آموزش رایگان پایتون (python)- پیاده‌سازی حلقه‌ها با فرمان for در پایتون

در شماره گذشته آموزش پایتون مبحث مدیریت خطاها را به پایان رساندیم و همان‌گونه که خاطر نشان کردیم، در این شماره به سراغ مبحث ساخت حلقه‌ها در پایتون خواهیم رفت.

برای مطالعه بخش بیستم آموزش رایگان پایتون اینجا کلیک کنید

تمامی مثال‌هایی که تاکنون مشاهده کردید به جزء قطعه کدی که در آموزش گذشته آموزش **پایتون** بررسی کردیم، به شکلی نوشته شده بودند که دستورات را تنها یک‌بار اجرا می‌کردند. با این حال، در دنیای واقعی این شیوه چندان مفید نیست. بیشتر دستوراتی که در برنامه خود می‌نویسید همانند کارهایی که در دنیای واقعی انجام می‌دهید باید چند مرتبه تکرار شوند. پایتون راهکارهای مختلفی برای تکرار دستورات در اختیارتان قرار می‌دهد. در حقیقت بیشتر زبان‌های برنامه‌نویسی تکنیک‌های مختلفی برای تکرار دستورات در اختیار توسعه‌دهندگان قرار می‌دهند که به این تکنیک حلقه‌سازی می‌گوییم. حلقه‌سازی شبیه به ساخت دایره‌ای است که دستورات در آن مادامی که حلقه به انتهای کار خود نرسیده باشد، تکرار می‌شوند.

پردازش داده‌ها با استفاده از فرمان کلیدی for

اولین دستوری که برای ساخت حلقه تکرار از آن استفاده می‌شود حلقه for است. به سختی می‌توان یک زبان برنامه‌نویسی مدرن را پیدا کرد که این حلقه تکرار درون آن قرار نگرفته باشد. در این حلقه تکرار دستورات به تعداد مشخصی تکرار می‌شوند و این شما هستید که بر روند تکرار دستورات با تعیین مقداری مشخص کنترل دارید. از آن‌جایی که شرط یک حلقه for در ابتدای آن مشخص می‌شوند، به‌کارگیری این حلقه با کمترین پیچیدگی ممکن امکان‌پذیر است. تنها نکته‌ای که در رابطه با این حلقه باید بدانید این است که دستورات قرار است چند مرتبه تکرار شوند.

عملکرد حلقه for چگونه است؟

یک حلقه for با یک عبارت کار خود را آغاز می‌کند. این دستور مشخص می‌کند که حلقه چگونه باید آغاز شود. به مثال ساده زیر دقت کنید.

For Letter in “shabakeh!”:

این خط کد با کلمه کلیدی for آغاز می‌شود. عنصر بعدی متغیری است که برای نگه‌داری مقادیری که خوانده شده‌اند استفاده می‌شود. در مثال ما این متغیر Letter نام دارد. پس از تعریف متغیری که فرآیند انتساب به آن انجام می‌شود، کلمه کلیدی in ظاهر می‌شود. این کلمه کلیدی به پایتون اعلام می‌دارد که باید یک فرآیند خواندن با رویکرد تکرارشوندگی را درون عبارتی که پس از کلمه کلید in مشخص شده است انجام دهد. در مثال ما این فرآیند تکرار باید در ارتباط با رشته shabakeh انجام شود. تعریف یک حلقه for همیشه با کاراکتر : پایان پذیرفته و پس از آن دستورات مختلفی همچون دستورات شرطی استفاده می‌شوند. دستوراتی که پس از این تعریف نوشته می‌شوند جزیی از بدنه حلقه for هستند که یکسری وظایف تکرار شونده را شامل می‌شوند.

