

پایتون جامع

محمد حسین مقدم خواه

Mhmk.nova@pm.me

جلسه اول



```
1  """
2  TODO This Session Topics:
3      1. Python Use Cases
4      2. Install Python Compiler
5      3. Primal Data Types
6      4. Operators
7      5. Print Function
8      6. Value Assignment
9      7. Assignment Operators
10     8. DocString
11     9. FString
12  """
```

Python Use Cases

توسعه وب: پایتون می‌تواند برای ساخت وبسایت‌ها، ربات‌ها، اپلیکیشن‌ها و سرویس‌های وب به کار رود. پایتون دارای فریم‌ورک‌های قدرتمند و محبوبی مانند Django, Flask, Pyramid و Web2py است که تسهیلات زیادی را برای توسعه دهندگان فراهم می‌کنند.



Python Use Cases

توسعه نرم افزار : پایتون می تواند برای ساخت
نرم افزار های دسکتاپ، موبایل، تجاری، علمی
و هوش مصنوعی به کار رود. پایتون دارای
کتابخانه ها و ابزار های زیادی است که
قابلیت های مختلفی را برای برنامه نویسان
فراهم می کنند. برخی از این کتابخانه ها عبارتند
از : Tkinter, PyQt, Kivy, PyGame,
NumPy, SciPy, TensorFlow,
PyTorch و Keras.



Python Use Cases

پردازش تصویر: پایتون می‌تواند برای اعمال عملیات مختلف بر روی تصاویر دیجیتال به کار رود. پایتون دارای کتابخانه‌های قدرتمند و پرکاربردی مانند OpenCV, Pillow, Scikit-image و Matplotlib است که امکانات زیادی را برای پردازش تصویر فراهم می‌کنند.



Install Python Compiler

برای برنامه نویسی پایتون اول از همه نیاز به مفسر یا
Compiler پایتون داریم که از سایت رسمی پایتون
(<https://www.python.org/downloads/>) قابل
دانلود برای انواع سیستم عامل ها میباشد.

[Donate](#)[GO](#)[Socialize](#)[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

Download the latest version for Windows

[Download Python 3.11.1](#)

Looking for Python with a different OS? Python for [Windows](#),
[Linux/UNIX](#), [macOS](#), [Other](#)

Want to help test development versions of Python? [Prereleases](#),
[Docker images](#)



Support Python in 2022! End of year fundraiser and membership drive are live now!

[GIVE NOW](#)

Active Python Releases

For more information visit the [Python Developer's Guide](#).

Start Python Programing

برای کد نویسی پایتون ابتدا باید با پسوند فایل های پایتون آشنا شویم. فایل های پایتون با پسوند .py ساخته میشوند و برای اجرای آنها باید با استفاده از خط فرمان یا Terminal سیستم عامل خود، کامپایلر پایتون را فراخوانی کرد و بعد نام فایل پایتون خود را در روبرو نوشت تا فایل اجرا گردد و خروجی را به ما نمایش دهد.

Suitable IDE

در قدم بعد نیاز به یک ویرایشگر متن یا IDE داریم که بتوان کد را در آن نوشت و اجرا کرد IDE. پیشنهادی این کارگاه VSCode میباشد که علاوه بر کدهای پایتون میتواند برای همه زبانهای برنامه نویسی در آن کد زد. این IDE را میتوانید از سایت رسمی آن به آدرس

<https://code.visualstudio.com/download>

برای همه سیستم عامل ها دانلود و اجرا کرد.

Version 1.74 is now available! Read about the new features and fixes from November.

