



در شماره گذشته آموزش پایتون با مفهوم آرگومان‌های ورودی توابع، مقداردهی اولیه آرگومان‌ها، مقدار بازگشتی از یک تابع و مقایسه مقادیر بازگشتی توابع آشنا شدیم. در این شماره قصد داریم به سراغ نحوه دریافت ورودی رفته و مباحث دیگر مرتبط با توابع را بررسی کنیم.

برای مطالعه بخش دوازدهم آموزش رایگان پایتون [اینجا](#) کلیک کنید

دریافت ورودی از کاربر

تعداد بسیار محدودی از برنامه‌های کاربردی در جهان وجود دارند که فارغ از تعامل با کاربر کار می‌کنند. در حقیقت اکثر برنامه‌های کاربردی برای اجرای هر چه بهتر با کاربر به تعامل می‌پردازند، زیرا کامپیوترها با هدف خدمت‌رسانی به مردم طراحی شده‌اند. برای ارتباط برقرار کردن با کاربر، یک برنامه کاربردی بر مبنای یکسری تکنیک‌های رایج مقادیر را از ورودی دریافت می‌کند. در دنیای پایتون تابع **input** از جمله توابعی است که برای دریافت ورودی از کاربر استفاده می‌شود. دقت کنید که تابع **Input** همیشه یک خروجی رشته‌ای بر می‌گرداند. به عبارت دیگر، حتی زمانی که کاربر یک مقدار عددی را تایپ کند خروجی این تابع باز هم از نوع رشته‌ای خواهد بود. این حرف به معنای آن است که اگر شما در نظر دارید از تابع فوق یک مقدار عددی دریافت کنید، باید از توابع تبدیل دهنده استفاده کنید. تابع **Input** یک تابع انعطاف‌پذیر است، زیرا به شما اجازه می‌دهد متنی به کاربر نشان تا کاربر بتواند مقدار درستی را وارد کند. برای آشنایی بیشتر با این تابع بهتر است آن را به شکل عملی بررسی کنیم.

1. محیط IDLE را باز کرده از منوی File گزینه New File را انتخاب کنید.

2. در پنجره شل دستورات زیر را وارد کنید.

```
Name = input("Tell me your name: ")
```

```
print("Hello ", Name)
```

3. فایل را به نام Myinput ذخیره کرده و از منوی Run گزینه Run Module را انتخاب کنید. خروجی این قطعه کد همانند تصویر زیر است.

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.191
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information
>>>
===== RESTART: C:/Node/Myinput.py =====
Tell me your name: Hamid Reza Taebi
Hello Hamid Reza Taebi
>>>
```

4. در قطعه کد بالا، تابع `input` از کاربر نامی درخواست می‌کند. پس از آن که کاربر نام را وارد کرده و کلید اینتر را فشار داد، مقدار وارد شده از سوی کاربر را همراه با رشته متنی که درون تابع `Print` قرار دارد در خروجی چاپ می‌کند.

شما می‌توانید از تابع `input` برای دریافت هر نوع داده‌ای استفاده کنید. اگر قصد دارید از تابع `input` برای دریافت نوع‌های داده‌ای عددی استفاده کنید در زمان به‌کارگیری تابع باید حتماً از تبدیل‌کننده خاص آن نوع عددی استفاده کنید.

1. در محیط IDLE روی گزینه `File` و سپس `New File` کلیک کنید.

2. قطعه کد زیر را در محیط شل تایپ کرده و فایل را با نامی دلخواه ذخیره کنید.

```
ANumber = float(input("Type a number: "))
```

```
print("You typed: ", ANumber)
```

3. از منوی `Run` گزینه `Run Module` را انتخاب کنید تا برنامه اجرا شود.

4. در قطعه کد بالا، ما قصد داریم یک مقدار شناور (اعشاری) را از کاربر دریافت کنیم. اما به دلیل این که تابع `input` همواره یک مقدار رشته‌ای بر می‌گرداند، مجبور هستیم از تبدیل‌کننده `float` برای تبدیل ورودی به یک مقدار اعشاری استفاده کنیم. خروجی قطعه کد بالا را در تصویر زیر مشاهده می‌کنید.

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Type "help", "copyright", "credits" or "license()" for more informatio
>>>
===== RESTART: C:/Node/Myinput.py =====
Tell me your name: Hamid Reza Taebi
Hello Hamid Reza Taebi
>>>
===== RESTART: C:/Node/Myinput.py =====
Type a number: 123.23
You typed: 123.23
>>> |
```