ساخت یک حلقه for ابتدایی

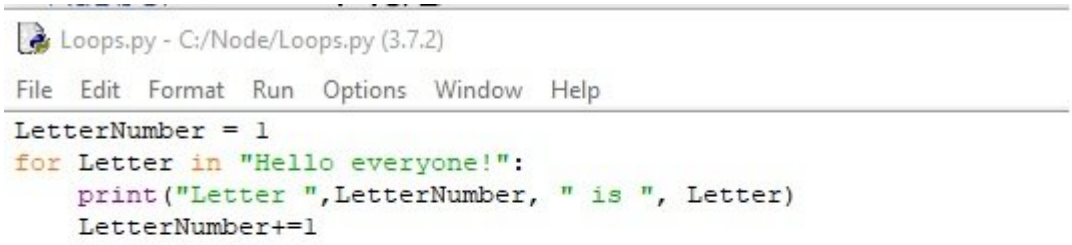
بهترین راه برای آشنایی با نحوه کار حلقه for ساخت یک نمونه ساده از این حلقه است. در مثال زیر حلقه for کاراکترهایی که درون عبارت زیر قرار دارند را یک‌به‌یک خوانده و در ادامه فرمان print این کاراکترها را به همراه تعداد آن‌ها روی صفحه نشان می‌دهد.

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

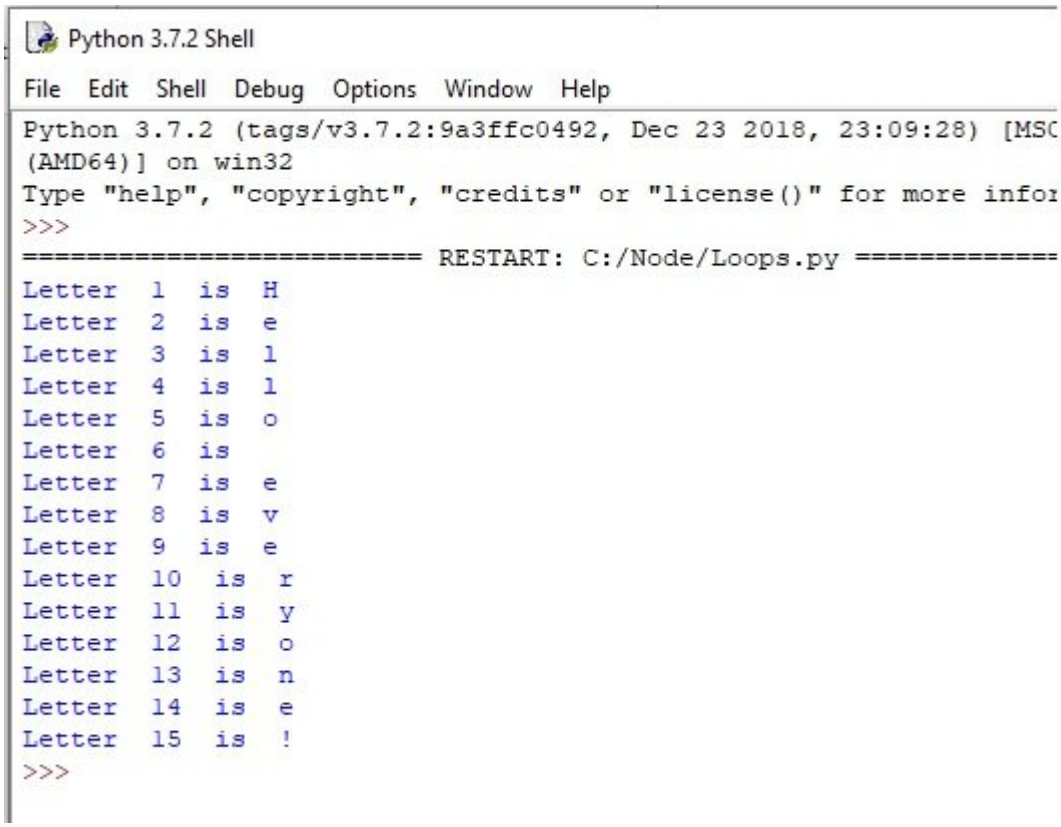
2. در پنجره باز شده دستورات زیر را وارد کنید.

```
LetterNumber = 1

for Letter in "Hello everyone!":

    print("Letter ",LetterNumber, " is ", Letter)

    LetterNumber+=1
```

