۲–۲– ساختارهای تکرار(python loops)

درحالت کلی ، دستورات به ترتیب اجرا می شوند. یعنی در یک تابع ابتدا دستور اول، سپس دستور دوم و به همین ترتیب ادامه می یابد . اما گاهی لازم است مجموعه دستورات داخل یک قطعه کد بارها تکرارشود.

از ساختار کنترلی حلقه ها در برنامه نویسی، برای اجرای مجموعه ای از دستورها به تعداد دفعات لازم یا تا زمانی که یک شرط معین درست و برقرار باشد، استفاده می شود.

در حلقه ، هنگامی که مجموعه دستورات حلقه به طور کامل اجرا می شوند، برنامه بار دیگر به ابتدای مجموعه دستورات حلقه رفته و در صورت برقرار بودن شرط حلقه، یکبار دیگر دستورات آن به طور کامل اجرا می کند.

زبان برنامه نویسی پایتون دو حلقه ی for و while پشتیبانی می کند که هریک در زیر تشریح شده:

حلقه	شرح
while loop	در این نوع حلقه ، مجموعه دستورالعمل ها به تعداد معلوم و مورد نیاز ، تا زمانی که شرط مشخص شده صحیح می باشد، تکرار خواهد شد . این حلقه قبل از اجرای دستورات بدنه ی حلقه، شرط را بررسی می کند.
for loop	در این نوع حلقه ، مجموعه دستورالعمل ها به تعداد معلوم و مورد نیاز تکرار خواهد شد.
nested loops (حلقه های تودرتو)	می توان درون یک حلقه ی while ، do ..while و forیک یا چند حلقه ی دیگرگنجاند.

۲–۲–۱– الگوی دستور حلقه while:

```
While عبارت شرطی:
    دستورات
    .
    .
```

کار این حلقه اینگونه است که تا زمانیکه نتیجه عبارت شرطی درست باشد دستورات حلقه اجرا می گردد و کار تکرار می شود.اما به محض false یا غلط شدن نتیجه شرط از حلقه خارج شده و دیگر تکرار انجام نمی شود.

مثال۱: برنامه ای بنویسید که اعداد بین ۱ تا ۱۰۰ را نمایش دهد.

```
a=1
x=100
while a<=x:
        print(a)
        a=a+1
```

مثال ۲: برنامه ای بنویسید که اعداد زوج بین ۰ تا ۲۰ را نمایش دهد؟

```
n=1

while n<=20:
    if n%2==0:
      print n
    n=n+1
```

لازم به ذکر است اگر میخواهید از اجرای حلقه خارج شوید و آن را متوقف کنید می توانید از دستورات کنترلی حلقه ها به شرح زیر استفاده نمایید:

۲-۲-۲- دستورات کنترلی حلقه ها

دستورات کنترلی اجرا را از روال عادی خود خارج کرده و تغییراتی در آن بوجود می آورد .پس از اینکه اجرا آن حوزه را ترک می کند، تمامی اشیایی که به صورت خودکار در آن حوزه بوجود آمده بودند، از بین خواهد رفت.

دستور کنترلی	شرح
break statement	از دستور break برای خروج کامل از ادامه اجرای دستورات یک حلقه درصورت بر قرار بودن شرط تعیین شده برای آن استفاده می شود.
continue statement	از دستور continue ، برای خارج شدن از ادامه اجرای یکبار دستورات حلقه و پرش به گام بعدی حلقه استفاده می شود.
pass statement	دستور pass در زبان پایتون زمانی مورد استفاده قرار می گیرد که یک دستور ازلحاظ ساختار برنامه نویسی مورد نیاز باشد ، اما شما هیچ فرمان یا کدی را برای اجرا لازم نداشته باشید.

