Python practice 1

---

0 Getting Inputs

There are many kind of inputs, but we will focus on numbers for now. Python have to deal with these two types of inputs differently.

1. One Line

```
1 2 3 4 5...
```

There is *much* cleaner way to do this, which we will learn next week.

```
lst = input().split() # note that after split(), type is lst.
nums = [] * len(lst)
for i in range(len(nums)):
    nums[i] = int(lst(i))
```

2. Seperate Lines

```
1
2
3
4
5
...
```

We can do this by using For Loops. Usually, the number of inputs is also given. The following code can store numbers given in this manner.

```
k = int(input()) # number of inputs
lst = [0] * k

for i in range(k):
    lst[i] = int(input())
```

1 Lists

Generate a list of numbers [1, 100] with range. Print the value of $1 * 2 * ...99 * 100$.

2 99 dan

Print out the following:

```
1 * 1 = 1
1 * 2 = 2
  ...
9 * 9 = 81
```

We *can* construct the string by concatanation like this:

```
str(8) + ' * ' + str(9) + ' = ' + str(72)
```

but there is a better way. Google It!

3 Lotto

Create and print a list of length *6*, where each element is a random integer in range $[1, 45]$.

```
lst = [0] * 6 # create a list of length 6, with all the elements set to 0.

# Do Something

print(*lst) # remember unpacking?
```

To learn how to generate random numbers Google It! Among all the search result google provides, use stackoverflow.

4 Buy Lotto

Create a list of length *6*, where each element is a random integer in range $[1, 45]$. Now, get input from user(you) until the user has correctly typed the 6 numbers.
Note that order of numbers do not matter and CONFIRM that your program finishes correctly.

5 369

Print numbers in [1, 50]. But for numbers that are multiples of *3* (*e.g.* 3,6,9...) and numbers that contain(*membership??*) *3* (*e.g.* 13, 23...), print clap.

```
...
11
clap
clap
14
clap
16
...
```

6 2nd

Create a list of 100 random numbers.
output: the 2nd largest element from the numbers.
Sure, we can do the following. But solve it without sorting the list.

```
print(sorted(lst[-2]))
```

7 Challange: Fibonacci

The Fibonacci numbers are defined as following:

$$F_0 = 0, F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2}, n > 1$$

Input: N, Output: N^th Fibonacci.
Check that this program also prints F_0 and F_1 well, for input 0 and 1.
Hint: We probably should store F_0 and F_1 somewhere, and work with them.

8 Challange: Matching parentheses Strings

Mismatched ()s ($e.g.$ ”(()”, ”)(()”) are painful when we are coding. Write a program that prints ’yes’ when the ()s match, and ’no’ when they do not match.

```
())()
no
(())()
yes
```

Googling this would probably lead you to the stack data structure. It is pretty simple to implement a stack using python lists. Try to figure it out. But, we do not need a stack to solve this one.