قطعه کد بالا با تعریف یک متغیر به نام LetterNumber کار خوا را آغاز می‌کند. ما قصد داریم از این متغیر برای شمارش تعداد کاراکترها استفاده کنیم. هر زمان که حلقه کامل می‌شود یک مقدار به LetterNumber اضافه می‌شود. پس از تعریف متغیر LetterNumber، دستور for قرار دارد که برای شمارش کاراکترهایی که درون رشته Hello everyone قرار دارد استفاده می‌شود. هر زمان کاراکتری از درون رشته خوانده شود، کاراکتر خوانده شده درون متغیر Letter قرار می‌گیرد. در ادامه فرمان print تک به تک کاراکترهایی که خوانده می‌شوند را همراه با تعداد شمارش شده روی صفحه چاپ می‌کند. به فرورفتگی دستورات دقت کنید که همانند شکل زیر باشند.



3. برنامه را ذخیره کرده و از منوی Run گزینه New Module را انتخاب کنید. خروجی این برنامه همانند شکل زیر است:



کنترل اجرای یک حلقه با دستور break

در زندگی استثناءهای مختلفی برای قوانین وجود دارد. به‌طور مثال، شما در نظر دارید خط مونتاژی برای تولید ساعت راه‌اندازی کنید.، اما ممکن است مواد اولیه در اختیار نداشته باشید، در این حالت اگر بخشی مواد لازم در اختیار نداشته باشد، خط مونتاژ در اواسط پردازش کار خود را متوقف می‌کند، این توقف به معنای آن نیست که کارها خراب شده‌اند یا قرار است کار کنار گذاشته شود؛ بلکه به معنای آن است که فعالیت‌ها به شکل موقت به حالت تعلیق درآمده‌اند تا مواد اولیه از راه برسند. در دنیای کامپیوترها نیز وقفه‌های مختلفی رخ می‌دهد. شما ممکن است در حال استریم کردن داده‌هایی از یک سایت آنلاین باشید، اما زمانی که وقفه‌ای رخ می‌دهد، اتصال شما قطع می‌شود. قطعی موقت ارتباط وقفه‌ای در کار شما به وجود می‌آورد؛ اما پس از آن‌که اتصال برقرار شد، شما می‌توانید کارهایی که تکمیل نشده‌اند را به سرانجام برسانید. فرمان break به شما اجازه می‌دهد از یک حلقه خارج شده یا به عبارت دقیق‌تر یک حلقه را بشکنید. شما از یک دستور if برای تعریف وضعیتی که قرار است باعث خروج از حلقه شوند استفاده می‌کنید. به‌طور مثال ممکن است در یک حلقه شرطی را مشخص کنید که اگر ارتباط قطع شد فرمان break باعث خروج از حلقه شود. برای روشن شدن بهتر نحوه استفاده از فرمان Break به مثال زیر دقت کنید:

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

2. در پنجره جدید قطعه کد زیر را وارد کنید.

```
Value =input("Type less than 6 characters: ")

LetterNum = 1

for Letter in Value:

    print("Letter " , LetterNum, " is ", Letter)

    LetterNum+=1
```

```
if LetterNum > 6:
    print("The string is too long!")
    break
```

Loops.py - C:/Node/Loops.py (3.7.2)

File Edit Format Run Options Window Help

```
Value =input("Type less than 6 characters: ")
LetterNum = 1
for Letter in Value:
    print("Letter " , LetterNum, " is ", Letter)
    LetterNum+=1
    if LetterNum > 6:
        print("The string is too long!")
        break
```

در قطعه کد بالا فرمان input از کاربر درخواست می‌کند تا رشته‌ای که کمتر از 6 کاراکتر است را وارد کند. مقدار وارد شده درون متغیر Value قرار می‌گیرد. در ادامه به متغیر LetterNum مقدار 1 تخصیص داده می‌شود. در ادامه دستور حلقه for تعریف می‌شود که قرار است کاراکترهای درون متغیر Value را خوانده و درون متغیر Letter قرار دهد. در ادامه فرمان print کاراکترها را همراه با تعداد آن‌ها چاپ می‌کند. در فرمان if بررسی می‌شود که اگر مقدار شمارنده LetterNum بزرگ‌تر از 6 بود به کار حلقه for پایان دهد. در ادامه پیغامی روی صفحه ظاهر شده و به کاربر اعلام می‌شود که تعداد کاراکترهای رشته بیشتر از 6 عدد بوده‌اند و در ادامه فرمان break اجرا می‌شود. این فرمان باعث می‌شود تا از حلقه خارج شوید.

