



برنامه نویسی با پایتون

سعید محققى

04: Loops and Data Structures

کار با لیست، حلقه for و ماژول random در پایتون

■ پروژه 3: تولید پسورد قوی (strong password generator)

1. دریافت تعداد حروف از کاربر

2. دریافت تعداد کاراکترهای خاص و اعداد از کاربر

3. تولید پسورد تصادفی با مشخصات داده شده

What will we learn? (Keywords)

- Random Module
- Lists
- Range Function
- For Loop

<https://haveibeenpwned.com/>

سایت‌هایی که اطلاعات کاربرانشان لو رفته

■ List : یکی از 4 نوع متغیر پایتون برای ذخیره گروهی از داده‌ها

■ انواع دیگر: Tuple / Set / Dictionary

```
x = [1, 2, 3, 4]  
x = [1, 2.5, 'a', True] } → list
```

https://www.w3schools.com/python/python_lists.asp

ویژگی‌های لیست‌ها در پایتون

عملیات بر روی لیست (List Operations)

- دسترسی به آیتم‌های لیست https://www.w3schools.com/python/python_lists_access.asp
- تغییر آیتم‌های لیست https://www.w3schools.com/python/python_lists_change.asp
- اضافه کردن آیتم به لیست https://www.w3schools.com/python/python_lists_add.asp
- حذف آیتم از لیست https://www.w3schools.com/python/python_lists_remove.asp
- حلقه روی لیست https://www.w3schools.com/python/python_lists_loop.asp

Python Collections

- ویژگی‌های ساختارهای داده گروهی (Collection ها) در پایتون

Data Structure	Indexing	Ordered	Mutable	Duplicate
[] List	✓	✓	✓	✓
() Tuple	✓	✓	✗	✓
{}	✗	✗	✗	✗
{K:V} Dictionary	✗	✓	✓	✗

■ حلقه روی آیتم‌های لیست

```
names = ['Robot', 'Potato Head', 'Bean']  
for name in names:  
    print(name)  
    print('Mr.', name)
```

Range در پایتون

■ ساختار `range(start, stop, step)`

- `x = range(6)`
- `x = range(3, 6)`
- `x = range(3, 20, 2)`

■ حلقه روی Range

- ```
for n in range(1, 100):
 print(n)
```

■ مثال: محاسبه میانگین نمرات / بازی FizzBuzz / ماکزیمم یا مینیمم اعداد

### ■ استفاده از Enumerate

```
words = ['a', 'b', 'c', 'd']
for i, word in enumerate(words):
 print(i+1, word)
```

```
1 a
2 b
3 c
4 d
```

### ■ استفاده از Iterator

```
nums = [1, 2, 3, 4]
for num in nums:
 print(num)
```

```
1
2
3
4
```

### ■ استفاده از Generator

```
for i in range(5):
 print(i)
```

```
0
1
2
3
4
```

این دستورها را امتحان کنید (List Comprehension):

- `[n for n in range(1,21)]`
- `[n/2 for n in range(1,21)]`



# ماژول Random

■ جستجوی عبارت random module در سایت [askpython.com](http://askpython.com)

```
import random
```

`random.randint(a, b)` → عدد صحیح بین  $a$  و  $b$  (شامل  $a$  و  $b$ )

`random.random()` → عدد اعشاری بین  $0$  و  $1$  (غیر از خود  $1$ )

`random.choice(range(0,10))` → انتخاب یک آیتم تصادفی از یک sequence

```
mylist = ["apple", "banana", "cherry"]
```

```
random.choice(mylist)
```

