

۲۲ - (python) ۲۲ - ۲۲



در این قسمت به شما خواهیم آموخت که چگونه حلقه های while و for را در پایتون استفاده کنید.

در این قسمت به شما خواهیم آموخت که چگونه حلقه های while و for را در پایتون استفاده کنید.

pass

در این قسمت به شما خواهیم آموخت که چگونه حلقه های while و for را در پایتون استفاده کنید.

.Python Shell을 실행하고 New File 버튼을 클릭하여 File 메뉴에서 Python Shell을 선택합니다. IDLE.1

.Python Shell을 실행하고 Run 버튼을 클릭하여 Run 메뉴에서 Run을 선택합니다. IDLE.2

```
LetterNumber = 1
```

```
for Letter in "Howare you!":
```

```
    if Letter=="w":
```

```
        pass
```

```
        print ("Encountered w, not processed.")
```

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

```
    LetterNumber +=1
```

이 코드는 for 루프를 사용하여 문자열 "Howare you!"의 각 문자를 처리합니다. if 문을 사용하여 문자가 'w'인지 확인합니다. 만약 'w'이면 pass를 사용하여 루프를 계속 진행시킵니다. 그렇지 않으면 print 문을 사용하여 현재 문자와 인덱스를 출력합니다. 인덱스는 LetterNumber 변수를 사용하여 관리됩니다.

.Python Shell을 실행하고 Run Module 버튼을 클릭하여 Run 메뉴에서 Run을 선택합니다. IDLE.3

```
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.19
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more informatio
>>>
===== RESTART: C:/Node/456.py =====
Letter 1 is H
Letter 2 is o
Encountered w, not processed.
Letter 3 is w
Letter 4 is a
Letter 5 is r
Letter 6 is e
Letter 7 is
Letter 8 is y
Letter 9 is o
Letter 10 is u
Letter 11 is !
>>>
```

else 문을 사용하여 조건문

else 문은 if 문과 함께 사용됩니다. if 문을 사용하여 조건을 확인하고, 조건이 참이면 실행할 코드를 실행합니다. 조건이 거짓이면 else 문을 사용하여 실행할 코드를 실행합니다. else 문은 if 문과 함께 사용됩니다. if 문을 사용하여 조건을 확인하고, 조건이 참이면 실행할 코드를 실행합니다. 조건이 거짓이면 else 문을 사용하여 실행할 코드를 실행합니다.

.Python IDE를 사용하여 New File 생성 후 File 메뉴를 클릭하여 Python File을 생성합니다. 1

.Python IDE에서 Python File을 생성한 후, Python File을 실행합니다. 2

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

```
LetterNum = 1
```

```
for Letter in Value:
```

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

```
    LetterNum+=1
```

```
else:
```

```
    print("The string is blank.")
```

.Python IDE에서 Run Module 버튼을 클릭하여 Python File을 실행합니다. 3

.Python IDE에서 Python File을 실행한 후, Python File을 실행합니다. 4
Hello Python

```
===== RESTART: C:/Node/456.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  
The string is blank.  
>>>
```

.Python IDE에서 Python File을 실행한 후, Python File을 실행합니다. 5
for else Python

```
===== RESTART: C:/Node/456.py =====  
Type less than 6 characters:  
The string is blank.  
>>> |
```

```
===== RESTART: C:/Node/456.py =====  
Type less than 6 characters:  
The string is blank.  
>>> |
```

.Python IDE에서 Python File을 실행한 후, Python File을 실행합니다. 6
for else Python

while 루프를 사용하여 반복문

이 코드는 while 루프를 사용하여 반복문을 실행하는 예제입니다. while 루프는 조건이 참인 동안에 코드를 반복적으로 실행합니다. 조건이 거짓이 되면 루프를 종료합니다. while 루프의 기본 문법은 다음과 같습니다.

```
while 조건:
    # 실행할 코드
```

위와 같이 while 루프를 사용하면 조건이 참인 동안에 코드를 반복적으로 실행할 수 있습니다. 조건이 거짓이 되면 루프를 종료합니다. while 루프의 조건은 True 또는 False로 표현됩니다. True는 참을 의미하고, False는 거짓을 의미합니다. while 루프의 조건이 True일 때는 루프를 실행하고, False일 때는 루프를 종료합니다.

While Sum<5:

이 코드는 while 루프를 사용하여 Sum이 5보다 작을 때까지 반복적으로 실행하는 예제입니다. while 루프의 조건은 Sum < 5입니다. Sum이 5보다 작을 때는 루프를 실행하고, Sum이 5보다 크거나 같을 때는 루프를 종료합니다.

```
Sum = 0
while Sum < 5:
    Sum += 1
    print(Sum)
```

1. while 루프의 조건은 Sum < 5입니다. Sum이 5보다 작을 때는 루프를 실행하고, Sum이 5보다 크거나 같을 때는 루프를 종료합니다.

