

فصل پنجم

تعاملی کردن

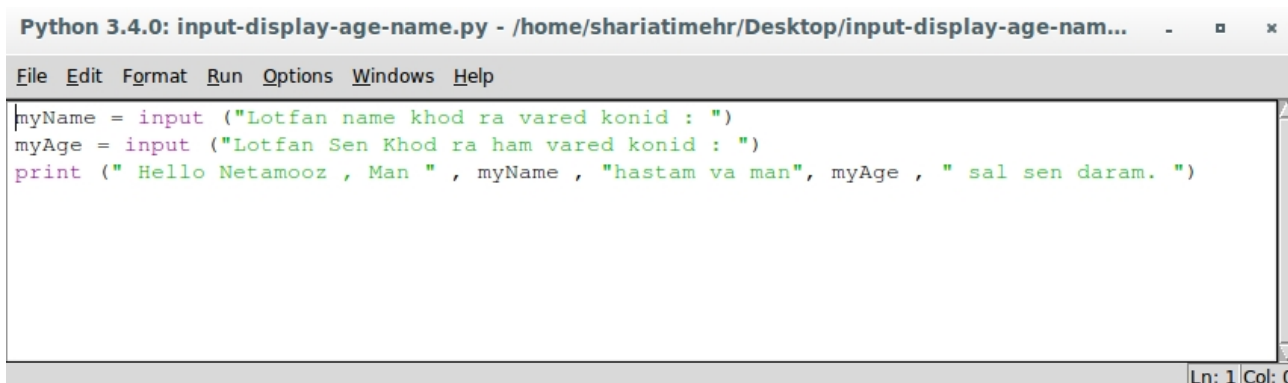
برنامه خود در پایتون



تعاملی کردن برنامه خودتان

حال که مقدمات متغیرها را یاد گرفتیم زمان آن رسیده که برنامه‌های کاربردی‌تر ایجاد کنیم. در این بخش همان برنامه `hello world` که قبلاً ساختیم را استفاده کرده ولی این بار به شکلی تعاملی. به جای اینکه فقط پیام `Hello World` را چاپ کنیم، این بار کاری می‌کنیم که برنامه ما اسامی و سن افراد را نیز بداند. به این منظور برنامه ما بایستی قادر به دریافت اطلاعات کاربر باشد و آن‌ها را در صفحه نمایش نشان دهد. دو تابع درون ساخت وجود دارند که این کار را برای ما انجام می‌دهند. توابع `input` و `print`. خوب به این منظور به سادگی برنامه را درون `IDLE` نوشته و اجرا کنیم.

```
myName = input ("Lotfan name khod ra vared
konid : ")
myAge = input ("Lotfan Sen Khod ra ham vared
konid : ")
print (" Hello Netamooz , Man " , myName ,
" hastam va man" , myAge , " sal sen daram. ")
```



The screenshot shows a Python 3.4.0 IDLE window titled "Python 3.4.0: input-display-age-name.py - /home/shariatimehr/Desktop/input-display-age-nam...". The window contains the following code:

```
myName = input ("Lotfan name khod ra vared konid : ")
myAge = input ("Lotfan Sen Khod ra ham vared konid : ")
print (" Hello Netamooz , Man " , myName , "hastam va man", myAge , " sal sen daram. ")
```

The status bar at the bottom right of the window shows "Ln: 1 Col: 0".



همانطور که ملاحظه می کنید ، پس از اجرای برنامه ، ابتدا نام شما پرسیده می شود و سپس سن شما و در نهایت هر دو را در قالب یک جمله چاپ می کند .

```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.0 (default, Jun 19 2015, 14:20:21)
[GCC 4.8.2] on linux
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Lotfan name khod ra vared konid : Mohammad ShariatiMehR
Lotfan Sen Khod ra ham vared konid : 28
Hello Netamooz , Man Mohammad ShariatiMehR hastam va man 28 sal sen daram.
>>> |
```

Ln: 9 Col: 4



تابع input در پایتون

در مثال قبلی ما از تابع input دو بار به منظور دریافت نام و سن کاربر برای نمایش در برنامه استفاده کردیم.

```
MyName = input (' Lotfan Name khod ra vared konid: ')
```

رشته " Lotfan Name khod ra vared konid " پیامی است که در صفحه نمایش نشان داده می‌شود و به کاربر توضیح می‌دهد که چه چیزی را بایستی وارد کند. پس از آنکه کاربر ورودی‌های خود را درون input وارد کرد، این اطلاعات به صورت رشته درون متغیر myName ذخیره می‌شوند. فیلد input بعدی که از کاربر خواسته می‌شود سن وی را از او می‌خواهد و آن را پس از دریافت درون متغیر myAge ذخیره می‌کند.

