آموزش رایگان پایتون (python) - آشنایی با آرگومان‌های ورودی، مقدار بازگشتی توابع و مقدار پیش‌فرض آرگومان‌ها

در شماره گذشته آموزش پایتون با مفهوم توابع، ساخت و فراخوانی توابع آشنا شدیم و دیدیم که آرگومان‌ها باعث انعطاف‌پذیری توابع می‌شوند. در این شماره قصد داریم اطلاعات بیشتری در ارتباط با توابع به دست آوریم.

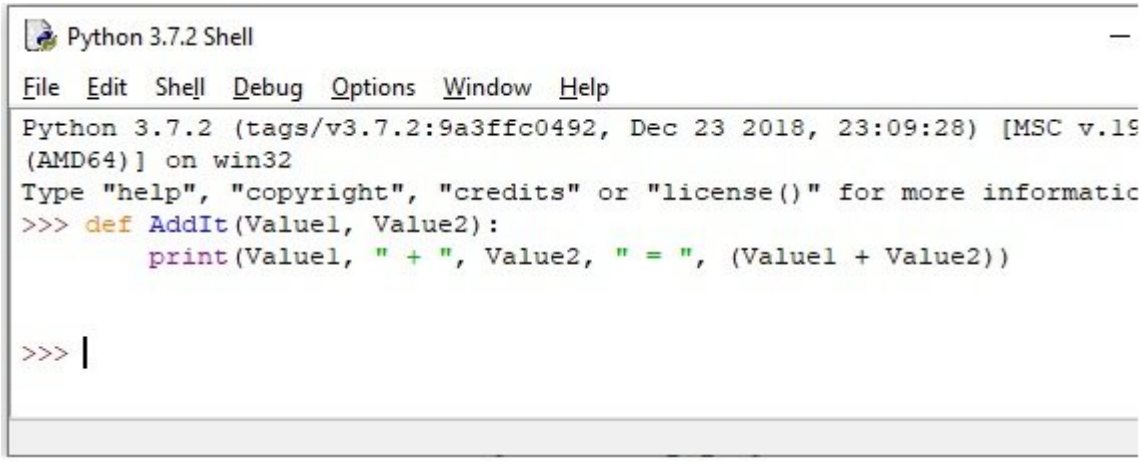
برای مطالعه بخش یازدهم آموزش رایگان پایتون اینجا کلیک کنید

با گذشت زمان توابعی که می‌نویسید پیچیده‌تر شده و متدهایی که از آن‌ها استفاده می‌کنید نیز حرفه‌ای‌تر می‌شوند، در چنین شرایطی باید کنترل بیشتری روی نحوه فراخوانی توابع اعمال کرده و آرگومان‌های درستی را برای توابع ارسال کنید. تا این مرحله، یاد گرفتید که آرگومان‌ها انعطاف‌پذیری را بیشتر کرده و به شما اجازه می‌دهند مقادیر مختلفی را برای پردازش به توابع ارسال کنید. اما این روش مشکلی دارد. شما در زمان فراخوانی توابعی که دارای آرگومان هستند مجبور هستید، در زمان فراخوانی تابع حتما مقداری را تعیین کنید تا تابع اجرا شوند. اما پایتون به شما اجازه می‌دهد در زمان تعریف آرگومان‌ها مقداری را به عنوان مقدار پیش‌فرض آرگومان مشخص کنید. در این حالت زمانی که قصد فراخوانی تابعی را دارید، فراخوانی تابع همراه با مقدار آرگومان اختیاری خواهد بود. به عبارت دقیق‌تر اگر برای یک تابع مقداری را مشخص نکرده و آن‌را فراخوانی کنید، پایتون مقدار پیش‌فرض را استفاده خواهد کرد. پایتون متدی دارد که اجازه می‌دهد آرگومان‌ها را از طریق کلیدواژه‌ها ارسال کنید. در این حالت شما در زمان تعریف تابعی که قرار است آرگومانی داشته باشد، آرگومان را همراه با یک علامت مساوی و مقداری که روبروی آن قرار می‌گیرد استفاده می‌کنید. ابتدا به مثال زیر دقت کنید:

IDLE.1 را اجرا کرده و تابع زیر را درون آن وارد کنید.

```
def AddIt(Value1, Value2):

print(Value1, " + ", Value2, " = ", (Value1 + Value2))
```