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



Windows

Windows 8, 10, 11

User Installer	x64	x86	Arm64
System Installer	x64	x86	Arm64
.zip	x64	x86	Arm64
CLI	x64	x86	Arm64



.deb

Debian, Ubuntu

.rpm

Red Hat, Fedora, SUSE

.deb	x64	Arm32	Arm64
.rpm	x64	Arm32	Arm64
.tar.gz	x64	Arm32	Arm64
Snap	Snap Store		
CLI	x64	Arm32	Arm64



Mac

macOS 10.11+

.zip	Intel chip	Apple silicon	Universal
CLI	Intel chip	Apple silicon	



Primal Data Types

Integer => int

Float => float

Boolean => bool

String => str

Complex



Integers

این نوع داده برای ذخیره اعداد صحیح در پایتون
استفاده میشود
برای مثال:



```
1 22
```

```
2 23333333
```

```
3 9223372036854775807
```



Floats

این نوع داده برای مقادیر اعشاری در پایتون استفاده میشود.

برای مثال:



1 3.14

2 0.212121212121

3 1.7888454884489

Booleans

یک نوع داده منطقی است که فقط مقادیر True و False میگیرد.




```
1 True
2 False
```




Strings



این نوع داده برای ذخیره متن به کار میروند و رشته ای از کارکتر ها را تشکیل میدهند.
برای مثال:



```
1 "Hello World"
2 'This Is A Stirng'
3 "12121212"
4 "A"
```

Complexes

این نوع داده برای ذخیره اعداد مختلط در پایتون استفاده میشود.

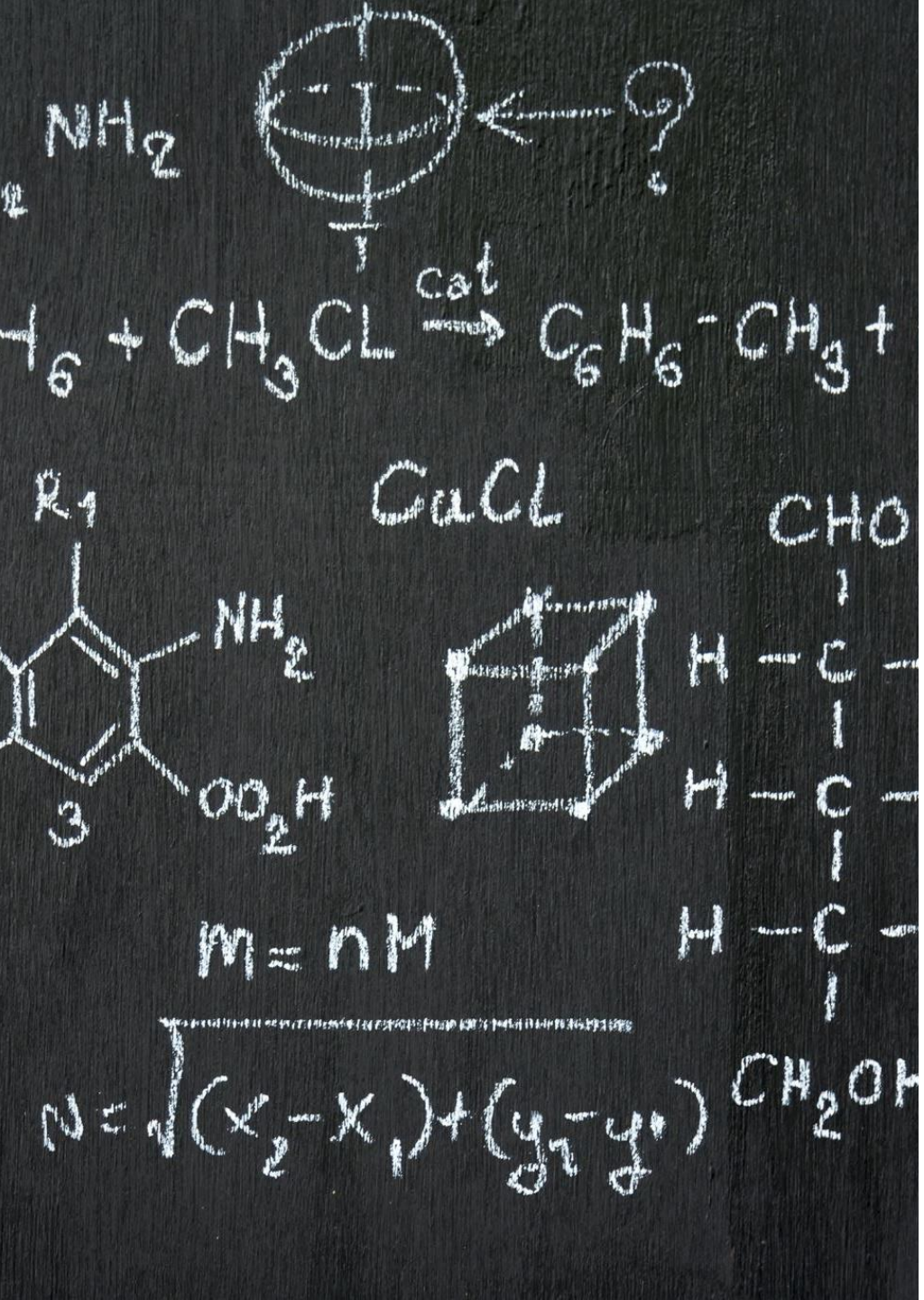
برای تعریف عدد مختلط ابتدا قسمت حقیقی آنرا تعریف میکنیم و بعد قسمت موهومی را به آن با حرف اضافه میکنیم.