کمی در دو نسخه ۲ و ۳ پایتون تفاوت دارد. در پایتون ۲ اگر که بخواهید input تابع استفاده raw_input ورودی کاربر را به صورت رشته ای قبول کنید، بایستی از تابع input استفاده کنید.



تابع print در پایتون

تابع print به منظور نشان دادن اطلاعات کاربر در صفحه نمایش استفاده می‌شود. این تابع هیچ یا چند عبارت را به عنوان پارامتر قبول می‌کند و آن‌ها را با کاما از هم جدا می‌کند.

در عبارت زیر ما ۵ پارامتر را به تابع print می‌دهیم. سعی کنید آن‌ها را تشخیص دهید :

```
print ('Hello World , Esme man hast' ,  
myName, ' va man ' , myAge, 'sal sen  
daram.')
```

این پارامترها به صورت زیر می‌باشند :

```
'Hello World , Esme man hast'  
myName  
' va man '  
myAge  
'sal sen daram.'
```

این پارامترها شامل دو متغیر و سه رشته می‌باشند.

زمانی که رشته‌ها را وارد می‌کنیم از دابل کوتیشن استفاده کرده و زمان وارد کردن

متغیرها نیاز به قرار دادن کوتیشن نیست. نتیجه آن می‌شود که مقادیر متغیرها

جایگذاری شده و در خروجی چاپ می‌شود. شیوه‌ای دیگر برای چاپ کردن متغیرها

استفاده از فرمت دهنده % می‌باشد که در درس‌های قبلی یاد گرفتیم. برای نوشتن مثال

بالا با فرمت دهنده % به صورت زیر عمل می‌کنیم :



```
print ('Hello World , Esme man hast %s va  
man %s sal sen daram' % (myName, myAge) )
```

همچنین برای نوشتن همان عبارت با استفاده از متد `format` به صورت زیر عمل می‌کنیم:

```
print ('Hello World , Esme man hast {} va  
man {} sal sen daram'.format(myName, myAge)  
)
```

تابع `print` هم از دیگر توابعی است که در نسخه های ۳ و ۲ پایتون تفاوت دارد. در پایتون نسخه ۲ تابع `print` را بایستی بدون پرانتز و به صورت زیر نوشت (به جای کاما از بعلاوه استفاده می‌کنیم).

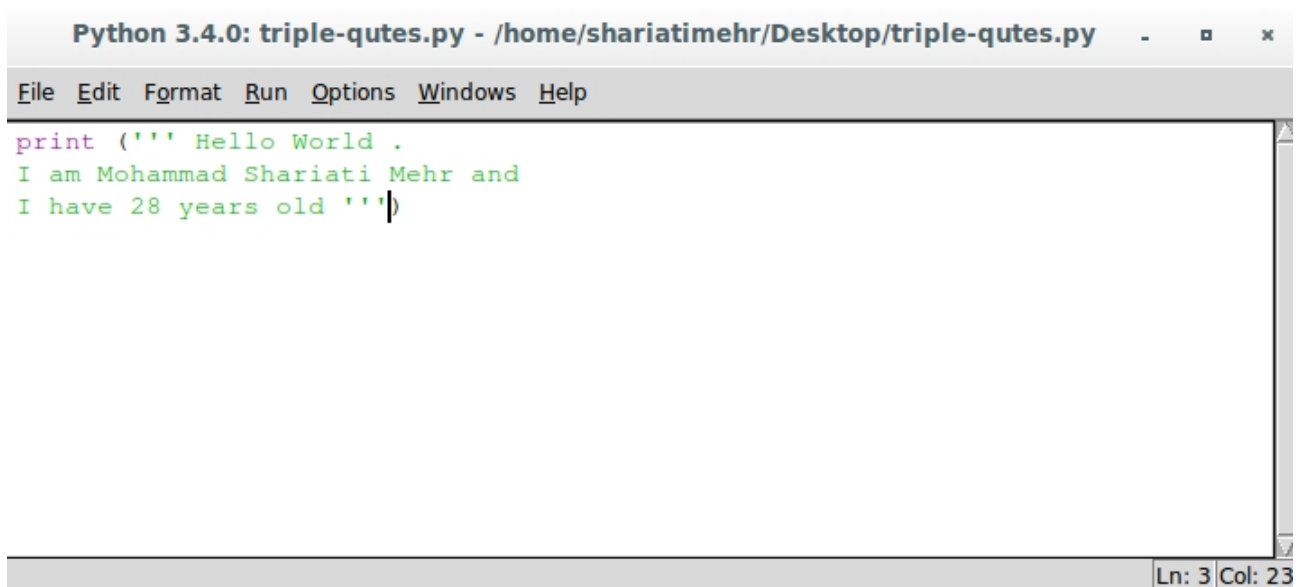
```
print 'Hello World , Esme man hast' +  
myName + ' va man ' + myAge + 'sal sen  
daram'
```