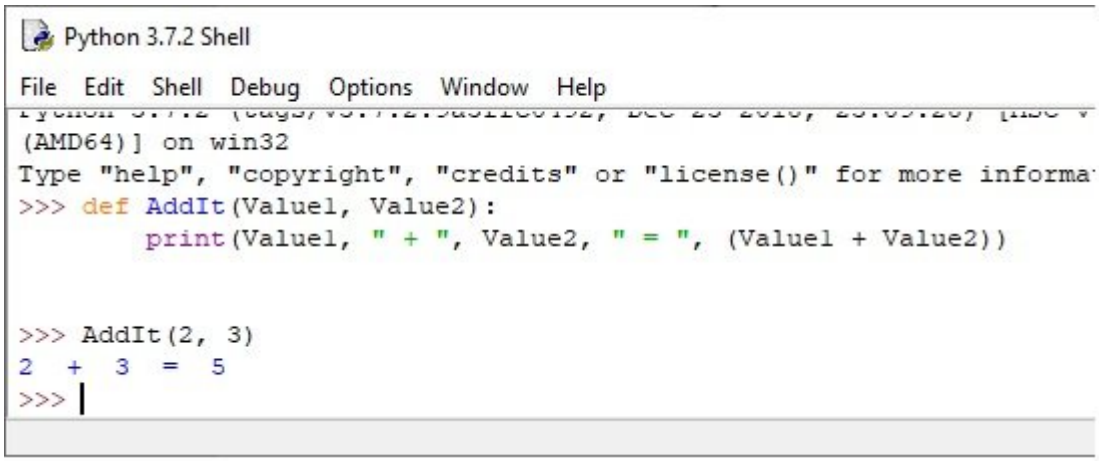
همان‌گونه که در مثال بالا مشاهده می‌کنید درون تابع AddIt فرمان print شامل فهرستی از آرگومان‌هایی است که با علامت کاما از یکدیگر جدا شده‌اند. آرگومان‌هایی که درون تابع print قرار دارند نوع‌های داده‌ای مختلفی دارند.



2. برای فراخوانی تابع فوق از فرمان زیر استفاده کنید.

```
AddIt(2, 3)
```

همان‌گونه که مشاهده می‌کنید تابع فوق عبارت 2+3=5 را در خروجی نشان می‌دهد.

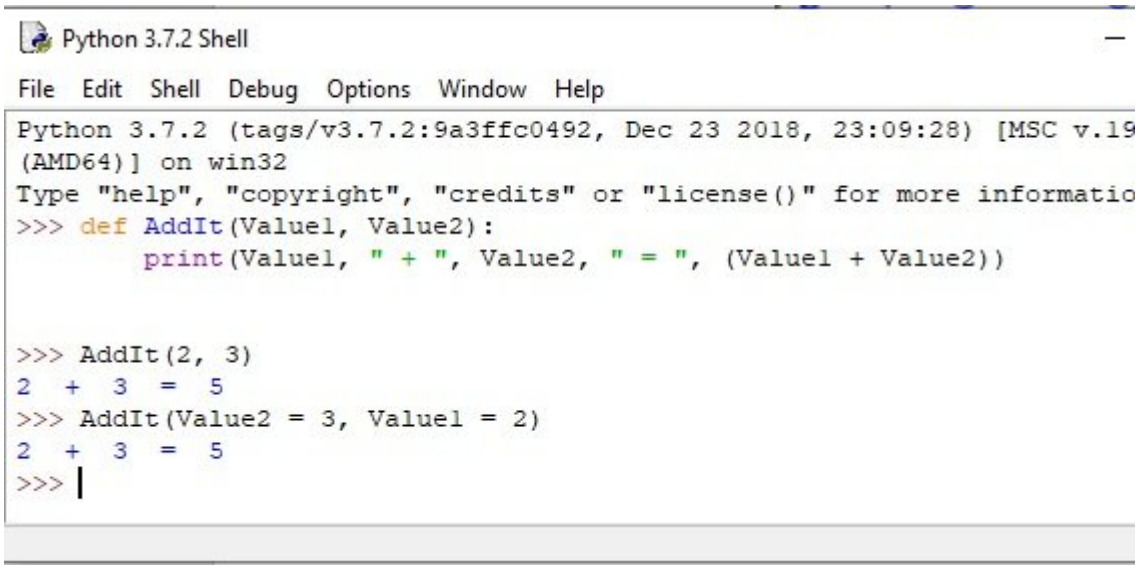


3. دقت کنید در زمان فراخوانی تابع دو مقدار را تعیین کرده‌اید، اما در فرمان print در آخرین بخش از طریق به‌کارگیری پرانتزها و علامت مثبت توانستید حاصل جمع دو مقدار را نیز چاپ کنید.

