

22. 06 . 27 月 _ for 문 원리 이해

Python for문의書き方を初心者向けに徹底解説！

和田 康宏 開発エンジニア/新卒6期/ポールド歴2年10ヶ月 ...

<https://www.bold.ne.jp/engineer-club/python-for>



Python len関数の使い方を初心者向けに徹底解説！

和田 康宏 開発エンジニア/新卒6期/ポールド歴2年10ヶ月 プログラミングをする際、文字列の長さや配列の要素数を取得して、それに応じた処理を行う場面が多く出てきます。Pythonにおいてそのような処理を行いたい場合、len関数を使うと実現できます。本記事では、len関数の使い方と使用例を解説します。まず

<https://www.bold.ne.jp/engineer-club/python-len>



Pythonのfor文によるループ処理（range, enumerate, zipなど） | note.nkmk.me

Pythonのfor文によるループ処理（繰り返し処理）について説明する。条件式が真 True である限り処理を繰り返すwhile文によるループ処理もある。以下の記事を参照。C系のプログラミング言語のfor文（forループ）はカウンタ変数（インデックス）と継続条件を使って記述する。Pythonのfor文はC系とは違い、カウンタ変数（インデックス）と継続条件を使って記述する。Pythonのfor文はC系とは違い、カウンタ変数（インデックス）と継続条件を使って記述する。Pythonのfor文はC系とは違い、カウンタ変数（インデックス）と継続条件を使って記述する。

L - 1.11.for文・break・continue

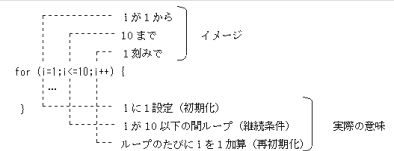
for文とwhile文の違いは主に2点あります。「カウンタ変数のスコープ」と「continueをしたときの動作」です。カウンタ変数のスコープ for文のカウンタ変数はwhile文よりスコープが狭くなります。次の例では、for文のカウンタ変数である i がスコープの範囲外で使われているため、コンパイルエラーが発生しています。

https://atcoder.jp/contests/apg4b/tasks/APG4b_1



for文

<http://www9.plala.or.jp/sgwr-t/c/sec06-2.html>



일본 파이썬 기초 설명 사이트

실행될 문장을 Block 으로 묶는다 >> 들여쓰기로 구분
함수가 실행되고, 사이에 공백으로 구분하여 하나의 “Block”으로 묶음

(形式)

```
for (初期化式; 継続条件式; 再初期化式) {  
    文;  
}
```

- 文は複合文も可。
- 複合文の文の場合には { } で囲み複合文に。
- 単文の場合には { } は省略可能。
- 再初期化式の後に「;」は入れてはいけません。

(フローチャート)

```
graph TD  
    A[初期化式] --> B[継続条件式]  
    B --> C{ }  
    C -- 真 --> D[文]  
    C -- 偽 --> E[再初期化式]  
    D --> B  
    E --> B
```

継続条件式が真の場合、ループする。偽になった場合、再初期化式を実行して次の処理を実行。

(例)

```
for (i=1; i<=10; i++) {  
    ...  
}
```

イメージ: 1が1から10まで、1刻みで

実際の意味: 1に1設定 (初期化)、1が10以下の間ループ (継続条件)、ループのたびに1を1加算 (再初期化)

- 初期化式は通常の代入文を記述。
- 継続条件式はこの式が真の間、ループを続けるという意味。
- 「~の間」であって、「~まで」ではないので注意。
- 再初期化式には「i++」ばかりではなく、「i--」や「i = i + 2」などいろいろな値をとることが可能。
- ループ変数 (使用例では i) は int型以外にも char型や double型なども可能。

2-2. リストを用いた基本的な書き方

では、上記を具体的なソースで解説します。ここでは、イテラブルオブジェクトのうち、リストを用いたソースコードの例を記載します。

```
places = ["東京", "神奈川", "千葉", "埼玉"]  
  
for place in places:  
    print(place + "に行きました")
```

実行結果

```
東京に行きました  
神奈川に行きました  
千葉に行きました  
埼玉に行きました
```

2-5. elseを用いて繰り返し終了後の処理を記述する

繰り返し終了後に何らかの処理を実行したい場合、elseを用いれば可能です。以下のソースを見てください。

```
places = ["東京", "神奈川", "千葉", "埼玉"]

for place in places:
    print(place + "に行きました")
else:
    print("繰り返し終了")
```

実行結果

```
東京に行きました
神奈川に行きました
千葉に行きました
埼玉に行きました
繰り返し終了
```

3-1. スライスを用いてオブジェクトの一部要素のみを取り出す

スライスを用いてオブジェクトの一部要素のみを取り出して繰り返し処理を実行する方法を紹介します。スライスとは、リストやタプルなどの一部分を、インデックスを指定して切り取る方法です。基本的な書き方は下記です。

```
obj[開始位置:終了位置:増分]
```