کوئیشن سه تایی

اگر که می خواهید یک پیام طولانی را نمایش دهید می توانید از کاراکترهای سه کوئیشن استفاده کنید تا پیام را در چندین خط نمایش دهید. برای نمونه :

```
print (''' Hello World .  
I am Mohammad Shariati Mehr and  
I have 28 years old ''')
```



The screenshot shows a Python 3.4.0 IDE window titled "Python 3.4.0: triple-quotes.py - /home/shariatimehr/Desktop/triple-quotes.py". The window contains a menu bar with "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The main area displays the following Python code:

```
print (''' Hello World .  
I am Mohammad Shariati Mehr and  
I have 28 years old ''')
```

The status bar at the bottom right indicates "Ln: 3 Col: 23".

که خروجی آن به صورت زیر در چند خط جداگانه چاپ می شود. این کار موجب می شود که خوانایی متن خود را افزایش دهید.



```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.0 (default, Jun 19 2015, 14:20:21)
[GCC 4.8.2] on linux
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
=====
>>>
    Hello World .
I am Mohammad Shariati Mehr and
I have 28 years old
>>> |
```

Ln: 9 Col: 4



عبور از کاراکترها

برخی اوقات شاید لازم باشد که برخی کاراکترهای چاپ نشدنی را مثل کاراکتر تب یا خط جدید را چاپ کنیم. در این شرایط نیاز دارید که از کاراکترهای عبور مثل بک اسلش (\) استفاده کنیم.

برای نمونه برای چاپ یک tab به این صورت می نویسیم \t در صورتی که قبل از t یک اسلش نگذاریم خود کاراکتر t چاپ می شود. برای چاپ تب به صورت زیر عمل می کنیم :

```
print('Hello \t World')
```

دیگر موارد استفاده از کاراکتر بک اسلش به صورت زیر می باشد :

\n که یک خط جدید را چاپ می کند

```
print ('Hello \n World')
```

خروجی آن

```
Hello  
World
```

\\ خود کاراکتر بک اسلش را چاپ می کند

```
print ('\\')
```

خروجی آن

```
\
```



دابل کوتیشن را در شرایطی چاپ می کند که با کوتیشن های دیگر تداخلی ایجاد نکند \"

```
print (''I am 5'9\" tall'')
```

خروجی آن

```
I am 5'9'' tall
```

اگر که نمی خواهیم کاراکتر \ به عنوان یک کاراکتر ویژه تفسیر شود ، می توانید قبل از آن r را اضافه کرده که به صورت raw نمایش داده شود . برای نمونه اگر نمی خواهید t به عنوان یک کاراکتر تب تفسیر شود ، بایستی به صورت زیر بنویسید :

```
print (r 'Hello \t World')
```

در نتیجه آن خروجی زیر بدست می آید یعنی کاراکتر t به صورت raw خام چاپ می شود :

```
Hello \t World
```



فصل شش

ایجاد انتخاب

و تصمیم



ایجاد انتخاب و تصمیم

در این فصل تلاش می‌کنیم تا برنامه خود را هوشمندانه‌تر کنیم و قابلیت تصمیم‌گیری و انتخاب را به آن اضافه کنیم. این قابلیت‌ها را با استفاده از عبارات `while` و `for`، `if` و `while` ایجاد می‌کنیم. این‌ها را ابزارهای کنترل جریان برنامه می‌نامند. این ابزارها جریان برنامه را کنترل می‌کنند.

بعلاوه عبارات `try`، `except` را بررسی می‌کنیم. این عبارات تشخیص می‌دهند که هنگام بروز خطا در برنامه چه اتفاقی بایستی رخ دهد.



عبارات شرطی

همه ابزارهای کنترل جریان شامل یک عبارت شرطی برای سنجش هستند. در این حالت برنامه شما در شرایط مختلف عمل کرد متفاوتی خواهد داشت.