5. همان‌گونه که در تصویر مشاهده می‌کنید مقدار `123.23` به عنوان یک مقدار اعشاری نشان داده شده است. اگر در مثال بالا کمی بیشتر دقت کنید کاربرد دیگر تکنیک مقدار بازگشتی یک تابع که در شماره قبل آن را بررسی کردیم

را مشاهده می‌کنید. همان‌گونه که در مقاله قبل اشاره کردیم، مقدار بازگشتی یک تابع را می‌توان به یک متغیر یا عبارت تخصیص داد که در مثال بالا خروجی input به عنوان ورودی در اختیار تبدیل‌کننده float قرار گرفته است.

دقت کنید به‌کارگیری تبدیل‌کننده‌ها در زمان تبدیل یک مقدار به مقدار دیگر بدون ریسک نیست. به‌طور مثال اگر سعی کنید مقداری متفاوت از یک مقدار عددی را به عنوان ورودی وارد کرده و سپس از یک تبدیل‌کننده استفاده کنید با پیغام خطایی همانند شکل زیر روبرو می‌شوید.

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Type a number: 123.23
You typed: 123.23
>>>
===== RESTART: C:/Node/Myinput.py =====
Type a number: Hello
Traceback (most recent call last):
  File "C:/Node/Myinput.py", line 1, in <module>
    ANumber = float(input("Type a number: "))
ValueError: could not convert string to float: 'Hello'
>>> |
```

همان‌گونه که در تصویر بالا مشاهده می‌کنید، پیغام خطا اعلام می‌دارد که امکان تبدیل یک مقدار رشته‌ای (Hello) به یک مقدار float امکان‌پذیر نیست.

دستورات شرطی

تا این بخش از سری آموزش‌های **پایتون** یاد گرفتیم که متغیر و نوع داده‌ای چیست، چگونه نوع‌های داده‌ای را به یکدیگر تبدیل کنیم، توابع چه هستند و چگونه باید توابع پیشرفته‌ای را نوشته و از آن‌ها استفاده کرده و در نهایت با نحوه دریافت ورودی از کاربر آشنا شدیم. اما یکی دیگر از مباحث مهم دنیای برنامه‌نویسی تصمیم‌گیری است. توانایی تصمیم‌گیری به ما اجازه می‌دهد بر مبنای شرایط و نتایجی که به دست می‌آوریم بهترین مسیر را انتخاب کنیم. ریاضیات به کامپیوترها اجازه می‌دهند بر مبنای اطلاعاتی که به آن‌ها می‌دهیم بهترین تصمیم‌گیری را انجام دهند. بدون داشتن توانایی تصمیم‌گیری کامپیوترها به یک ابزار بدون استفاده تبدیل می‌شوند. بنابراین هر زبانی مجموعه دستورات کاربردی برای اخذ تصمیمات در اختیار برنامه‌نویسان قرار می‌دهد.