上記の書き方をする事で、開始位置から終了位置までの値を、何個おきに取得するかを指定することができます。尚、開始位置と終了位置はインデックスを指定するのですが、最初の値は0から始まることに注意してください。これを応用すれば、下記のような繰り返し処理を実現できます。

```
places = ["東京", "神奈川", "千葉", "埼玉"]

for place in places[0::2]:
    print(place + "に行きました")
```

実行結果

```
東京に行きました
千葉に行きました
```

```
# IDE (Intergrated Development Environment)

# ***** 10개를 출력하라
# 흐름제어문 (Flow control)
# - 반복
#     1 for
#     2 while
# 2 + 3 -> 이항 연산자
# 좌항 우항
# +2 -> 단항 연산자

value = 20

# range(10) -> [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
for value in range(10) :
    print("")      #마지막에 엔터키가 자동으로 추가

print ("**" , end = " " )  엔터 추가하지마라!!
```

5 X 5 단 별

```
for 1번 사용

# *****
# *****
# *****
# *****
# *****

for value in range(1, 26):
    print("", end="") # 25번 출력

    if value % 5 == 0:
        print()
```

```
for 2번 사용

# *****
# *****
# *****
# *****
# *****

for value_row in range(5): # 5번 반복 <- 세로
```

```
# 가로 반복 -> 5번
for value_col in range(5):
    print("", end="")

print() # 가로 출력 종료 후 엔터키 삽입
```

```
# *****
# *****
# *****
# *****
# *****
row = 5

# range(row) -> [ 0, 1, 2, 3, 4]
for value_row in range(row): # 5번 반복 <- 세로
    # " " 가로 반복 -> 5번? 줄마다 틀림
    # 4, 3, 2, 1, 0
    for blank_count in range(row - value_row - 1):
        # 5, 4, 3, 2, 1
        print(" ", end="")

    # "" 가로 반복 -> 5번
    for value_col in range(5):
        print("", end="")

print() # 가로 출력 종료 후 엔터키 삽입
```

```
for 문 1개로 5*5 만들기
-----
for value in range(1, 26):
    print("", end="")
    if value % 5 == 0:
        print()
```

```
#for 문 2개로 5*5 만들기
-----
for value_r in range(5):
    for value_c in range(5):
        print("", end="")
    print()
```

반토막 피라미드

```
# 총 층수 변수 설정
row = 5

# 설정해놓은 층수 만큼 (5행) 회전
for Value_r in range(row):
    # 5번 회전 내에서 반복될 것들 2가지(blank , star)

    # n번째 行일 때 "blank" 수 4,3,2,1,0 순으로 감소
    # (-1 은, 첫번째 회전시 Value_r 값이 0이기 때문에 설정 )
    for blank in range(row-Value_r-1): # blank = 4 , 3 , 2 , 1 => 4번 반복
        print(" ", end="")

    # n번째 行일 때 "star" 수 1, 2, 3, 4, 5 순으로 증가
    for star in range(0, Value_r+1): # star = 0 , 1 , 2 , 3 , 4 => 5번 반복
        print("", end="")

    # 해당 行 전부 출력 후 줄 바꾸기
    print()
```



속빈 피라미드

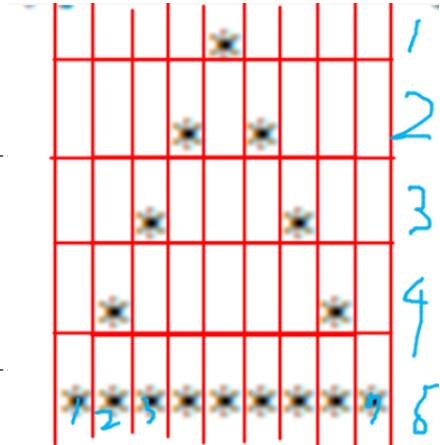
```
# 전체 행을 나타낼 for 문 작성
dan = 5
box = 9

for Floor in range(dan):
    # 첫번째 층
    # -----
    if Floor == 0:
        for blank in range(dan-Floor-1):
            print(" ", end="")
        # 조건2. 별
        for star in range(1, Floor+2):
            print("**", end="")
        print()

    # 마지막 층
    # -----
    elif Floor == 4:
        for lastfloor in range(box):
            print("**", end="")

    # 가운데 숫자
    # -----
    else:
        for blank in range(dan-Floor-1):
            print(" ", end="")

        for star in range(Floor, Floor+2):
            print("**+"**(2*star-1), end="")
        print()
```



재일이의 응용 문제

```
# 층 만들 변수 작성
floor = int(input("홀수인 층을 입력하세요 : "))
# 층의 반
division = floor // 2 + 1
#중간 이상부터 수열 맞춤 변수 작성
starplus = 1
# 층 만들 반복문 작성
for row in range (floor) :
    # 층 안에서 반복될 요소 작성
    # 조건 1. division까지는 감소
    if row < division :
        for star_up in range (division-row) :
            print("**",end=" ")
        print()
    # 조건 2. division 다음부터 증가
    else :
        # 1씩 증가 시켜 별 개수 맞추기
        starplus += 1
        for star_down in range (starplus) :
            print("**",end=" ")
        print()
```

9층

```
* * * * *
* * * *
* * *
* *
*
* *
* * *
* * * *
* * * * *
```