2. while 루프의 조건은 Sum < 5입니다. Sum이 5보다 작을 때는 루프를 실행하고, Sum이 5보다 크거나 같을 때는 루프를 종료합니다.

3. while 루프의 조건은 Sum < 5입니다. Sum이 5보다 작을 때는 루프를 실행하고, Sum이 5보다 크거나 같을 때는 루프를 종료합니다.

while 루프의 조건은 Sum < 5입니다. Sum이 5보다 작을 때는 루프를 실행하고, Sum이 5보다 크거나 같을 때는 루프를 종료합니다.

:Break

:Continue

:Else

while 루프의 조건은 Sum < 5입니다. Sum이 5보다 작을 때는 루프를 실행하고, Sum이 5보다 크거나 같을 때는 루프를 종료합니다.

1. New File

2.

Sum = 0

while Sum < 5:

print(Sum)


```

for X in range(1,11):

    print('{:>4}'.format(X), end=' ')

    while Y<=10:

        print('{:>4}'.format(X*Y),end=' ')

        Y+=1

    print()

    Y=1

```

000000 000000 000 00000 000000 00 0000 00 0000 00000 00 0000 0000 Y 0 X 000000 00 000000 00 00000 00 00000
 000 000000 X .0000000 000000000 000000 00000 00 00 0000 00000 00 0000 00000 00 0000000000 0 000000 00 000000
 0000000 00 000000 00000 000000 00000000 00000 0000 00000 00000 .0000 00000 000000 000000 Y 0 00000
 0000000 00 0000 00 0 00000 00 00 1 000000 000000 00 000000 .00000 00000 00000 000000 00 0000 9 00 1 000000
 print 000000 000000 .000000 00000000 000000 00000 00 00 000000 0000000 00 000000 00 0000 0000 00 00000000
 00000 00 0000000000 0000 00 .00000 00000000 00000 000000 00 000000 000000000 000000 00 00000 000000 00000
 00000000 00 0000 0000 00000 {4<:} 0000 .000000 000000000 00000 00000 000000000 4 00 00 0000 000000000 0000000000
 000000000 0000 00000 00000 0000 00000 00000 00000 00000 (' ')format 00000 .0000000 000000 00 00000 00
 0000000 00 range 00000 .000000 000000000 00000 000000 00 9 00 1 0000000 000000 00000 00000 000000 .000000
 00000 000000 00 00000 0000 000000000 000000000 Range 00000 00 00 000000 .000000 000000 00 000000 00 0000000
 0000 00 .00000 00000 00 0000 11 000000 00000 0000 00 00 000000000 000000 00 0 0000 1 0000 00000 0000 00
 0000 00 00 000000000 0000000 00000 .0000000 00000 000000 00000 00 00000 00 000000 000000 00 000000000 000000
 00000 00 0000000 0000 000000 00 00000 00 .0000000 0000000000 0000 0000 00 00000000 00000 00000 print 00000
 00000 00 000000000 00 000000 000000 0000000 00 00000 00 0 0000 00 00000 00000 0000 00 10* 10 00 1*1 0000
 00 00000 00 00000 0000 0000 0000 00 0000 .000000 00000 0000 00 00 00 00 0000000 0000000 00 for 00000 .00000
 00 000000000 0000 00000 0000 000000 .000000 00000 00 00000 0000 000000 print 00000 0000000000 .00000000 00000000
 00 .0000000 0000000000 000000 0000000 00000 000000000 00000 00000 00000 00 00 00 0000 .00000 0000000000
 00000000 00000000 .000000 0000 000000000 00000 00 00 00 0000000 00000 0000 .000000 00000 while 00000 000000
 00000 00000 0000 0 00000 00000000000 000000000 00000 00 00 0000 00000 00 .000000 X*Y 0000000 00 000000000
 10 00 0000000 Y 000000 0000000000 00 0000000 0000000 Y 000000 00 00000 00 while 00000 00000000 00 .00000000
 0000000 .00000000 0000 for 00000 00 000000000 0 0000 00000 while00000 00 00000000 00 000000000 00 000000 .0000
 00 000000000 000000000 000000 000000000 1 00 000000 Y 0000000 0 0000 00000 00000 0000 00 00000 000000 00000 print
 00 00 0000000 00 0000000 00000 0000000 00000 0000 .000000 0000 00 00000 0000 0000000000 0 0000 00000 while 00000
 .00000 0000 00000000 00 10 00 10 000000

00000000 00000000 0000 0000000 .00000 00000000 00 Run Module 000000 Run 00000 00 0 00000 0000000 00 00000000 .3
 :0000 0000 0000000