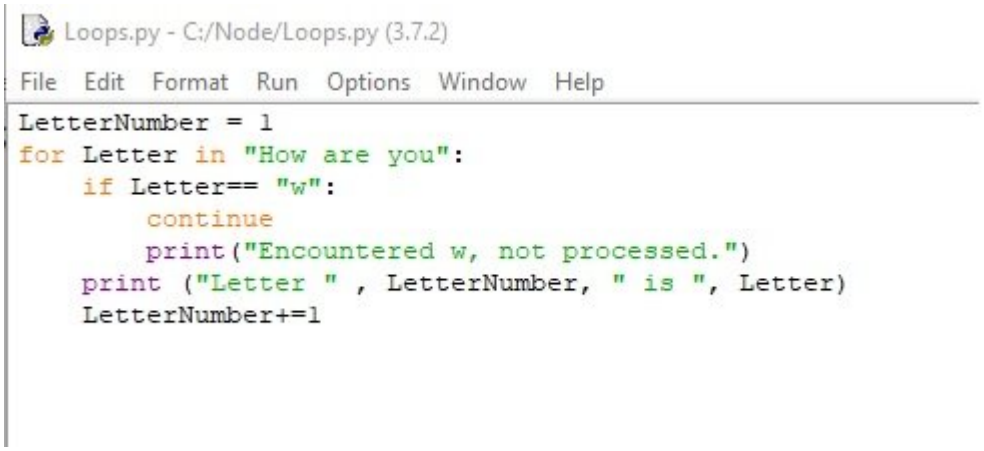
```
========== RESTART: C:/Node/Loops.py =======
Type less than 6 characters: Hello
Letter  1  is  H
Letter  2  is  e
Letter  3  is  l
Letter  4  is  l
Letter  5  is  o
>>>
```

3. برنامه را ذخیره کرده و از منوق Run گزینه Run Module را انتخاب کنید.

4. مقدار Hello را تایپ کرده و کلید اینتر را فشار دهید.

```
========== RESTART: C:/Node/Loops.py =============
Type less than 6 characters: I am too long
Letter  1  is  I
Letter  2  is
Letter  3  is  a
Letter  4  is  m
Letter  5  is
Letter  6  is  t
The string is too long!
>>>
```

5. برنامه را دومرتبه اجرا کرده و این بار رشته I am too long را وارد کرده و کلید اینتر را فشار دهید. همان‌گونه که تصویر زیر نشان می‌دهد، حلقه ما 6 کاراکتر رشته را خوانده و مقادیر را روی صفحه چاپ می‌کند، اما زمانی که شمارنده از 6 عبور کرد، دستوراتی که درون فرمان if قرار دارند اجرا شده و حلقه بدون محاسبه مابقی کاراکترهای درون رشته خاتمه پیدا می‌کند.



کنترل اجرای دستورات با فراخوانی دستور continue

گاهی اوقات شما می‌خواهید هر عنصری که درون مجموعه‌ای قرار دارد را بررسی کنید، اما در نظر ندارید فرآیند بررسی روی عناصر خاصی انجام شوند. به‌طور مثال، شما ممکن است تصمیم گرفته باشید که همه اطلاعات مرتبط با ماشین‌هایی که اطلاعاتشان درون یک بانک اطلاعاتی ثبت شده است را به جز ماشین‌هایی که رنگ قهوه‌ای دارند پردازش کنید. در چنین شرایطی باید از یک دستور پرشی استفاده کنید که ماشین‌های دارای رنگ قهوه‌ای را پردازش نکند. در پاراگراف قبل دیدید که فرمان break باعث می‌شود از یک حلقه خارج شوید، در نتیجه به‌کارگیری این فرمان در چنین شرایطی مناسب نیست، زیرا اجازه نمی‌دهد همه عناصر درون یک مجموعه ترتیبی پردازش شوند. در چنین شرایطی فرمان continue به شما کمک می‌کند. درست شبیه به فرمان break برای فرمان continue نیز باید از یک دستور if درون یک حلقه for استفاده کنید. زمانی که فرمان فوق فراخوانی می‌شود، دستوراتی که پس از این فرمان قرار دارند اجرا می‌شوند، بی آن‌که از حلقه خارج شده یا مشکل خاصی به وجود آید. برای روشن شدن بحث اجازه دهید این مسئله را با ذکر مثالی بررسی کنیم.