`random.sample(range(1,7), 2)` → انتخاب چند آیتم تصادفی از یک sequence

■ مثال: برنامه پرتاب سکه / پرتاب تاس / برنامه قرعه کشی

### ■ پروژه 4: برنامه قرعه کشی (ghore keshi)

1. دریافت اسامی و تعداد برنده ها (n) از کاربر

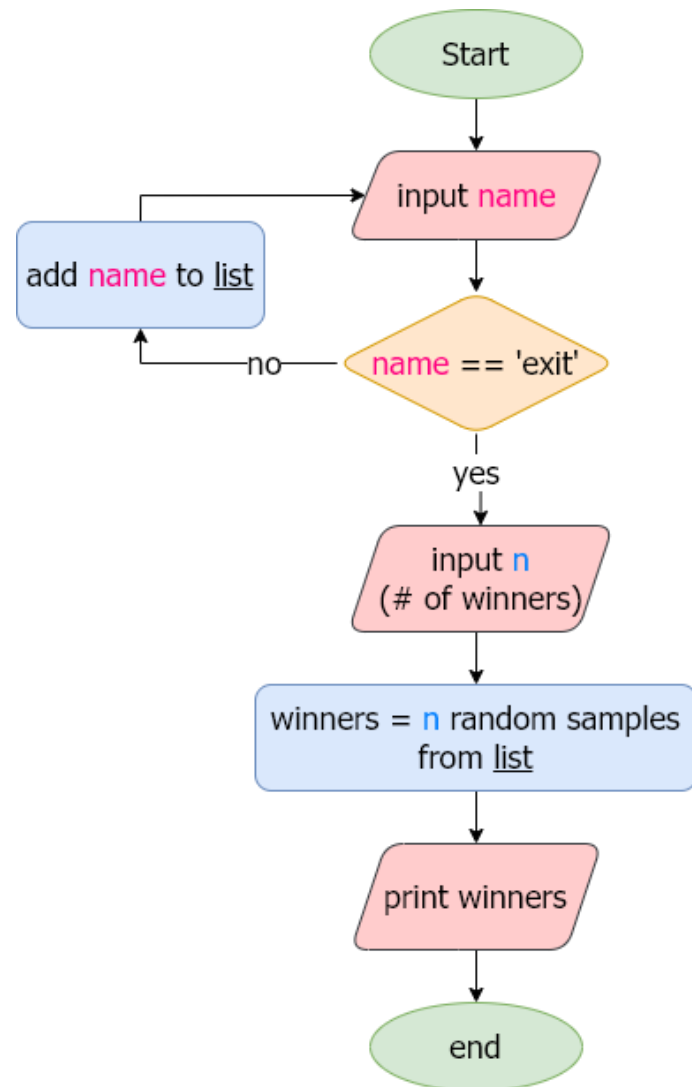
2. انتخاب تصادفی n اسم از لیست اسامی





3. پاک کردن صفحه (با توجه به نوع سیستم عامل)

4. نمایش نتایج به صورت تایپ کاراکتر به کاراکتر

### What will we learn? (Keywords)

- While Loop
- Detect Operating System
- Clear Screen
- Random module → sample
- Time module → sleep
- Create Python Modules



| کاربرد           | شکل                                                                                  |
|------------------|--------------------------------------------------------------------------------------|
| شروع / پایان     |   |
| ورودی / خروجی    |   |
| محاسبه / پردازش  |   |
| شرط / تصمیم‌گیری |  |

## حلقه while در پایتون

### ■ حلقه مشروط

```
i = 1
while i < 5:
 print(i)
 i += 1
```

1  
2  
3  
4

### ■ استفاده از Break

```
i = 1
while True:
 print(i)
 if i == 4:
 break
 i += 1
```

1  
2  
3  
4

### ■ استفاده از Continue

```
i = 1
while i < 10:
 i += 1
 if i % 2 == 0:
 continue
 print(i)
```

3  
5  
7  
9

■ بازی سنگ / کاغذ / قیچی (rocks paper scissors)

1. نمایش منوی انتخاب (سنگ / کاغذ / قیچی / خروج)

2. دریافت انتخاب کاربر

3. انتخاب تصادفی برای ربات

4. محاسبه نتیجه (مقایسه انتخاب ها)

5. بازگشت به منو (تکرار تا وقتی کاربر به بازی پایان دهد)

## What will we learn? (Keywords)

- While Loop
- random.choice
- Ascii Arts
- Create Python Modules

- <https://projecteuler.net/problem=1>

## Multiples of 3 and 5

### Problem 1



If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23.

Find the sum of all the multiples of 3 or 5 below 1000.

- <https://projecteuler.net/problem=2>

## Even Fibonacci numbers

### Problem 2

Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.