

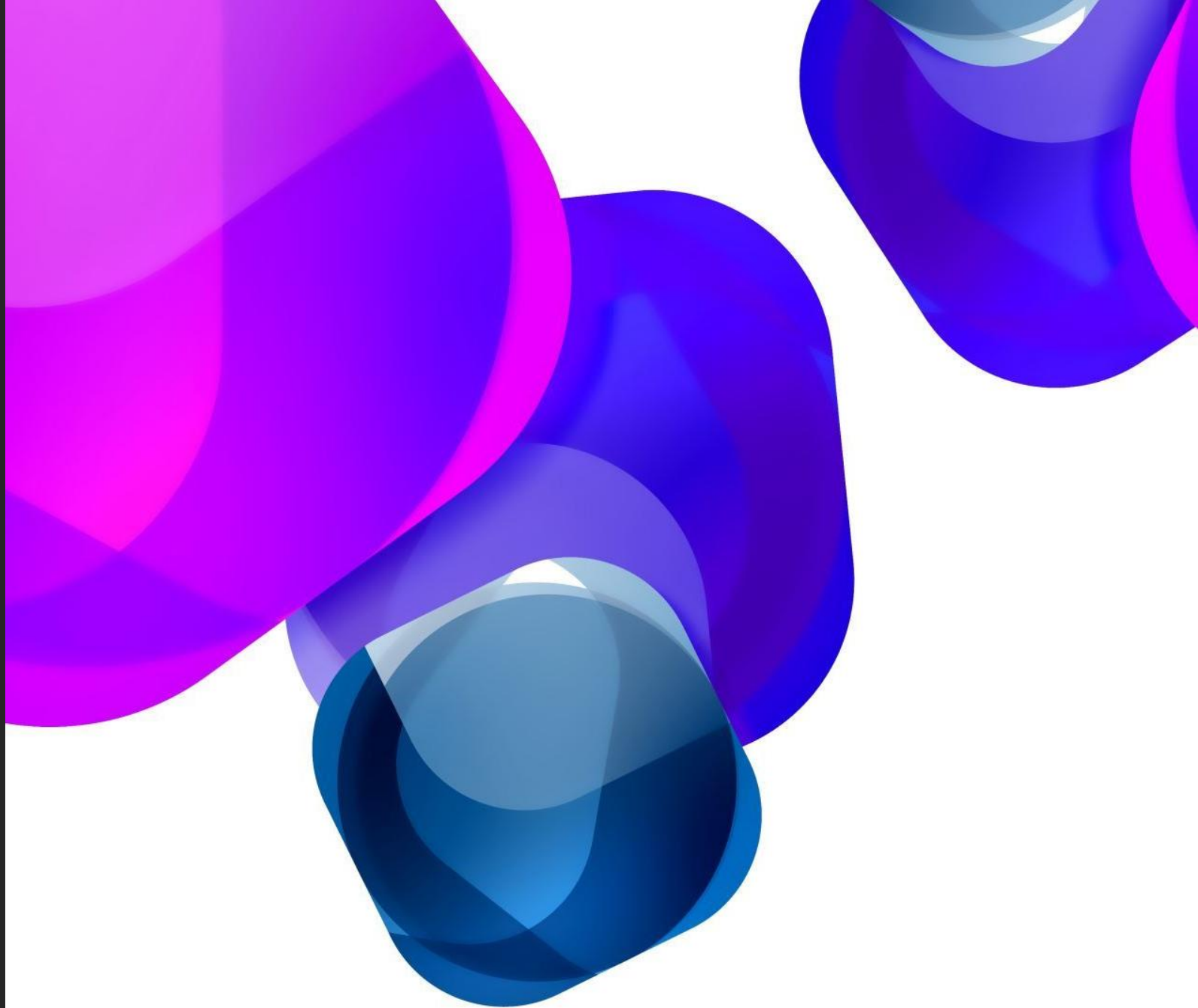
# پایتون جامع

---

محمد حسین مقدم خواه

[MHMK.NOVA@PROTON.ME](mailto:MHMK.NOVA@PROTON.ME)

جلسه دوم





# Comparison Operators

```
1 print(10 > 2)
2 OUT => True
3
4 print("Hello" == 'Hello')
5 OUT => True
```

`==` `=` Is Equal

`!=` `=` Is Not Equal

`>` `=` Bigger Than

`<` `=` Smaller Than

`>=` `=` Equal Or Bigger Than

`<=` `=` Equal Or Smaller Than

# Logical Operators

---

برای ترکیب عملگرهای مقایسه‌گر استفاده میشوند که به شرح زیر اند:

and: تا زمانی که همه شرط‌های ما True نباشند، False برمیگرداند.

or: حتی اگر یک شرط هم True باشد، مقدار True را برمیگرداند.

not: مقدار شرط را برعکس میکند.



```
1 print(10 > 2 and 8 >= 8)
2 OUT => True
3
4 print(not "Hello" == 'Hello')
5 OUT => False
```

# Getting User Input

---

در پایتون برای گفتن ورودی از کاربر، از تابع `input()` استفاده میکنیم.

