Programming Assignment 05
-
Functions

Instructions

This programming assignment consists of **2 programming exercises**.
You have to:

1. **download** the empty Python files on NYU Classes
2. **edit** them according to the assignment
3. **verify** on your computer that it works
4. **upload** them back on NYU Classes (**do not change the filenames**)

Exercise 1 - Diamond function

Write the code for function `diamond` in the file `exercise1.py`.

The function `diamond`:

- takes one **parameter** `N` (type `int`), its value must be **odd** and in the **range** $[3, 99]$
- it draws a **square shape** made of integer values and spaces:
  - integer values are always printed with **2 digits** (for example, value `7` is printed `07`)
  - every line consists of integer values from `1` up to `N`, where some values have to be replaced by spaces instead
  - the **first** and **last row** are **full** (no space)
  - the **middle row** is **empty** except for values `1` and `N`,
  - each row from the first one to the middle one is increasingly empty from its center
  - each row from the middle one to the last one is increasingly full to its center

Below are a couple of example outputs when using the `diamond` function from the shell:

```
>>> diamond(5)
0102030405
0102  0405
01      05
0102  0405
0102030405
```

```
>>> diamond(15)
010203040506070809101112131415
01020304050607  09101112131415
010203040506      101112131415
0102030405          1112131415
01020304              12131415
010203                  131415
0102                      1415
01                          15
0102                      1415
010203                  131415
01020304              12131415
0102030405          1112131415
010203040506      101112131415
01020304050607  09101112131415
010203040506070809101112131415
```

Exercise 2 - Password validation

Write the code for function `valid_password` in the file `exercise2.py`.

The function `valid_password`:

- takes 1 parameter `pwd` → type `str`: password
- **displays** a message (depending on password rules (see below))
- returns if the password is valid → type `bool`

Valid password rules are:

1. The password is made of at least 10 characters
2. It only contains alphanumeric characters (no space, no underscore, ...)
3. There is at least:
   - 1 uppercase character
   - 1 lowercase character
   - 1 digit

The **printed out message/returned value** will be:

- `Password is too short` / `False` if rule #1 is not respected
- `Wrong characters` / `False` if rule #2 is not respected
- `Need at least 1 uppercase letter, 1 lowercase letter and 1 digit` / `False` if rule #3 is not respected
- if multiple rules are not respected, only print the message corresponding to the first rule that is not respected
- `Valid password` / `True` if all the rules are respected

Hint: the methods `.isalpha()`, `.isdigit()` and `.isalnum()` can be useful

Below are a couple of example outputs when using the `valid_password` function from the shell:

```
>>> valid_password('qwertyuiop')
Need at least 1 uppercase letter, 1 lowercase letter and 1 digit
False
```

```
>>> valid_password('ICPSpring2019')
Valid password
True
```

```
>>> valid_password('pwd!!!')
Password is too short
False
```

```
>>> valid_password('PWD01_#PWD01')
Wrong characters
False
```