به موافقی فکر کنید که مجبور هستید تصمیمی را اتخاذ کنید. برای اخذ یک تصمیم در دنیای واقعی ارزش واقعی یک لحظه یا موجودیتی را می‌سنجید و بر مبنای آن ارزش کاری را انجام می‌دهید. به‌طور مثال، زمانی که به چراغ راهنمایی نگاه کرده و آن‌را قرمز مشاهده می‌کنید، نور قرمز را با نور سبز ارزیابی می‌کنید و سپس تصمیم می‌گیرید که نور چراغ سبز نیست و باید توقف کنید. اکثر مردم برای اخذ تصمیماتی مشابه با چراغ راهنمایی زمان زیادی را صرف نمی‌کنند، زیرا در طول روز بارها و بارها این‌کار را انجام می‌دهند. تصمیم‌گیری یکی از خصایص نهادینه شده در فطرت انسان‌ها است، در حالی که باید به یک کامپیوتر برای اخذ هرگونه تصمیمی کمک کنید. کامپیوترها برای اخذ یک تصمیم مراحل زیر را انجام می‌دهند.

1. مقدار جاری یا واقعی چیزی را به دست می‌آورند.

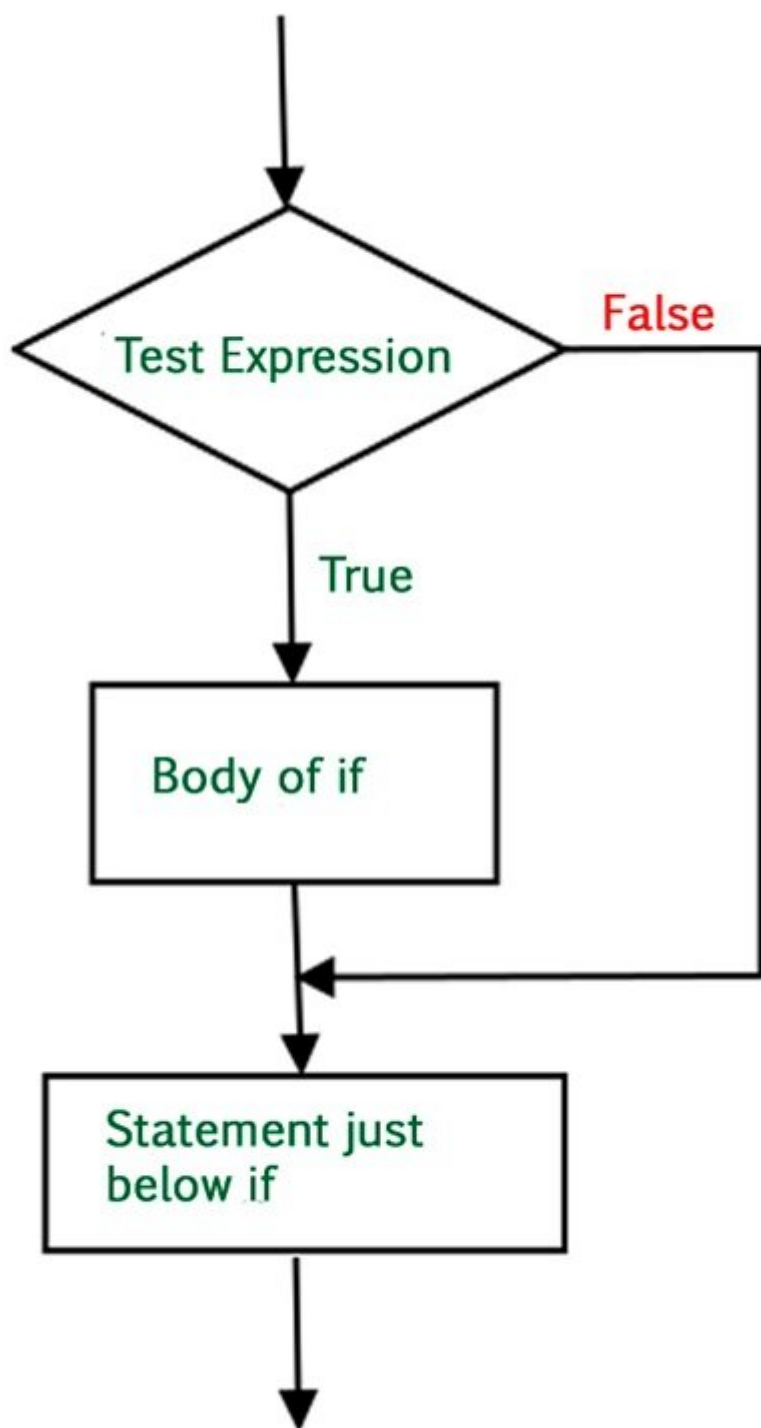
2. مقدار جاری یا واقعی را با مقدار دلخواه/از پیش تعیین شده مقایسه می‌کنند.

3. عملی را متناظر با نتیجه‌ای که از ارزیابی به دست آورده‌اند انجام می‌دهند.

اتخاذ تصمیمات ساده با دستور if

دستور if یکی از ساده‌ترین فرمان‌های **پایتون** است که برای اخذ تصمیمات استفاده می‌شود. عملکرد دستور فوق به این شکل است که دو مقدار یا دو عبارت را ارزیابی کرده و اگر نتیجه حاصل از یک ارزیابی درست (true) باشد

دستوراتی که از سوی برنامه‌نویس مشخص شده‌اند را اجرا می‌کند. بد نیست بدانید که شما در دنیای واقعی هر روزه از دستور if استفاده می‌کنید. به‌طور مثال، شما ممکن است با خود بگویید اگر امروز چهارشنبه است، من یک ساندویچ سرد خواهم خورد. (از دید زبان‌شناسان از مضارع اخباری استفاده می‌کنید که معادل آینده در گذشته زبان انگلیسی است!) در زبان برنامه‌نویسی **پایتون** عملکرد دستور if یک مقدار متفاوت بوده، اما از الگو و ترکیب نحوی یکسانی پیروی می‌کند. **پایتون** به شما می‌گوید ابتدا یک متغیر ایجاد کنید، مقداری درون آن قرار داده و سپس متغیر خود را با مقداری که مدنظر دارید مقایسه کرده و دستوراتی را در بدنه دستور if قرار دهید تا اگر نتیجه ارزیابی درست بود آن دستورات اجرا شوند. فلوچارت زیر فرآیند اخذ تصمیمات بر مبنای دستور if را نشان می‌دهد.



به قطعه کد زیر دقت کنید:

