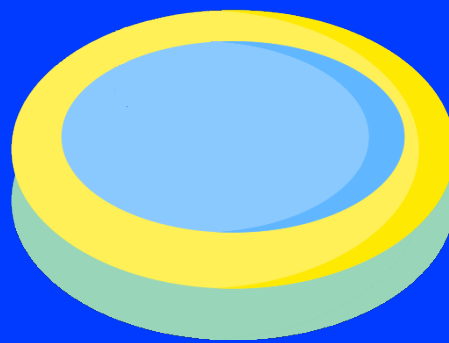




# WADING POOL



< 05 - LISTS />



# WADING POOL



If you need a little break in your day, we suggest you to play [the bandit wargame](#). We also encourage you to resume your game later.



## Lists creation and browsing

### Task 1.1



Create a [list](#) of 5 elements. Then, print its first element.

### Task 1.2



Display the last element of your list.



Your code must be functional whichever number of elements the list contains.

### Task 1.3



Add the integer 42 at the end of your list.  
Then, add the string [forty-two](#) at the end of your list.

### Task 1.4



Display your entire list.  
Then, display each element of your list one by one.

### Task 1.5



Delete the last element of your list.  
Then, display your list to check if you did it properly.



Your code must be functional whichever number of elements the list contains.

### Task 1.6



Add an element at the beginning of the list and display all its elements.

### Task 1.7



Display the sub-list from the 2<sup>nd</sup> to the 4<sup>th</sup> element (included).

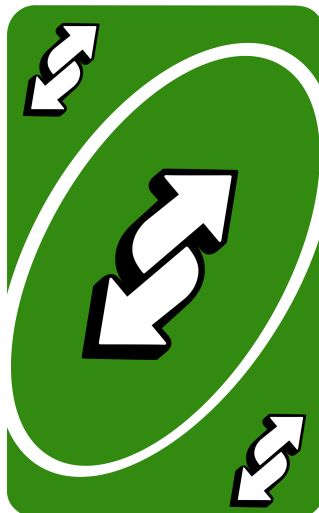


Can you do it in one line?

### Task 1.8



Reverse the list to create a new list with the same elements, but starting from the end.  
Then, display all the elements of this new list.



### Task 1.9



Display one element out of two of the list.

### Task 1.10



Add the ten integers from 11 to 21 at the end of your list.



Please, do not do it in 10 similar lines. Be smart and lazy.

### Task 1.11

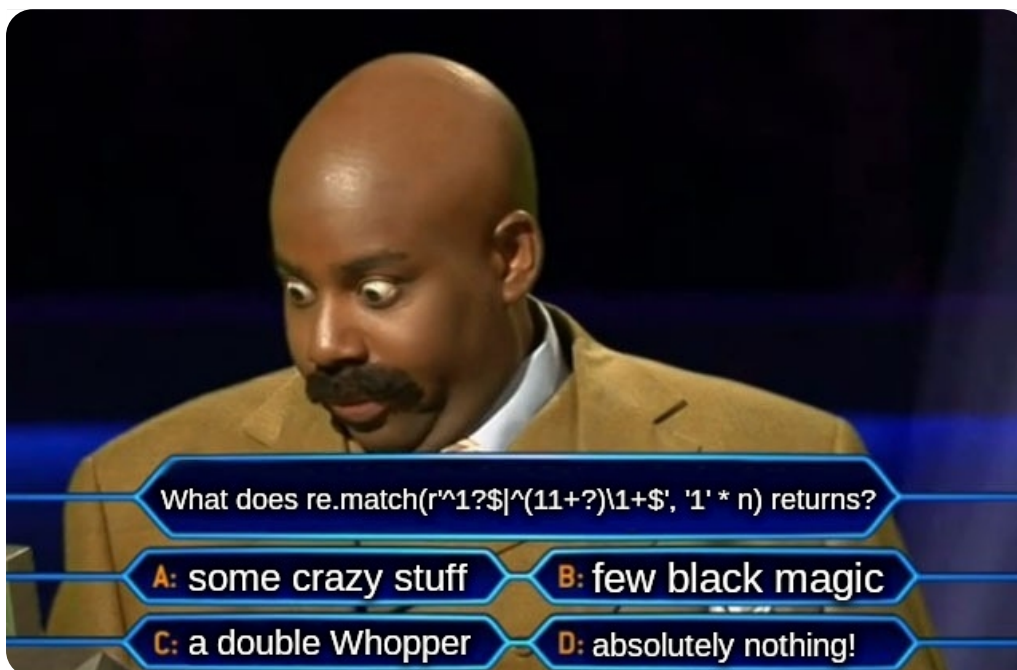


What does the following code do?

```
my_first_list = [4, 5, 6]
my_second_list = [1, 2, 3]
my_first_list.extend(my_second_list)
```