این تابع به عنوان ورودی میتواند رشته ای جهت نمایش به کاربر برای گرفتن ورودی استفاده کرد.

همچنین این تابع همواره ورودی کاربر را بصورت `str` به ما برمیگرداند.

```
1 usr_input = input("Enter a number: ")
2 print(usr_input)
3
4 OUT => Enter a number:
5 IN  => 32
6 OUT => 32
```

# Casting

در پایتون میتوانیم با شرایط خاص نوع داده ها را تغییر بدهیم.

برای این کار توابع زیر وجود دارند:

`str()` هر نوع مقداری را میتواند به عنوان ورودی بگیرد و با نوع `str` برگرداند.

`int()` فقط مقادیر عددی را به عنوان ورودی میگیرد و مقدار صحیح آن عدد را برمیگرداند.

`float()` فقط مقادیر عددی را به عنوان ورودی میگیرد و مقدار اعشاری آن عدد را برمیگرداند.

`bool()` هر نوع مقداری را میگیرد و معادل منطقی آنرا برمیگرداند.



```
1 print(int("32.45"))
```

```
2 print(float(88))
```

```
3 print(bool(""))
```

```
4
```

```
5 OUT => 32
```

```
6 OUT => 88.0
```

```
7 OUT => False
```

# Escape Character

---

در رشته های پایتون، بک اسلش "\" یک کاراکتر خاص است که به آن کاراکتر "escape" نیز می گویند. برای نشان دادن برخی از کاراکترهای فضای سفید استفاده می شود "t\" : یک تب و "n\" یک خط جدید است.

همچنین برای جلوگیری از رندر شدن کوتیشن ها به عنوان پایان دهنده یا آغاز کننده رشته نیز استفاده میشود.

```
1 print("He said: \"Im ready for lunch!\")  
2  
3 OUT => He said: "Im ready for lunch!"
```



```
1 print("Hello\nWorld!")
2 print("Im\tPython!")
3
4 OUT => Hello
5 World!
6 OUT=> Im      Python!
```

# Special Characters

---

همانطور که در اسلاید قبل ذکر شد دو مورد از کارکترهای خاص در پایتون عبارتند از:

\n: برای ایجاد خط جدید استفاده میشود.

\t: برای ایجاد یک تب (فاصله 8 اسپیس) استفاده میشود.



# String Operations

---

بر روی رشته ها میتوان دو نوع عملیات را انجام داد:

Concatenation: اتصال دو رشته بوسیله جمع کردن آنها با هم.

Exponentiation: تکرار یک رشته بوسیله ضرب آن در عدد صحیح.

```
1 print("Hello" + " " + "World!")
2 print("Hey" * 3)
3
4 OUT => Hello World!
5 OUT => Hey Hey Hey
```


# Len() Function

---

از تابع len() برای مشخص کردن طول  
دیتا تایپ هایی مثل str استفاده  
میکنیم.



```
1 str_1 = "Hello World!"  
2  
3 print(len(str_1))  
4 OUT => 12
```



```
1 str_1 = "Hello World!"
2 sub_str = "llo"
3
4 print(sub_str in str_1)
5 OUT => True
6
```

## In Operator

---

برای چک کردن اینکه مثلاً مقداری رشته ای در رشته اصلی ما موجود دارد یا نه از این عملگر استفاده میکنیم.

# Upper() & Lower() Methods

---

در پایتون برای کوچک کردن همه حروف رشته از متد lower() و برای بزرگ کردن همه حروف از متد upper() استفاده میکنیم.

```
1 str_1 = "HELLO"
2 str_2 = "world!"
3
4 print(str_1.lower() + " " + str_2.upper())
5 OUT => hello WORLD!
```

# Strip() Method

---

گاهی اوقات رشته ای که میخواهیم با آن کار کنیم، در اول یا آخر یا هر دو آن، به اصطلاح WhiteSpace یا فضای خالی داریم. برای حذف کردن این فضای خالی از متد strip() بر روی رشته مورد نظرم استفاده میکنیم.

```
1 str_1 = "          HELLO          "|
2
3 print(str_1.strip())
4 OUT => HELLO |
```

# Replcae() Method

---

برای تغییر قسمت مشخصی از رشته مان با مقدار رشته ای دیگر استفاده میشود.

اولین ورودی این متد، رشته مدنظرمان است و بعدی مقداری که میخواهیم با آن عوض کنیم.

```
1 str_1 = "hello world!"  
2  
3 print(str_1.replace("h", "H"))  
4 OUT => Hello world!
```

Session Two  
Ended

To be continued...