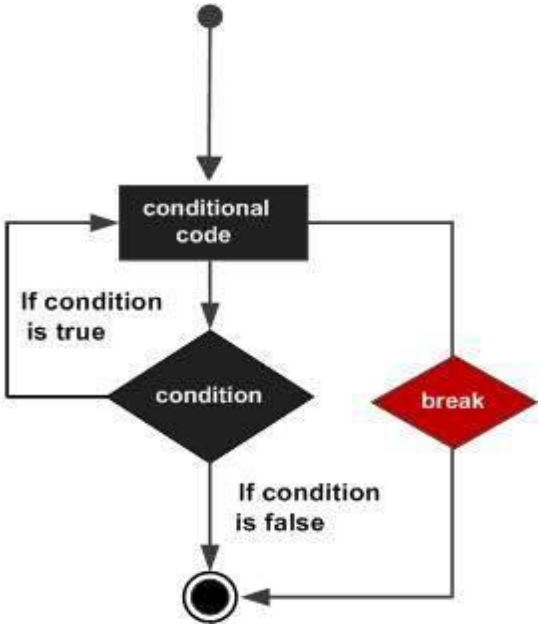
الف) دستور break

دستور **break** در زبان پایتون ، حلقه جاری را به پایان رسانده و درست همانند **break** سنتی موجود در **C** ، اجرا را از دستور بعدی از سر می گیرد.

بیشترین کاربرد **break** زمانی است که اتفاقی در خارج از حلقه رخ داده است که خروج سریع از حلقه را می طلبد.

دستور **break** را می توان در هر دو حلقه **while** و **for** استفاده نمود.

در صورت استفاده از حلقه های تودرتو، دستور **break** اجرای درونی ترین حلقه را متوقف کرده و به اجرای دستور بعدی پس از قطعه کد می پردازد.



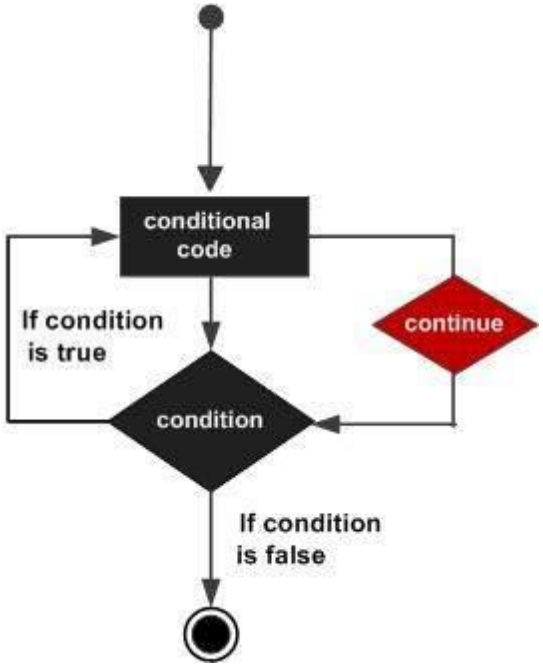
مثال۳:برنامه ای بنویسید که اعداد بین ۱ تا ۱۰۰ را چاپ کند و در حال چاپ اگر به عدد ۵۰ رسید از حلقه خارج شود.

```
a=1
x=100
while a<=x:
   print(a)
   a=a+1
   if a<=50:
      break
```

ب)دستور continue در زبان برنامه نویسی پایتون:

دستور **continue** از روی تمامی دستورهای باقی مانده در تکرار جاری حلقه پریده و کنترل را به بالای حلقه بازمی گرداند(انتقال می دهد.)

دستور **continue** را می توان در هر دو حلقه **while** و **for** بکار برد.



مثال۴: خروجی برنامه زیر چیست؟

```
var = 10
while var > 0:
   print (('Current variable value :'),var)
   var = var -1
   if var == 5:
      continue
print ("Good bye!")
```

ج)دستورpass

دستور **pass** در **زبان پایتون** زمانی بکار می رود که یک دستور از لحاظ ساختار برنامه نویسی مورد نیاز است، اما شما هیچ فرمان یا کدی را برای اجرا لازم نداشته باشید.

دستور **pass** ، یک عملیات تهی می باشد؛ بدین معنا که هنگام اجرای آن هیچ اتفاقی نمی افتد.

دستور **pass** همچنین در جاهایی که کد شما بعدا نوشته خواهد شد ، اما هنوز نوشته نشده است،بسیار مفید می باشد (: به عنوان مثال ، در **stub** ها.)