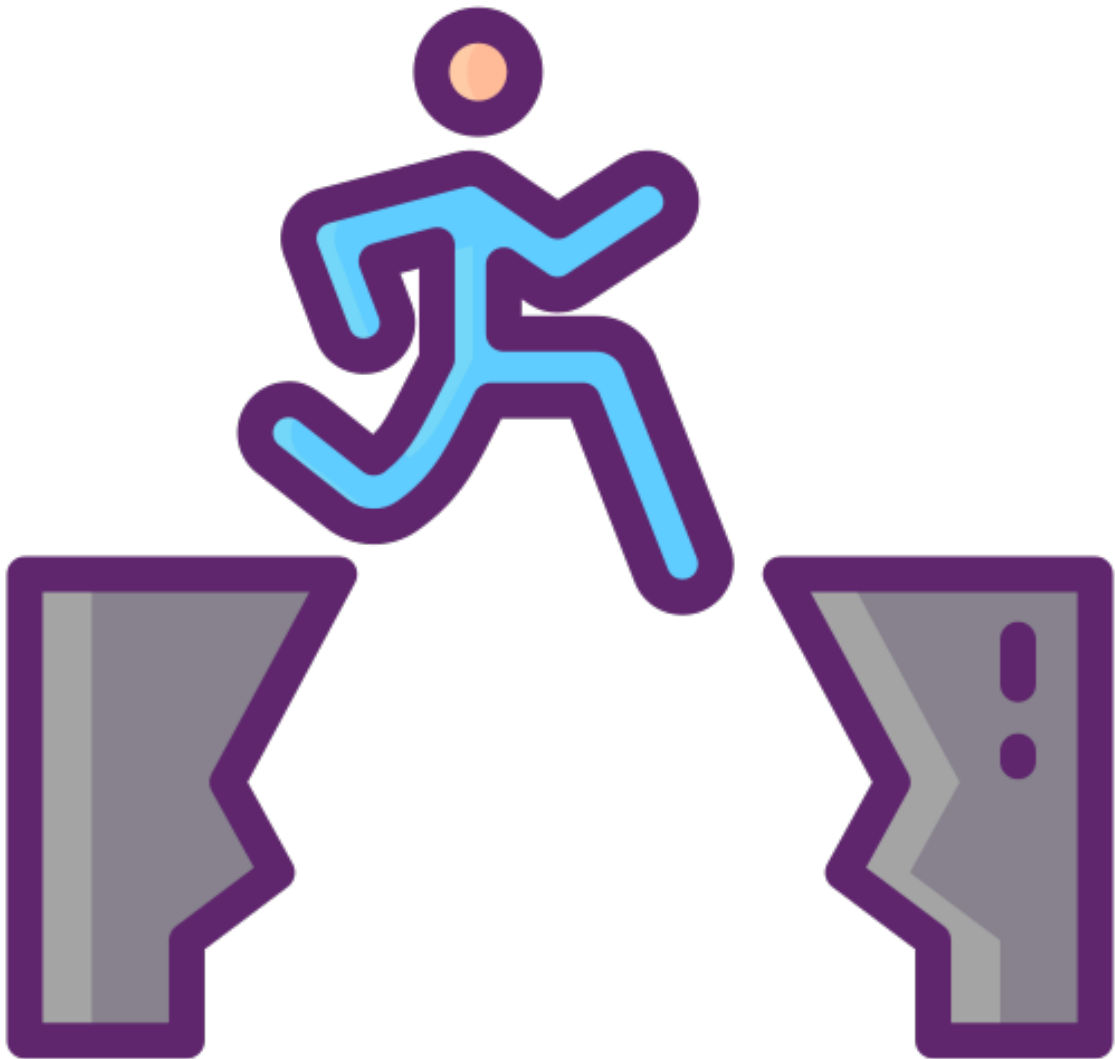
Same with:

```
my_first_list = [7, 8, 9]
my_second_list = [4, 5, 6]
my_first_list = [*my_first_list, *my_second_list]
```



## CHALLENGE

Create a list of 1000000 random integers. Then, sort this list as fast as possible.



If you want to know how long the execution of your program lasted, you can:

- ✓ insert `start=time.time()` at the top of your code and `print(time.time()-start)` at the end ;
- ✓ use the command `time YOUR_SCRIPT.py`.

## Operations on lists

### Task 2.1



Create a list of 5 numbers. Then, print the result of the multiplication of all its elements.



The input [1, 2, 3, 4, 5] should return 120.

### Task 2.2



Test this code and try to explain it: `[x + 10 for x in [3, 2, 6, 7, 1, 4]]`

### Task 2.3



Create a list of 5 numbers. Then, display the smallest element. Finally, display its biggest element.

### Task 2.4



Sort your list in descending order.



## Lets' go further

### Task 3.1

---



Let's consider a list of names (the ambassador's banquet guests).  
Write a program that displays:

- ✓ "welcome in" if a given name belongs to this list ;
- ✓ and "get lost!" otherwise.



### Task 3.2

---



Write a program that deletes all the duplicated elements in a list.  
Test it with [1, 1, 2, 2, 3] and with ['a', 2, 'a', 2, 'A'], they should return 3 elements.

### Task 3.3



Let consider a list of meetings. Each meeting is a list containing the day, the time of the meeting and the name of all the participants.

For instance:

```
[  
  ["Monday", "3:30 PM", "Joe", "Sam"],  
  ["Monday", "4:30 PM", "Bob", "Alice"],  
  ["Tuesday", "3:30 PM", "Joe", "Bob", "Alice", "Sam"],  
  ["Tuesday", "9:30 AM", "Joe", "Bob"]  
]
```

Write a program that, given a name, displays all the day and the time of all meetings in which this person is involved.

### Task 3.4



Hmm... you seem like a fast scorer!

Wander in the halls and find some desperate colleagues that need help.



Of course, review their code. You may even try a bit of peer-coding.



v 2.2

{EPITECH}