

UNIT 7

출력 방법 알아보기

7 출력 방법 알아보기

» 출력 방법 알아보기

- print 하나로 여러 개의 값을 출력하는 방법과 출력 형태를 설정하는 방법을 알아보자

7.1 값을 여러 개 출력하기

» 값을 여러 개 출력하기

- print에는 변수나 값 여러 개를 ,(comma)로 구분하여 넣을 수 있음

- `print(값1, 값2, 값3)`
- `print(변수1, 변수2, 변수3)`

```
>>> print(1, 2, 3)
1 2 3
>>> print('Hello', 'Python')
Hello Python
```

- print에 변수나 값을 콤마로 구분해서 넣으면 각 값이 공백으로 띄워져서 한 줄로 출력
- 값을 여러 개 출력할 때 print 함수를 여러 번 쓰지 않아도 됨

7.1 값을 여러 개 출력하기

» sep로 값 사이에 문자 넣기

- 값 사이에 공백이 아닌 다른 문자를 넣고 싶을 때 print의 sep에 문자 또는 문자열을 지정해주면 됨(sep는 구분자라는 뜻의 separator에서 따왔음)

```
• print( 값1, 값2, sep='문자 또는 문자열')  
• print( 변수1, 변수2, sep='문자 또는 문자열')
```

```
>>> print(1, 2, 3, sep=', ')    # sep에 콤마와 공백을 지정  
1, 2, 3  
>>> print(4, 5, 6, sep=',')    # sep에 콤마만 지정  
4,5,6  
>>> print('Hello', 'Python', sep='')    # sep에 빈 문자열을 지정  
HelloPython  
>>> print(1920, 1080, sep='x')    # sep에 x를 지정  
1920x1080
```

- sep=', '처럼 콤마와 공백을 넣어주면 1, 2, 3과 같은 형태로 출력
- 공백 없이 콤마만 지정해도 됨
- sep=""처럼 빈 문자열을 지정하면 각각의 값은 서로 붙어서 출력
- sep에는 'x'와 같은 일반적인 문자도 넣을 수 있음

7.2 줄바꿈 활용하기

» 줄바꿈 활용하기

- 줄바꿈(개행)을 활용해보자
- print에 값을 여러 개 지정하면 한 줄에 모든 값이 출력

```
>>> print(1, 2, 3)
1 2 3
```

- print 한 번으로 값을 여러 줄에 출력할 수는 없을까?
- print의 sep에 개행 문자(\n)라는 특별한 문자를 지정하면 값을 한 줄에 하나씩 출력할 수 있음

```
>>> print(1, 2, 3, sep='\n')
1
2
3
```

7.2 줄바꿈 활용하기

» 줄바꿈 활용하기

- \n은 값을 다음 줄에 출력하게 만드는 제어 문자
- sep에 \n을 지정하면 1 2 3 사이에 \n이 들어가므로 1을 출력한 뒤 다음 줄에 2를 출력하고 다시 다음 줄에 3을 출력하게 됨
- 단, \n 자체는 제어 문자이므로 화면에 출력되지 않음

```
1\n2 \n3
```

7.2 줄바꿈 활용하기

» 줄바꿈 활용하기

- 참고로 \n도 문자이므로 print에 바로 넣어서 사용할 수도 있음
- 문자열 안에 \n를 넣으면 1 2 3은 세 줄로 출력되겠죠? print(1, 2, 3, sep='\n')와 결과가 같음

```
>>> print('1\n2\n3')    # 문자열 안에 \n을 사용하여 줄바꿈
1
2
3
```

- '1\n2\n3'과 같이 \n은 옆에 다른 문자나 숫자와 붙여서 씀
- \n 양 옆에 공백을 넣어버리면 공백이 그대로 출력되므로 주의해야 함

7.2 줄바꿈 활용하기

» 참고 | 제어 문자

- 제어 문자는 화면에 출력되지는 않지만 출력 결과를 제어한다고 해서 제어 문자라 부름
- 제어 문자는 \로 시작하는 이스케이프 시퀀스
- \n: 다음 줄로 이동하며 개행이라고도 부름
- \t: 탭 문자, 키보드의 Tab 키와 같으며 여러 칸을 띄움
- \\: \ 문자 자체를 출력할 때는 \를 두 번 써야 함

7.2 줄바꿈 활용하기

» end 사용하기

- print는 기본적으로 출력하는 값 끝에 \n을 붙임
- print를 여러 번 사용하면 값이 여러 줄에 출력됨
- IDLE의 소스 코드 편집 창에 입력한 뒤 실행

print_multiple.py

```
print(1)
print(2)
print(3)
```

실행 결과

```
1
2
3
```

7.2 줄바꿈 활용하기

» end 사용하기

- print를 여러 번 사용해서 print(1, 2, 3)처럼 한 줄에 여러 개의 값을 출력할 수는 없을까?
- print의 end에 빈 문자열을 지정해주면 됨

- `print(값, end='문자 또는 문자열')`
- `print(변수, end='문자 또는 문자열')`

print_multiple_end.py

```
print(1, end='')    # end에 빈 문자열을 지정하면 다음 번 출력이 바로 뒤에 오게 됨
print(2, end='')
print(3)
```

실행 결과

123

7.2 줄바꿈 활용하기

» end 사용하기

- end="와 같이 end에 빈 문자열을 지정하면 1, 2, 3이 세 줄로 출력되지 않고 한 줄로 붙어서 출력됨
- 기본적으로 print의 end에 \n이 지정된 상태인데 빈 문자열을 지정하면 강제로 \n을 지워주기 때문임
- end는 현재 print가 끝난 뒤 그 다음에 오는 print 함수에 영향을 줌
- 1 2 3 사이를 띄워주고 싶다면 end에 공백 한 칸을 지정하면 됨

```
print(1, end=' ')    # end에 공백 한 칸 지정
print(2, end=' ')
print(3)
```

실행 결과

1 2 3

- print의 sep, end에 제어 문자, 공백 문자 등을 조합하면 다양한 형태로 값을 출력할 수 있음
- 자신의 상황에 맞게 선택해서 사용하면 됨