مثال۵: خروجی برنامه زیر چیست؟

```
for letter in 'Python':
   if letter == 'h':
      pass
      print ('This is pass block')
   print ('Current Letter :', letter)
print ("Good bye")
```

۲-۲-۳- آشنایی با دستور حلقه for ... in

این ساختار هم یک حلقه است که عملکردی درست مشابه با حلقه while دارد با این فرق که برای اجرای حلقه نیاز دارد که از واژه کلیدی in استفاده کند.به این کلمه کلیدی و کاربردهایش در بخش دیگری پرداخته می شود. در زیر ساختار آن را مشاهده می کنید.

For نام متغییر **in** مقدار تکرار شدنی

مجموعه دستورات

مقدار تکرار شدنی هر نوع داده ای است که مینواند تکرار شود (اعضایش تکه تکه شوند) و شامل **string**ها (که میتواند کاراکترها تکه تکه شوند) و یا **list**ها و یا **tuple** ها و .... باشد.

مثال۱: خروجی برنامه زیر چیست؟

```
for item in [1, 2, 3]:
   print(item)
```

مثال ۲: خروجی برنامه زیر چیست؟

```
for char in 'python':
   print(char)
```

مثال ۳: خروجی برنامه زیر چیست؟

```
d = {'name': 'Jhon', 'job': 'designer', 'age': 40}
for key in d:
   print(key)
```

مثال۴: محاسبه مجموع اعداد داخل یک لیست به کمک حلقه **for**

```
numbers = [6, 5, 3, 8, 4, 2, 5, 4, 11]
sum = 0
for val in numbers:
   sum = sum+val
print ("The sum is", sum)
```

مثال۵ : فاکتوریل عدد **n** را بنویسید؟

```
n=int(input("enter a number:"))

f=1

for i in range(1,n+1):

   f=f*i

print (f)
```

نکته: تابع **range** وظیفه اش تولید اعدا بین دو مقدار داده شده در این دستور است.

تابع range() در پایتون

می‌توان یک توالی از اعداد را با استفاده از تابع range() تولید کرد. مثلا با دستور range(10) اعداد از ۰ تا ۹ را تولید می‌کند (ده عدد). همچنین، می‌توان سایز شروع، پایان و گام را به عنوان range(start,stop,step size) تعریف کرد. سایز گام به طور پیش‌فرض و در صورتی که مقدار دهی نشده باشد، برابر با یک خواهد بود. این تابع، همه مقادیر را در حافظه ذخیره نمی‌کند زیرا موجب عدم کارایی می‌شود. این در حالی است که نقطه شروع، توقف و سایز گام را به خاطر دارد و عدد بعدی را ضمن تکرار می‌سازد. برای مجبور کردن این تابع به خروجی دادن همه عناصر، می‌توان از تابع list() استفاده کرد.

حلقه for باelse

یک حلقهfor ، می‌تواند یک بلوک else انتخابی نیز داشته باشد. بخشelse ، در صورتی اجرا می‌شود که عناصر توالی مورد استفاده از حلقه for به پایان برسند. از عبارت break می‌توان برای متوقف کردن حلقه for نیز استفاده کرد. در چنین شرایطی، بخش else نادیده انگاشته می‌شود. بنابراین، قسمت else حلقهfor ، در صورتی که هیچ خطایی وجود نداشته باشد اجرا می‌شود. در ادامه، مثالی برای حلقه for همراه با else ارائه شده است.

```
digits = [0, 1, 5]
 for i in digits:
   print(i)
else:
   print("No items left.")
```

خروجی حاصل از اجرای برنامه بالا، به صورت زیر خواهد بود.

```
   0
   1
   5
   No items left.
```

در اینجا، حلقهfor ، عناصر لیست را تا هنگامی که حلقه متوقف شود، پرینت می‌کند. هنگامی که حلقه for متوقف شد، بلوک کد موجود در else اجرا و پرینت می‌شود.

مسائل حل شده

تمرین۱ : برنامه ای که نمره عددی دانشجویی را بر مبنای ۱۰۰ خوانده ، باتوجه به جدول زیر و نمره کسب شده پیغام مناسبی را نمایش می دهد(در این برنامه متغیر grade نمره است)

نمره	پیغام
0-70	**Fail**
71-80	**Good**
81-90	**Very Good**
90-100	**Excellent**
<0, >100	**Invalid Grade**

مراحل طراحی و اجرا:

ماژول جدیدی ایجاد کرده، دستورات آن را به صورت زیر تایپ کنید:

```
grade = int(input("Enter grade: "))
if  0 <= grade <= 70 :
        print("Fail")
elif  71 <= grade <= 80:
        print("Good")
elif  81 <= grade <= 90:
        print("Very good")
elif  91 <= grade <= 100:
        print("Excellent")
else:
        print("Invalid grade")
```

.