برای مثال:



1 $2+3j$

2 $4-5j$




Operators

- $+$ \Rightarrow Sum
- $-$ \Rightarrow Minus
- $/$ \Rightarrow Float Division
- $//$ \Rightarrow Integer Division
- $\%$ \Rightarrow Modulus
- $*$ \Rightarrow Multiply
- $**$ \Rightarrow Exponent



Sum Operator

برای جمع کردن مقادیر در پایتون از عملگر
+ استفاده میکنیم.



```
1 10 + 5
2 OUT => 15
3 "Hello" + " " + "World!"
4 OUT => Hello World!
```

Minus Operator

در پایتون برای تفریق مقادیر از عملگر - استفاده میکنیم.



```
1 10 - 5
```

```
2 OUT => 15
```

```
3 10 - 0.5
```

```
4 OUT => 9.5
```

Float Division

در پایتون برای تقسیم مقادیر با خروجی اعشاری از عملگر `/` استفاده میکنیم.



```
1 10 / 5
```

```
2 OUT => 2.0
```

```
3 7.5 / 3
```

```
4 OUT => 2.5
```


Integer Division

در پایتون برای تقسیم مقادیر با خروجی صحیح از عملگر `//` استفاده میکنیم.



1 10 // 5

2 OUT => 2

3 7.5 // 3

4 OUT => 2

Modulus Division

در پایتون برای گرفتن باقیمانده تقسیم مقادیر
از عملگر % استفاده میکنیم.



```
1 9 % 2
```

```
2 OUT => 1
```

```
3 7.5 % 2
```

```
4 OUT => 1.5
```

Multiply Operator

در پایتون برای ضرب مقادیر از عملگر * استفاده
میکنیم.



```
1 9 * 2
```

```
2 OUT => 18
```

```
3 "Hey" * 2
```

```
4 OUT => Hey Hey
```

Exponent Operator

در پایتون برای ضرب مقادیر از عملگر * استفاده میکنیم.



```
1 9 * 2
```

```
2 OUT => 18
```

```
3 "Hey" * 2
```

```
4 OUT => Hey Hey
```

Exponent Operator

در پایتون برای به توان رساندن مقادیر از عملگر ****** استفاده میکنیم.



```
1 9 ** 2
```

```
2 OUT => 81
```

```
3 1.5 * 2
```

```
4 OUT => 2.25
```



```
1 print("Hello World!")  
2 OUT => Hello World
```



Print Function

در پایتون برای نمایش اطلاعات در ترمینال
از تابع `print()` استفاده میکنیم

Variables

در اکثر اوقات ما نیاز داریم که مقداری را در برنامه داشته باشیم که از آن در قسمت های دیگر استفاده کنیم.