4. اکنون فرمان زیر را تایپ کرده و کلید اینتر را فشار دهید.

```
AddIt(Value2 = 3, Value1 = 2)
```

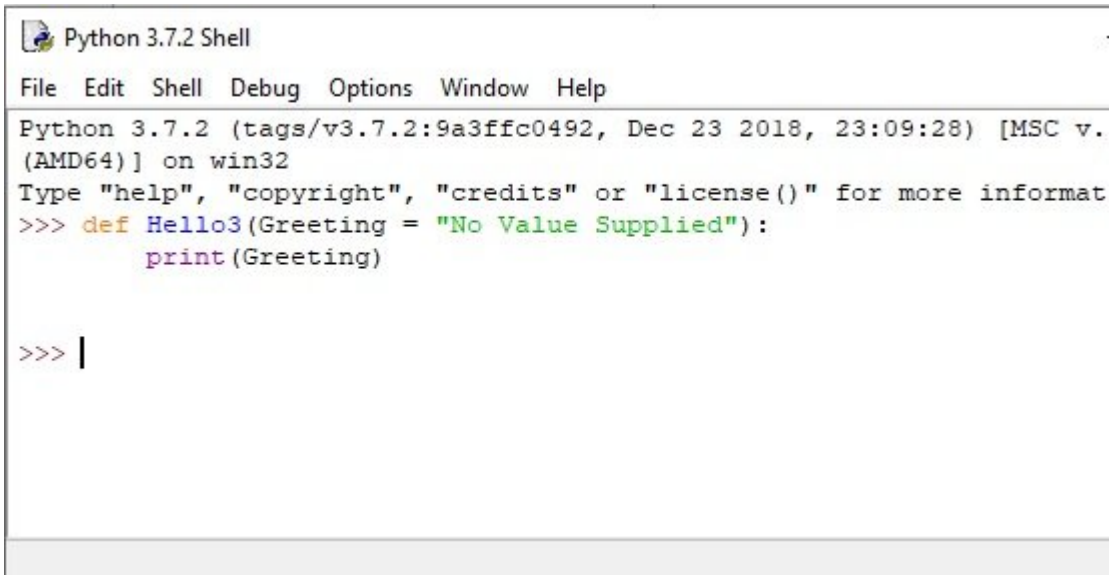
دومرتبه مشاهده می‌کنید که تابع عبارت 2+3=5 را نشان می‌دهد، حتما زمانی که جای آرگومان‌ها با یکدیگر عوض شده‌اند.



چگونه می‌توانیم توابع را همراه با مقادیر پیش‌فرض فراخوانی کنیم؟

زمانی که تابعی را فراخوانی می‌کنید که آرگومان‌ها یا کلیدهاواژه‌هایی برای آن تعریف شده است، مجبور هستید در زمان فراخوانی حتما آرگومان‌های آن‌را مشخص کنید، در غیر این صورت در زمان فراخوانی با مشکل روبرو خواهید شد. اما برای آن‌که بتوانید یک تابع را با آرگومان تعریف کرده، اما مجبور نباشید در هر بار فراخوانی آرگومان‌ها را مشخص کنید باید از مقادیر پیش‌فرض برای آرگومان‌ها استفاده کنید. مقادیر پیش‌فرض برای آرگومان‌ها فرآیند ساخت توابع را ساده‌تر کرده و احتمال بروز خطا زمانی که برنامه‌نویس ورودی را تعیین نمی‌کند را کاهش می‌دهند. برای ساخت یک مقدار پیش‌فرض، شما باید به سادگی در مقابل نام یک آرگومان از علامت مساوی استفاده کرده و مقداری را مقابل آن درج کنید. برای آن‌که ببنید این تکنیک چگونه کار می‌کند در محیط IDLE تابع زیر را بنویسید.