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

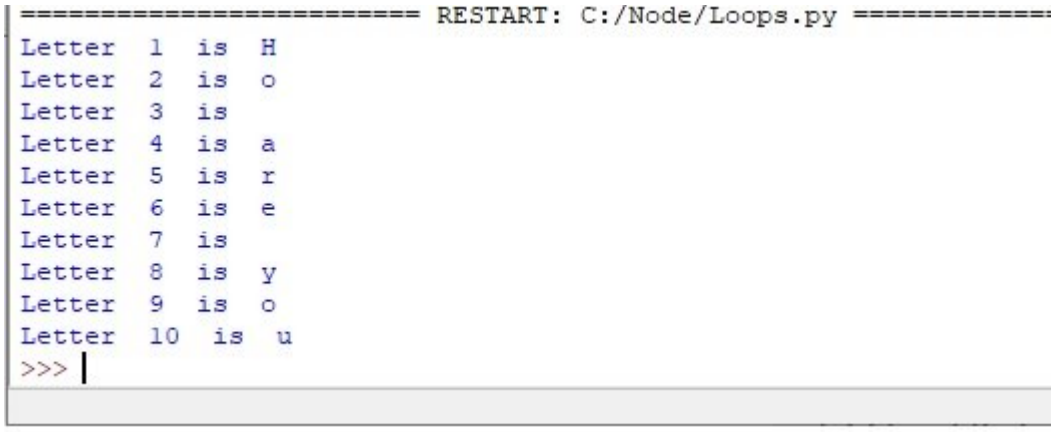
2. در پنجره ظاهر شده قطعه کد زیر را وارد کنید.

```
LetterNum = 1

for Letter in "How are you!":

if Letter == "w":

continue

print("Encountered w, not processed.")

print("Letter ", LetterNum, " is ", Letter)

LetterNum+=1
```

قطعه کد بالا مشابه مثالی است که در پاراگراف قبل به آن اشاره کردیم، با این تفاوت که از فرمان continue به جای break استفاده کردیم. در قطعه کد بالا در بخش if تعیین کرده‌ایم که اگر کاراکتر w درون رشته ما قرار داشت، چاپ نشود. دقت کنید فرمان print که پس از فرمان continue قرار دارد جزیی از بلوک if بوده اما هیچ‌گاه

اجرا نخواهد شد، زیرا با اجرای فرمان continue حلقه فورا به ابتدا بازگشته و کاراکتر w نیز در خروجی چاپ نخواهد شد.

3. از منوی Run گزینه Run Module را انتخاب کنید. همان‌گونه که شکل زیر نشان می‌دهد، ترکیب فرمان if و continue به ما اجازه می‌دهد تا محدودیت‌های کاملا کنترل شده‌ای در یک حلقه اعمال کنیم.



در شماره آینده آموزش **پایتون** به سراغ مباحث دیگر مرتبط با حلقه‌ها خواهیم رفت.

تاریخ انتشار:

14 اسفند 1397

نشانی منبع:
https://www.shabakeh-mag.com/workshop/programming/14745/%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-%D9%BE%DB%8C%D8%A7%D8%AF%D9%87%E2%80%8C%D8%B3%D8%A7%D8%B2%DB%8C-%D8%AD%D9%84%D9%82%D9%87%E2%80%8C%D9%87%D8%A7-%D8%A8%D8%A7-%D9%81%D8%B1%D9%85%D8%A7%D9%86-%D8%AF%D8%B1-%D9%BE%D8%A7%DB%8C%D8%AA%D9%88%D9%86