```
Testme=6
```

```
If Testme==6
```

```
Print("TestMe does equal 6!")
```

کاری که در قطعه کد بالا انجام دادیم به این شرح است که از طریق عملگر = یک فرآیند انتصاب انجام داده و یک مقدار صحیح به متغیر Testme اختصاص دادیم. در مرحله بعد از کلمه کلیدی if استفاده کرده و مقدار متغیر Testme را با یک مقدار صحیح (6) ارزیابی کردیم. از آنجایی که حاصل مقایسه این دو مقدار برابر با 6 است فرمان print که در ادامه مشخص شده چاپ می‌شود.

ترکیب نحوی دستور if

زمانی که **پایتون** یک چنین ترکیبی را مشاهده می‌کند متوجه می‌شود که شما می‌خواهید تصمیمی را اتخاذ کنید. پس از کلمه کلید if یک شرط ظاهر می‌شود. یک شرط به سادگی نشان می‌دهد که شما در نظر دارید مقایسه‌ای انجام دهید. در مثال بالا شما از **پایتون** انتظار دارید که مقدار درون متغیر Testme را با مقدار 6 ارزیابی کند. دقت کنید در این قطعه کد ما از عملگر رابطه‌ای دو علامت مساوی == و نه عملگر انتصاب = استفاده کرده‌ایم. یک اشتباه رایجی که برنامه‌نویسان تازه‌کار انجام می‌دهند این است که در زمان ارزیابی دو مقدار در دستور if از عملگر انتصاب = استفاده می‌کنند که باعث می‌شود **پایتون** دچار مشکل شود. زمانی که از دو نقطه : استفاده می‌کنید به معنای آن است که فرآیند ارزیابی یک مقدار به پایان رسیده است. اگر از دو نقطه استفاده نکنید، **پایتون** متوجه نمی‌شود که شرط به پایان رسیده و به دنبال آن است تا شرایط دیگری را نیز بررسی کند. پس از دو نقطه باید دستوراتی که قرار است پس از برقراری شرط اجرا شوند را تعیین کنید که در این مثال چاپ مقدار متغیر TestMe است. شما از دستور if **پایتون** به اشکال زیر استفاده می‌کنید:

از دستور if برای ارزیابی و درست بودن یک عبارت استفاده می‌کنید تا دستور بعد از if اجرا شود.

از دستور if برای ارزیابی و درست بودن یک عبارت استفاده می‌کنید تا دستورات بعد از if اجرا شوند.

چند عبارت را در دستور if ترکیب کرده و اگر حاصل ارزیابی عبارت‌ها درست بود، دستوراتی را برای اجرا شدن مشخص می‌کنید.

در شماره آینده آموزش **پایتون** این بحث را ادامه خواهیم داد.

تاریخ انتشار:

نشانی منبع:

<https://www.shabakeh-mag.com/workshop/programming/14632/%D8%A2%D9%85%D9%88%D8%B2%D8%B4-%D8%B1%D8%A7%DB%8C%DA%AF%D8%A7%D9%86-%D9%BE%D8%A7%DB%8C%D8%AA%D9%88%D9%86-python-%E2%80%93%D8%A2%D8%B4%D9%86%D8%A7%DB%8C%DB%8C-%D8%A8%D8%A7-%D9%86%D8%AD%D9%88%D9%87-%D8%AF%D8%B1%DB%8C%D8%A7%D9%81%D8%AA-%D9%88%D8%B1%D9%88%D8%AF%DB%8C-%D8%A7%D8%B2-%DA%A9%D8%A7%D8%B1%D8%A8%D8%B1%D8%8C-%D8%A7%D8%AA%D8%AE%D8%A7%D8%B0>