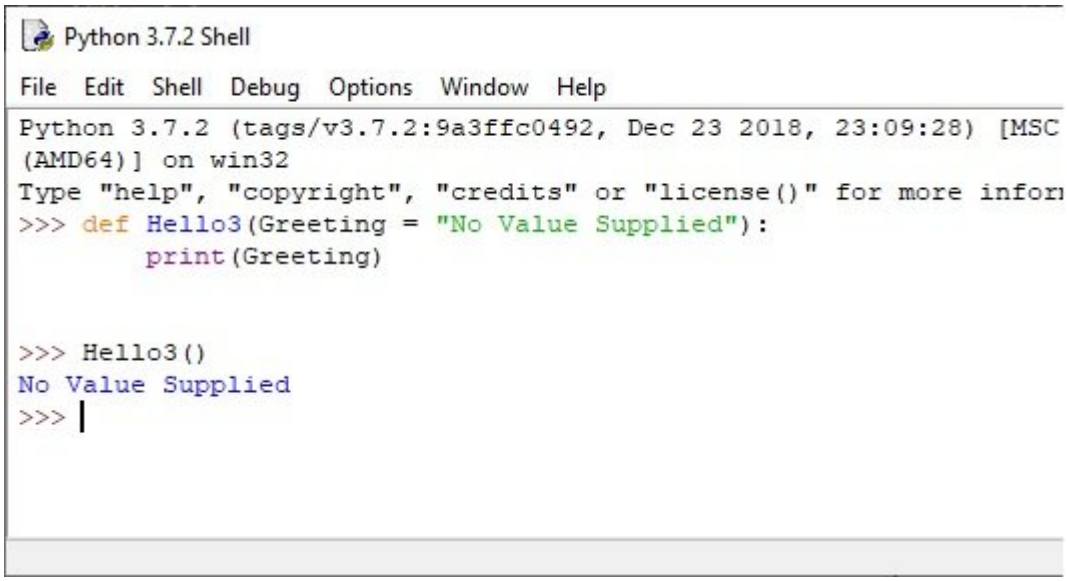
```
def Hello3(Greeting = "No Value Supplied"):

print(Greeting)
```



زمانی که کاربری سعی کند تابع فوق را بدون تعیین آرگومان مربوطه فراخوانی کند، هیچ پیغام خطایی دریافت نخواهد کرد، زیرا تابع فوق دارای یک مقدار پیش‌فرض است که در صورت عدم تعیین آرگومان این مقدار استفاده خواهد شد. فرمان زیر را تایپ کرده و کلید اینتر را فشار دهید.

```
Hello3()
```

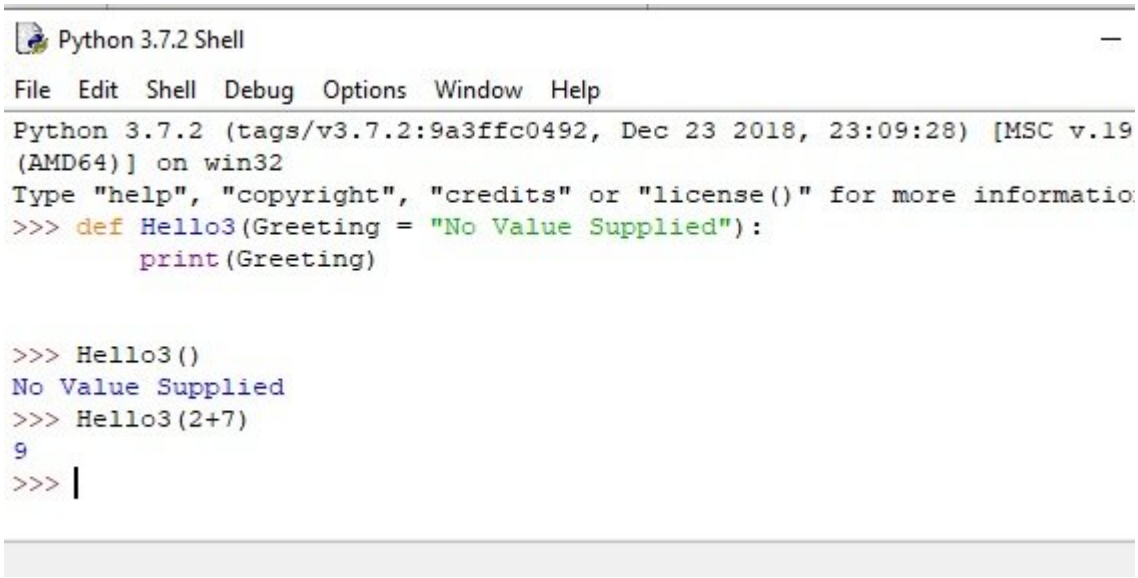
مشاهده می‌کنید که مقداری که مقابل آرگومان تعیین شده است چاپ می‌شود.



