



برنامه نویسی با پایتون

سعید محققى

03: Python Tips

رند کردن اعداد اعشاری

- رند کردن با تعیین تعداد رقم اعشار

```
x = 3.1415  
y = round( x , ndigits=2 )
```

← y = 3.14

- استفاده از کتابخانه math

```
import math
```

- رند کردن به بالا با ceil و به پایین با floor (تبدیل به int)

```
a = 2.3  
b = math.ceil(a)  
c = math.floor(a)
```

← b = 3

← c = 2

چک کردن عدد بودن STRING

■ آیا عدد صحیح است؟

True

```
s = "1234"  
s.isdigit()
```

■ آیا عدد است؟ (صحیح و اعشاری و منفی)

False

```
n = "-34" / s = "2.3"  
s.isdigit() / n.isdigit()
```

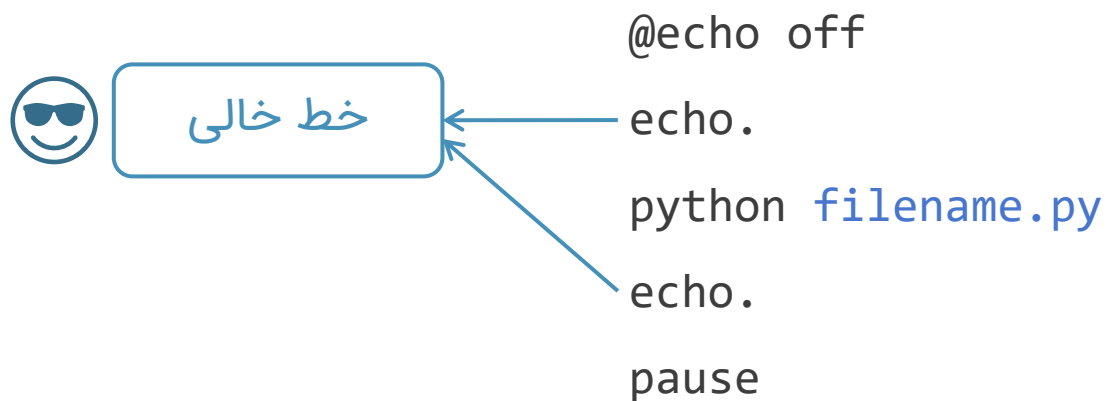
True

```
s.replace('.', '').isdigit()
```

True

```
n.replace('-', '').isdigit()
```

- ساخت فایل با پسوند .bat یا .cmd در کنار کد پایتون (در ویندوز)
- محتوای فایل:



```
@echo off
echo.
python filename.py
echo.
pause
```

- برعکس کردن $\leftarrow [::-1]$

```
s = "salam12345"
```

```
s_r = s[::-1]
```

s_r = '54321malas'

- طول رشته

```
len(s)
```

int

- تعداد یک حرف در رشته

```
s.count('a')
```

- چک کردن وجود یک عبارت در رشته

```
if ('salam' in s):
```

```
...
```

String Methods

- خروج از prompt (python / ipython / IDLE)

Ctrl - D

- خروج از کد (در حال اجرا)

`exit()`

گزینه بهتر ✓

`sys.exit()`

`raise SystemExit`

استفاده از رنگ در print

ANSI کد نماد ←
`\033[32m`
کد رنگ

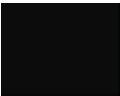










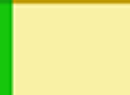



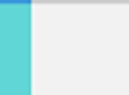
- استفاده از رنگ های ANSI (ANSI Colors)

- 16 رنگ (8 رنگ تیره و 8 رنگ روشن)

- قابلیت نمایش 256 رنگ هم وجود دارد

- `print('\033[34m Salam \033[0m')`
Blue Reset Color

>> کدهای رنگ

30	31	32	33	34	35	36	37
							
							
30;1	31;1	32;1	33;1	34;1	35;1	36;1	37;1

- در حال حاضر نمایش رنگ های ANSI در CMD ویندوز با مشکل مواجه می شود
- راه حل:
- در ابتدای کد (قبل از استفاده از رنگ) دستور زیر را اجرا کنید:

```
import os  
  
os.system('')
```


بررسی پکیج‌ها (کتابخانه‌ها)

- لیست پکیج‌های نصب شده:

```
pip list  
conda list
```

- مشاهده اطلاعات و وابستگی‌های هر پکیج:

```
pip show <package>
```

← `pip show scipy`

- ذخیره خروجی دستورات در یک فایل:

```
pip list > packages.txt
```

<https://docs.conda.io/projects/conda/en/latest/commands.html>

دستورات conda و مقایسه با pip

نصب پکیج‌ها یا کتابخانه‌ها

■ نصب با یکی از دو روش زیر:

مخزن Pypi

<https://pypi.org/> ← `pip install <package name>`

مخزن Anaconda

<https://anaconda.org/> ← `conda install <package name>`

■ نصب از مخزن غیررسمی conda-forge

مخزن conda-forge

<https://conda-forge.org/> ← `conda install <package name> -c conda-forge`

- نصب با یکی از دو روش زیر در CMD:

```
> pip install prettytable  
> conda install keyboard -c conda-forge
```

- کاربرد

```
from prettytable import PrettyTable  
ptt = PrettyTable()  
ptt.field_names = ["Name", "Height", "Weight"]  
ptt.add_row(["Ali", 75, 175])  
ptt.add_row(["Sima", 65, 165])  
print(ptt)
```

<https://pypi.org/project/prettytable>
<https://github.com/jazzband/prettytable>

لینک های مرجع برای prettytable