تمرین۲ : برنامه ای بنویسید که عدد n را خواند ، اعداد n تا 1 نمایش می دهد و در پایان حاصل ضر ب ای اعداد را نمایش دهد.

مراحل طراحی واجرا:

۱. ماژول جدیدی ایجاد کرده، دستورات آن را به صورت زیر تایپ کنید:

```
n = int(input("Enter n:"))
p = 1
for i in range(n, 0, -1):
print(i, end = '\t')
p = p * i
print("\nMultiply is ", p)
```

۲ . ماژول را ذخیره و اجرا کرده، و مثلا عدد ۲۵ را وارد نمایید تا خروجی زیر را مشاهده کنید:

```
Enter n:25
25      24      23      22      21      20      19      18      17      16
15      14      13      12      11      10      9       8       7       6
5       4       3       2       1
 Multiply is  15511210043330985984000000
```

تمرین ۳: برنامه ای بنویسید که عددی را خوانده ،سپس بزرگترین رقم آن را نمایش می دهد.

مراحل طراحی و اجرا:

۱.ماژول جدیدی ایجاد کرده، دستورات آن را به صورت زیر تایپ کنید:

```
n = int(input("Enter n:"))
max = n % 10
while n > 0:
        if max < n % 10 :
        max = n % 10
        n = n // 10
print("Max is ", max)
```

۲. ماژول را ذخیره و اجرا کرده، عدد دلخواهی مثل ۸۵۶۴۹۰۱ را وارد کنید تا خروجی زیر را ببینید:

```
Enter n:8564901
Max is 9
```

تمرین ۴ . برنامه ای بنویسید که عددی را خوانده ، مشخص نماید که آیا عدد خوانده شد اول است یا خیر؟

نکته : اگر عددی بر یک عدد کوچک تر یا مساوی نصف خودش به جزیک بخش پذیر باشد، آن عدد اول نیست.

مراحل طراحی و اجرا:

۱. ماژول جدیدی ایجاد کرده، دستورات آن را به صورت زیر تایپ کنید:

```
n = int(input("Enter n:"))
isPrime = True
for i in (2, n // 2+ 1):
        if n % i == 0:
                isPrime = False
                break
if (isPrime == True):
        print("Yes")
else:
        print("No")
```

۲. ماژول را ذخیره و اجرا کنید .عدد ۳۷ را وارد کرده تا خروجی زیر را مشاهده نمایید.

```
Enter n:37
Yes
```

تمرین ۵ . برنامه ای بنویسید که رشته ای را خوانده ، سپس تعداد ارقام رشته را نمایش می دهد.

مراحل طراحی و اجرا:

۱. ماژول جدیدی ایجاد کرده، دستورات آن را به صورت زیر تایپ کنید:

```
s = input("Enter a string:")
count = 0
for i in s :
    if '0' <= i <= '9':
        count = count + 1
print("Count is: ", count)
```

۲. ماژول را ذخیره و اجرا کرده، اطلاعات زیر را وارد کنید تا خروجی را ببینید:

Enter a string: One equal 1 and Seven equal 7.
Count is : 2

پروژه جهت دانشجویان:

سوالات زیر را حتما در محیط برنامه نویسی پایتون تایپ کرده و اجرا کنید. سپس خروجی برنامه را بصورت عکس اسکرین شات برای ایمیل من ارسال نمایید. البته در گروه واتساپ هم میتوانید بفرستید.

Khorshid.ci@gmail.com

۱- برنامه ای بنویسید که اعداد مضرب پنج دو رقمی را نمایش دهد.

۲- برنامه ای بنویسید که یک جدول ضرب ۱۰ در ۱۰ را در خروجی نمایش می دهد.

۳- برنامه ای بنویسید که مجموع اعداد بین ۱ تا ۱۰ را محاسبه نموده و نمایش دهد.