حال اگر تابع فوق را همراه با مقداری فراخوانی کنید، مقدار شما جایگزین مقدار پیش‌فرض شده و روی صفحه‌نمایش نشان داده می‌شود. تابع فوق را همراه با مقدار زیر فراخوانی کنید.

```
Hello3(2+7)
```

خروجی تابع این بار حاصل جمع دو مقدار وارد شده است.



ساخت توابع با تعداد شناوری از آرگومان‌ها

در بیشتر مواقع، شما می‌دانید که یک تابع چه تعداد آرگومان را به عنوان ورودی دریافت می‌کند. فراخوانی توابعی که دارای تعداد ثابتی آرگومان هستند کار سختی نیست، اما گاهی اوقات شما به سادگی نمی‌توانید مشخص کنید تابعی که در حال نوشتن آن هستید چه تعداد آرگومان را باید دریافت کند. در برخی موارد مجبور هستید توابعی را بنویسید که تعداد ثابتی آرگومان نخواهند داشت. به‌طور مثال، ممکن است یک برنامه‌ای در پایتون بنویسید که قرار است در محیط خط فرمان اجرا شود، اما تعداد ورودی‌هایی که قرار است از کاربر دریافت کند ثابت نیست. در چنین شرایطی شما نمی‌توانید یک تعداد مشخص آرگومان را برای یک تابع تعریف کنید. در این حالت پایتون به شما پیشنهاد می‌کند از تکنیک آرگومان‌های شناور استفاده کنید. آرگومان‌های شناور اجازه می‌دهند یک تعداد شناور از آرگومان‌ها را برای یک تابع مشخص کنید. برای ساخت توابعی با آرگومان‌های شناور کافی است پیش از تعریف یک آرگومان از کاراکتر

ستاره استفاده کنید. در این حالت تابع شما محدود به یک آرگومان مشخص نبوده و هر تعداد آرگومانی که برای آن ارسال کنید را به عنوان ورودی دریافت می‌کند. برای آشنایی بیشتر با این تکنیک در محیط IDLE تابع زیر را بنویسید. (در قطعه کد زیر چند دستور جدید وجود داد که در آینده با آن‌ها بیشتر آشنا خواهید شد.) در زمان تایپ فرمان‌های فوق به مکان قرارگیری اشاره‌گر دقت کنید. همان‌گونه که مشاهده می‌کنید **پایتون** از تورفتگی‌ها برای مشخص کردن بلوک‌هایی از دستورات استفاده می‌کند.

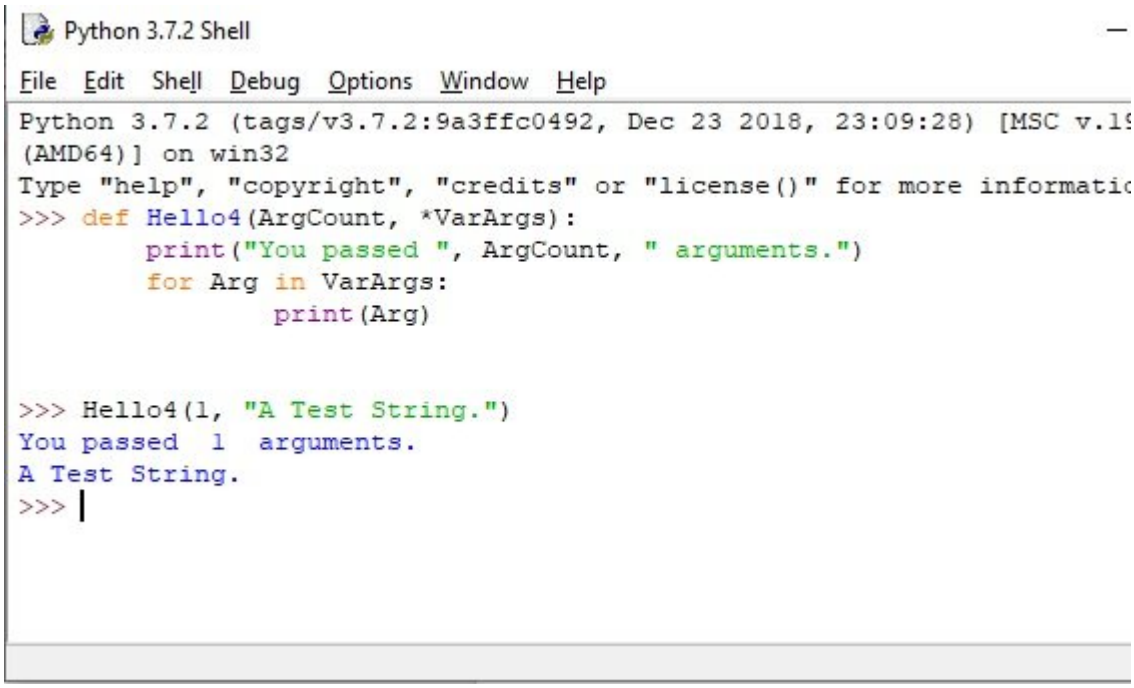
```
def Hello4(ArgCount, *VarArgs):

print("You passed ", ArgCount, " arguments.")

for Arg in VarArgs:

print(Arg)
```

