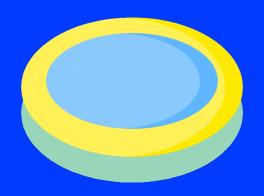


# WADING POOL

< 03 - STRINGS />



## **WADING POOL**



#### **OH MY GIT!**

In addition to the tasks below, you must go as far as possible in this game. Work on it as soon as you have a bit of time, or whenever you need a break in you day!



i

For EACH future exercise, create a separate file properly named, such as task4.2.

### **Strings**

#### **Task 1.1**

Store a long string (such as a sentence), in a variable. Then, print it.

#### **Task 1.2**

Print the 1<sup>st</sup> character of this string. Then, print the 13<sup>th</sup> character of this string.

#### **Task 1.3**

Print the last character of this string.

#### **Task 1.4**

Using one line of code, print from the 5<sup>th</sup> to the 10<sup>th</sup> character of this string.



### **String methods**

#### **Task 2.1**

Write a snippet of code to transforms this string in lower case.



Your code should work for any non-empty string, such as:

Just remember ALL CAPS when you spell the man name.

#### **Task 2.2**

Write a snippet of code that replaces every "tu" in a string by "ta".



The input tutu on the tuki-kata gives the output tata on the taki-kata.

#### **Task 2.3**

Explain the following code and predict its output. Then, test your prediction by running the code.

```
string = "Hello world!"
position = string.find("a")
print(position)
```





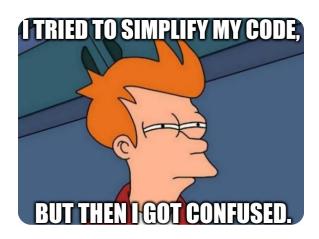


Can you predict the result of the following snippet of code?

```
p = "abcdefghij"
print(p[::-2][:5][::-1][3:])
```

#### **Task 2.5**

Can you break down each step in order to simplify the previous code?



#### **Task 2.6**

Write a snippet of code that prints 10 times a given string.

#### **Task 2.7**

Debug print("hello"+ 42).



#### **Task 2.8**

Complete the following code so that it prints "42 is the answer"contains 16 characters.

```
s1, s2, s3 = "42", "is", "the answer"
print(??????)
```



#### **CHALLENGE**



The challenge uses several notions scattered throughout the subject's exercises. If you don't feel up to it straight away, that's normal. Come back later to try again.

You should compare your productions with others and discuss the pros and cons of each solution. The best answers could be presented publicly.

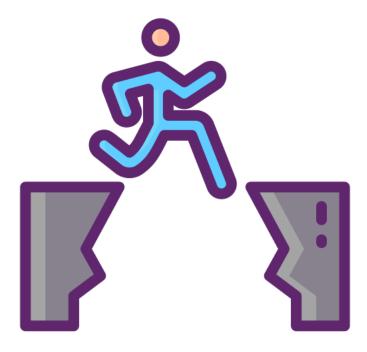
Write a snippet of code that counts the total of occurrences of the strings "cat", "garden" and "mice" in any string, considering that:

- ✓ each substring can be read left to right OR right to left;
- ✓ matches must be case insensitive;
- ✓ you should not bother with overlaps.

i

For instances, the strings:

- ✓ "the CataCat attaCk a Cat" should return 5;
- ✓ "thE Cat's tactic wAS tO surpRISE thE mIce iN tHE gArdeN" should return 4.







## **User input**

#### **Task 3.1**

Ask the user his/her name and then greet him/her with "Hello <Username>!". Careful, only the first letter is capitalized, such as "Hello Archibald!".









#### **Task 3.2**

Prompt the user for a number.

Then, print the **type** of the provided input.



It should return <class 'str'>.

Can you understand what does that means?

#### **Task 3.3**

Prompt the user for two numbers.

Then, print "The sum of the provided numbers is <SUM>.".



For instance, the inputs 42 and 666 should return 708.





Write a program that:

- ✓ ask the user to type a string;
- ✓ extracts the first letter of each word in the string;
- ✓ join these letters to make a word.



For instance, the input Play your trumpet happily on nights. returns Python.

#### **Task 3.5**



Write a program that counts the frequency of each letter from a given text and infers the language of this very text.



#### **Task 3.6**



Copy your previous program. Then, modify it to work with UTF-8.





##