print(birthdays.get(name, 'Not found.'))

def add(birthdays):
    name = input('Enter a name: ')
    bday = input('Enter a birthday: ')
    if name not in birthdays:
        birthdays[name] = bday
    else:
        print('That entry already exists.')
```

Solution 3 - Part 4/4

```
def change(birthdays):
    name = input('Enter a name: ')
    if name in birthdays:
        bday = input('Enter the new birthday: ')
        birthdays[name] = bday
    else:
        print('That name is not found.')
```

```
def delete(birthdays):
    name = input('Enter a name: ')
    if name in birthdays:
        del birthdays[name]
    else:
        print('That name is not found.')
```

```
main()
```

Question 4. Set Operations - Part 1/2 (20 minutes)

In this section, you will look at a program that creates two sets: one that holds the names of students on the baseball team, and another that holds the names of students on the basketball team. The program then performs the following operations:

- It finds the intersection of the sets to display the names of students who play both sports.
- It finds the union of the sets to display the names of students who play either sport.
- It finds the difference of the baseball and basketball sets to display the names of students who play baseball but not basketball.

Question 4. Set Operations - Part 2/2 (20 minutes)

- It finds the difference of the basketball and baseball (basketball – baseball) sets to display the names of students who play basketball but not baseball. It also finds the difference of the baseball and basketball (baseball – basketball) sets to display the names of students who play baseball but not basketball.
- It finds the symmetric difference of the basketball and baseball sets to display the names of students who play one sport but not both.

Solution 4 - Part 1/6

```
baseball = set(['Jodi', 'Carmen', 'Aida', 'Alicia'])
basketball = set(['Eva', 'Carmen', 'Alicia', 'Sarah'])

print('The following students are on the baseball team:')

## The following students are on the baseball team:

for name in baseball:
    print(name) # Display members of the baseball set.

## Aida
## Alicia
## Carmen
## Jodi
```

Solution 4 - Part 2/6

```
print('The following students are on the basketball team:')

## The following students are on the basketball team:

for name in basketball:
    print(name) # Display members of the basketball set.

## Alicia
## Carmen
## Eva
## Sarah
```

Solution 4 - Part 3/6

```
print('The following students play both baseball and basketball:')

## The following students play both baseball and basketball:

for name in baseball.intersection(basketball):
    print(name) # Demonstrate intersection

## Alicia
## Carmen
```

Solution 4 - Part 4/6

```
print('The following students play either baseball or basketball:')

## The following students play either baseball or basketball:

for name in baseball.union(basketball):
    print(name) # Demonstrate union

## Alicia
## Jodi
## Eva
## Sarah
## Carmen
## Aida
```

Solution 4 - Part 4/6

```
print('The following students play baseball, but not basketball:')

## The following students play baseball, but not basketball:

for name in baseball.difference(basketball):
    print(name) # Demonstrate difference of baseball and basketball

## Jodi
## Aida
```

Solution 4 - Part 5/6

```
print('The following students play basketball, but not baseball:')

## The following students play basketball, but not baseball:

for name in basketball.difference(baseball):
    print(name) # Demonstrate difference of basketball and baseball

## Eva
## Sarah
```

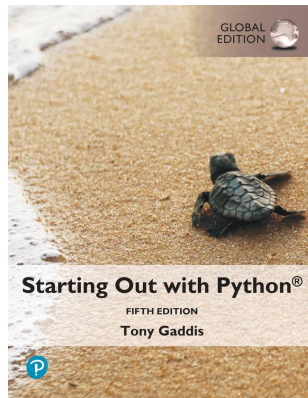
Solution 4 - Part 6/6

```
print('The following students play one sport, but not both:')

## The following students play one sport, but not both:

for name in baseball.symmetric_difference(basketball):
    print(name) # Demonstrate symmetric difference

## Jodi
## Eva
## Sarah
## Aida
```



Aforementioned contents are adapted from the book:

- 'Starting out with Python' written by Tony Gaddis.