این مثال از یک فرمان کلیدی به نام for استفاده می‌کند.for یک دستور کلیدی است که برای ساخت یک حلقه استفاده می‌شود. ما در شماره‌های آتی اطلاعات بیشتری در ارتباط با حلقه for به دست خواهیم آورد. در حال حاضر همه آن چیزی که باید بدانید این است که فرمان for آرگومان‌های VarArgs را دریافت کرده، مقادیر این آرگومان را درون متغیر Arg قرار داده و از فرمان print برای چاپ مقادیر Arg استفاده می‌کند. در این‌جا هدف این است که با نحوه تعریف تعداد شناوری از آرگومان‌ها و نحوه کار آن‌ها آشنا شوید.

پس از آن‌که تابع فوق را تعریف کردید در مرحله بعد با اجرای فرمان زیر تابع ساخته شده را فراخوانی کنید.

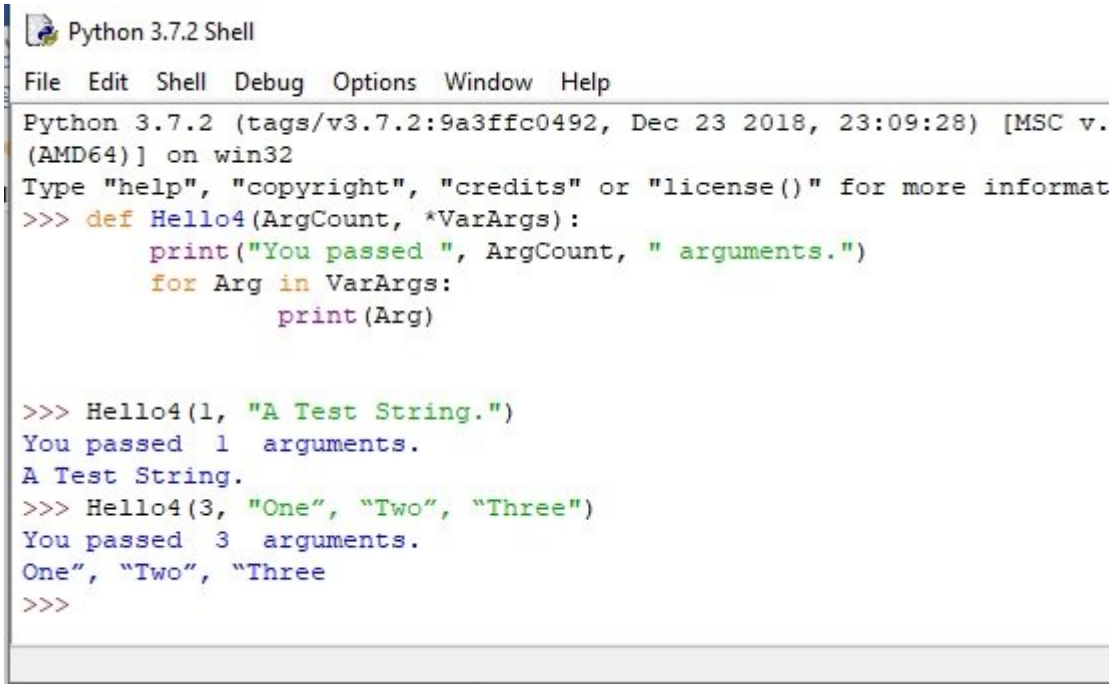
```
Hello4(1, “A Test String.”)
```

مشاهده می‌کنید که تعداد آرگومان‌ها و رشته A test String در خروجی چاپ می‌شود.