برای این کار مقادیر را در متغیر ها ذخیره میکنیم.



```
1 var_int = 12
2 var_float = 12.5
3 var_str = "Hello"
4 var_bool = True
5 var_complex = 13 + 5j
```



Value Assignment

در پایتون نوع داده متغیر ها بصورت خودکار مقداردهی میشوند و همچنین تمامی متغیر های عادی ما تغییر پذیر هستند و حتی میتوان نوع داده متغیر را هم تغییر داد.

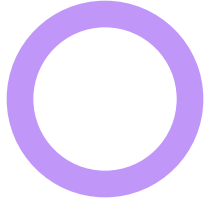



```
1 var = 45
2 var = 55.6
3 var = True
4 var = 12 - 3j
```







Assignment Operators

فرض کنیم دو متغیر a و b داریم که می‌خواهیم متغیر a را برای مثال با b جمع کنیم و جایگزین مقدار قبلی a کنیم. برای این کار دو روش داریم.



```
1 a = 78
2 b = 12
3
4 a = a + b
5 print(a)
6 OUT => 90
```



```
1 a = 78
2 b = 12
3
4 a += b
5 print(a)
6 OUT => 90
```

Assignment Operators

- `+=`
- `-=`
- `*=`
- `/=`
- `%=`
- `//=`
- `**=`

DocString

در پایتون میتوانیم برای نوشتن متن های عموماً طولانی و چند خطی از DocString استفاده میکنیم که با جفت `"""` یا `'''` استفاده میکنیم.



```
1 str1 = """
2 Hey Im DocString!
3 Blah Blah Blah....
4 """
5 print (str1)
6 OUT => Hey Im DocString!
7 Blah Blah Blah....
```



```
1 str1 = '''
2 Hey Im DocString!
3 Blah Blah Blah....
4 '''
5 print (str1)
6 OUT => Hey Im DocString!
7 Blah Blah Blah....
```

Commnets

در پایتون میتوانیم برای کد خود کامنت اضافه کنیم که صرفاً جهت اطلاعات اضافی برای کد میباشد و در فرآیند کامپایل شدن اثری ندارد.

دو نوع کامنت داریم:

(1) تک خطی: از کارکتر `#` برای نوشتن کامنت خود استفاده میکنیم.

(2) چند خطی: از `DocString` استفاده میکنیم.





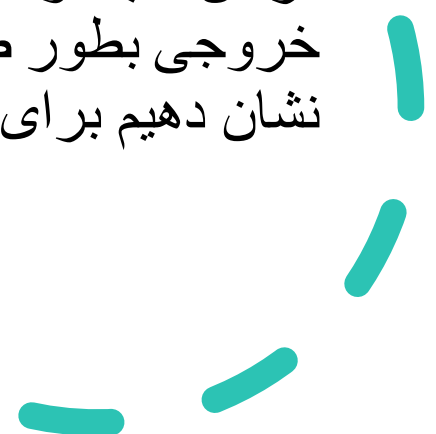
```
1 # This is a variable with 1 assigned to it
2 x = 1
3 """
4 Now we are doing sth different
5 like assigning increase it by 10
6 """
7 x += 10
```



```
1 OUT => a + b = 66
```

FStrings

فرض کنید دو متغیر داریم که از نوع `int` هستند و می‌خواهیم در خروجی بطور صراحتاً جمع این دو متغیر را بصورت ریاضی نشان دهیم برای مثال:



FStrings



```
1 a = 12
2 b = 13
3
4 print(f"a + b = {a + b}")
5 OUT => a + b = 25
```

برای این کار از FString استفاده میکنیم تا بتوانیم مقداری پردازش شده را داخل str خود جای بدهیم. برای این کار قبل 'یا' خود f را قرار میدهیم تا مشخص کنیم str ما از نوع FString است و عملیات قابل پردازشمان را داخل {} قرار میدهیم.

برای مثال:





1
2 Session One Done!

3 To Be Continued...

4 "" "" ""