رایج ترین عبارت شرطی، عبارت مقایسه می باشد. اگر که بخواهیم برابر بودن دو متغیر را مقایسه کنیم، از علامت `==` برای مقایسه استفاده می کنیم. مثلاً اگر بنویسیم `x == y` در حقیقت دارید از برنامه می پرسید که آیا `x` با `y` برابر است یا خیر. اگر که برابر باشند شرط برقرار می شود و عبارت خروجی `True` را بر می گرداند و اگر برابر نباشند شرط برقرار نشده و خروجی `False` خواهد بود.

دیگر علامت های مقایسه عبارت اند از `!=` (مقایسه عدم برابری)، `<` (مقایسه کوچکتر بودن)، `>` (مقایسه بزرگتر بودن)، `<=` (مقایسه کوچکتر یا برابری)، `>=` (مقایسه بزرگتر یا برابری).

در زیر لیستی از نحوه استفاده این علامت ها مثال هایی را برای درک بهتر نشان می دهیم :

عدم برابری

`5 != 2`

بزرگتر بودن

`5 > 2`

کوچکتر بودن

`2 < 5`



بزرگ‌تر یا مساوی بودن

$5 >= 2$

$5 >= 5$

کوچکتر یا مساوی بودن

$2 <= 5$

$2 <= 2$

علاوه بر علامت‌های ذکر شده، عملگرهای (و and، یا or، نفی not) در شرط‌های ترکیبی مفید هستند.

عملگر and فقط در صورتی که همه شرط‌ها برقرار باشند خروجی True باز می‌گرداند در غیر اینصورت خروجی False خواهد بود. برای مثال:

$5 == 5 \text{ and } 2 > 1$

خروجی true را باز می‌گرداند چرا که هر دو شرط برقرار است.

عملگر or تنها در صورتی که یکی از شرط‌ها برقرار باشد true باز می‌گرداند. در غیر اینصورت خروجی false باز می‌گرداند. برای مثال:

$5 > 2 \text{ or } 7 > 10 \text{ or } 3 == 2$

خروجی true باز می‌گرداند چرا که حداقل عبارت $5 > 2$ برقرار است.

عملگر not در صورتی که شرط پس از not برقرار نباشد، خروجی true را باز می‌گرداند. در غیر اینصورت خروجی false باز می‌گرداند. برای مثال:

$\text{not } 2 > 5$

خروجی true را باز می‌گرداند چرا که شرط $2 > 5$ برقرار نیست و ۲ بزرگ‌تر از ۵ نیست.



عبارت شرطی if

عبارت شرطی if یکی از رایج ترین عبارات شرطی برای کنترل جریان برنامه می باشد .
این عبارت برنامه را قادر می سازد تا وقتی برخی شرط ها بوجود آمد ، بر اساس شرط مورد نظر و نتیجه بدست آمده عکس العمل مناسب را انجام دهد . ساختار دستوری عبارت if به صورت زیر می باشد :

: شرط شماره ۱ برقرار شد if

do A

: شرط شماره ۲ برقرار شد elif

do B

: شرط شماره ۳ برقرار شد elif

do C

: شرط شماره ۴ برقرار شد elif

do D

else:

do E

عبارت elif همان else if می باشد به معنی در غیر اینصورت می باشد . در زبان های برنامه نویسی دیگر مثل C یا جاوا ساختار آن به شکل دیگری است و پرانتز وجود دارد یا براکت . در حالی که در پایتون نیازی به پرانتز یا براکت بعد از if یا elif یا else نیست . پایتون از براکت استفاده نمی کند . به جای آن برای شروع یا پایان عبارت شرطی از تورفتگی استفاده می کند .