خروجی تصویر بالا رشته‌ای که مشخص کرده‌اید را چاپ کرده و در ادامه اعلام می‌دارد که شما در زمان فراخوانی تابع از یک آرگومان استفاده کرده‌اید. اکنون از فرمان زیر برای فراخوانی تابع استفاده کنید:

```
Hello4(3, “One”, “Two”, “Three”)
```



در فراخوانی فوق بدون این‌که هیچ کار اضافه‌ای انجام دهیم، چهار مقدار برای تابع ارسال کردیم که این مقادیر در خروجی چاپ شدند. تکنیک تعریف توابع با آرگومان‌های شناور به ما کمک می‌کنند در زمان نوشتن برنامه‌های حرفه‌ای بدون مشکل توابع منعطف بنویسیم.

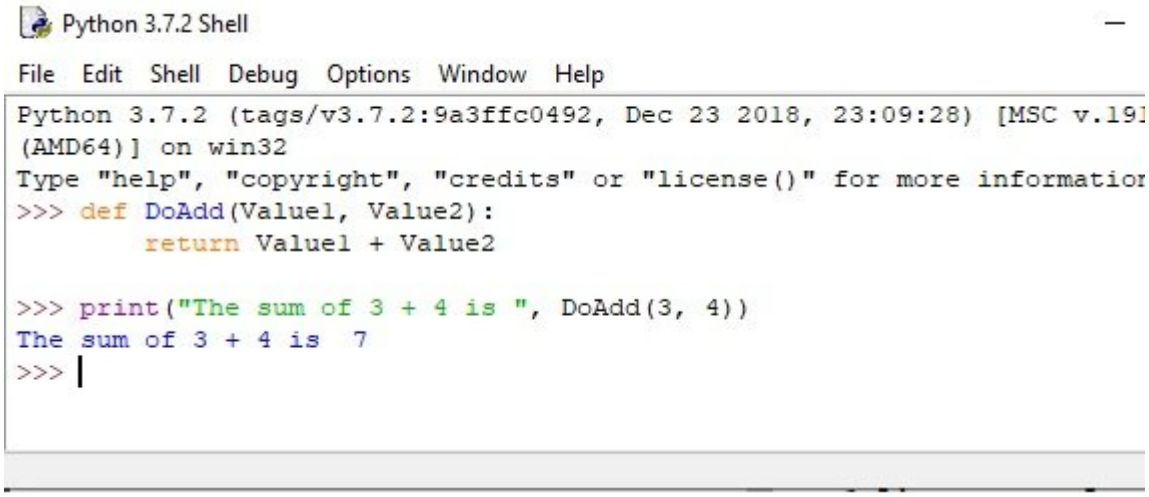
اطلاعات بازگشتی از توابع

توابع با هدف پردازش اطلاعات و ارائه خروجی ایجاد می‌شوند. در مثال‌های بالا کاری که ما انجام دادیم، تعریف توابع همراه با آرگومان‌ها و سپس چاپ نتایج بود. اما در برنامه‌های حرفه‌ای توابع پس از آن‌که محاسبه‌ای روی اطلاعات انجام دادند مقداری را به عنوان مقدار بازگشتی نشان می‌دهند. مقداری که ماحصل یک محاسبه بوده و قرار است از آن استفاده شود. به عبارت دیگر، توابع تنها با هدف محاسبه و چاپ اطلاعات روی صفحه‌نمایش تعریف نمی‌شوند. برای آن‌که یک توابع بتواند مقداری را به عنوان مقدار بازگشتی نشان دهد باید از کلیدواژه return استفاده کنید. در بیشتر موارد مقدار بازگشتی از یک تابع درون یک متغیر ذخیره شده یا در یک عبارت محاسباتی از آن استفاده می‌شود. اما این تکنیک چگونه کار می‌کند؟ محیطIDLE را باز کرده و دستورات زیر را درون آن تایپ کنید:

```
def DoAdd(Value1, Value2):

return Value1 + Value2
```