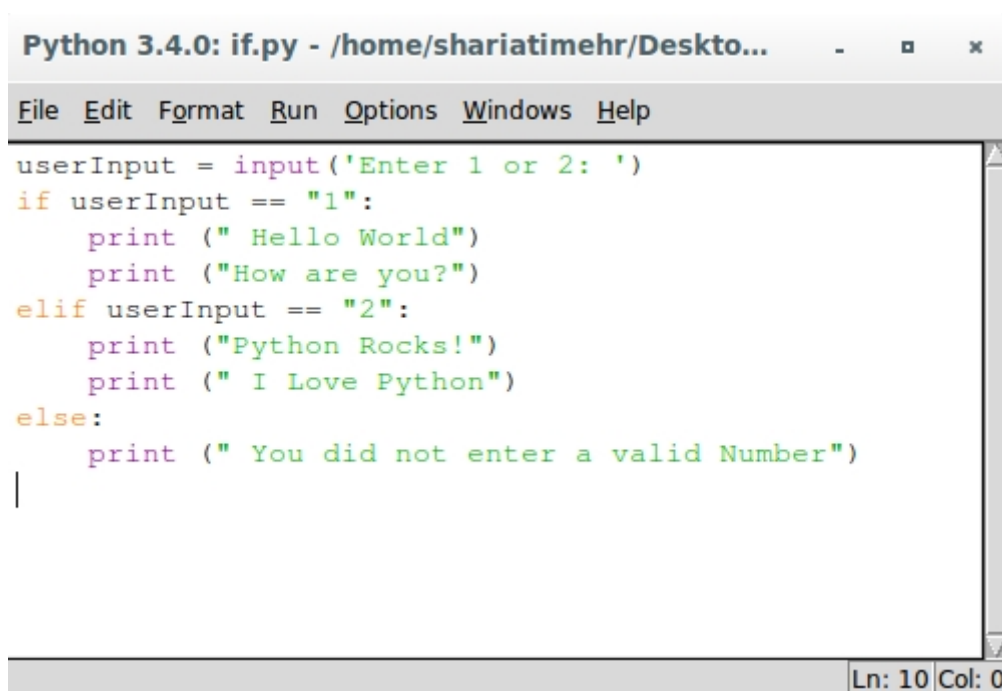
هر چیزی که دارای تورفتگی (Indentation) باشد ، به عنوان یک بلاک کد در نظر گرفته می شود که در صورت برقرار بودن شرط بلاک کد اجرا می شود .



به منظور ایجاد تورفتگی های صحیح و عدم بروز خطا در کدنویسی توصیه می شود از یک ادیتور مناسب پایتون مثل IDLE استفاده کنید .

برای اینکه درک درستی از نحوه اجرای شرط if پیدا کنید ، IDLE را باز کنید و کد زیر را درون آن اجرا کنید :

```
userInput = input('Enter 1 or 2: ')
if userInput == "1":
    print (" Hello World")
    print ("How are you?")
elif userInput == "2":
    print ("Python Rocks!")
    print (" I Love Python")
else:
    print (" You did not enter a valid
Number")
```



The screenshot shows a window titled "Python 3.4.0: if.py - /home/shariatimehr/Deskto...". The window contains a menu bar with "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The code is displayed in a monospaced font with syntax highlighting: keywords are in orange, strings in green, and print statements in purple. The code is identical to the one in the previous block. At the bottom right of the window, the status bar shows "Ln: 10 Col: 0".



برنامه ابتدا از کاربر می‌خواهد که یکی از اعداد ۱ یا ۲ را وارد کند. در صورتی که عدد ۱ را وارد کند برنامه یک خروجی را نشان می‌دهد و در صورتی که عدد ۲ را وارد کند خروجی متفاوت و در صورتی که عدد اشتباهی را وارد کند پیام خطایی را نشان می‌دهد.

```
Python 3.4.0 Shell - □ ×
File Edit Shell Debug Options Windows
Help
Python 3.4.0 (default, Jun 19 20
15, 14:20:21)
[GCC 4.8.2] on linux
Type "copyright", "credits" or "
license()" for more information.
>>> =====
==== RESTART =====
>>>
Enter 1 or 2: 2
Python Rocks!
  I Love Python
>>> |
```

Ln: 9 Col: 4



عبارت شرطی Inline If

عبارت شرطی inline if شکلی ساده‌تر از عبارت شرطی if می‌باشد و برای انجام وظایف ساده‌تر بسیار مناسب می‌باشد. ساختار دستور inline If به صورت زیر می‌باشد :

وظیفه شماره ۱ را انجام بده else شرط برقرار بود if وظیفه شماره ۱ را انجام بده

برای مثال :

```
num1 = 12 if myInt==10 else 13
```

این عبارت عدد ۱۲ را به متغیر num1 اختصاص داده اگر متغیر myInt برابر ۱۰ باشد در غیر اینصورت عدد ۱۳ را به num1 اختصاص می‌دهد. مثالی دیگر :

```
print ('' This is task A'' if myInt == 10  
else ''This is task B'')
```

این عبارت در صورتی که متغیر myInt برابر ۱۰ باشد عبارت This is task A را چاپ می‌کند و در غیر اینصورت عبارت This is task B را چاپ می‌کند.