این تابع دو مقدار را به عنوان ورودی دریافت کرده و حاصل جمع دو مقدار ورودی را به عنوان خروجی خود باز می‌گرداند. دقت کنید این تابع قرار نیست محتوایی را چاپ کند، بلکه تنها نتیجه یک محاسبه ساده را بر می‌گرداند که این مقدار بازگشتی می‌تواند درون متغیری قرار گرفته یا به عنوان ورودی در اختیار تابعی قرار گیرد. از فرمان زیر برای فراخوانی تابع ساخته شده استفاده کنید:

```
print(“The sum of 3 + 4 is ”, DoAdd(3, 4))
```

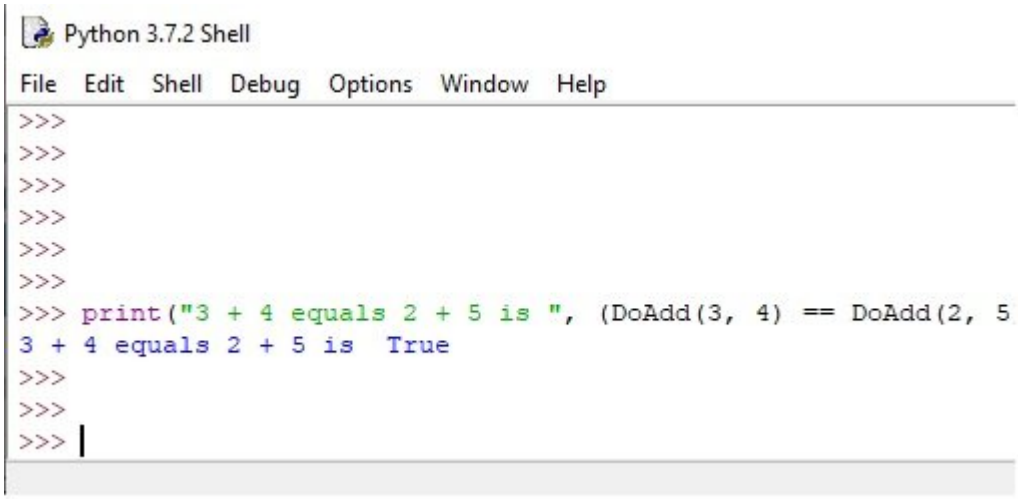


به خروجی دستورات بالا دقت کنید. شما از مقدار بازگشتی تابع DoAdd به عنوان یکی از آرگومان‌های فرمان print استفاده کردید. پیشنهاد می‌کنم برای فهم بهتر موضوع توابع مختلفی را به این شکل بنویسید که خروجی رشته‌ای، اعشاری و صحیح را باز گردانند.

مقایسه خروجی توابع

شما مقدار بازگشتی توابع را به اشکال مختلفی استفاده می‌کنید. به‌طور مثال، در قطعه کد قبلی خروجی تابع به عنوان ورودی در اختیار تابع دیگری قرار گرفت. مقادیر بازگشتی با هدف مقایسه نتایج نیز استفاده می‌شوند. به‌طور مثال زمانی که قصد دارید یک خروجی منطقی را به دست آورید، از مقادیر بازگشتی توابع برای مقایسه استفاده می‌کنید. برای درک بهتر این مسئله تابعی که در پاراگراف قبل تعریف کردیم را با استفاده از فرمان زیر فراخوانی کنید:

```
print("3 + 4 equals 2 + 5 is ", (DoAdd(3, 4) == DoAdd(2, 5)))
```



با اجرای فرمان فوق شما در خروجی یک مقدار منطقی را درست را مشاهده می‌کنید که اعلام می‌دارد 3+4 برابر با 2+5 است. توابع به ما اجازه می‌دهند کدهای انعطاف‌پذیرتر و کوتاه‌تری را بنویسیم.

در شماره آینده آموزش **پایتون** به سراغ مبحث دریافت ورودی از کاربر خواهیم رفت.

تاریخ انتشار:

23 بهمن 1397

نشانی منبع:
https://www.shabakeh-mag.com/workshop/programming/14594/%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-%D8%A2%D8%B1%DA%AF%D9%88%D9%85%D8%A7%D9%86%E2%80%8C%D9%87%D8%A7%DB%8C-%D9%88%D8%B1%D9%88%D8%AF%DB%8C%D8%8C-%D9%85%D9%82%D8%AF%D8%A7%D8%B1-%D8%A8%D8%A7%D8%B2%DA%AF%D8%B4%D8%AA%DB%8C