

f-string 4주차_03_02

한 동 대 학 교
김경미 교수

학습목표

2

- ▶ f-string 사용 시 장점 이해하기
- ▶ f-string 활용하기

f-string, 3.6

3

```
x = 1
```

```
y = 2
```

```
print('f-string 사용할 때')
```

```
print(f'{x} + {y}는 {x + y}입니다.')
```

1 2 3

```
f-string 사용할 때  
1 + 2는 3 입니다.
```



f-string, 비교 예제, 연산식

4

```
x = 1  
y = 2
```

```
print(" f-string 사용하지 않을 때")  
print ( x, '+', y, '는', x+y, '입니다.' )
```

```
print(" f-string 사용할 때")  
print(f"{x} + {y}는 {x + y}입니다.")
```

```
print(" f-string 아닐 때 중괄호 표현 결과")  
print("{x} + {y}는 {x + y}입니다.")
```

```
f-string 사용하지 않을 때  
1 + 2 는 3 입니다.
```

```
f-string 사용할 때  
1 + 2는 3 입니다.
```

```
f-string 아닐 때 중괄호 표현 결과  
{x} + {y}는 {x + y}입니다.
```

```
> █
```

f-string, 변수 치환

```
for mon in range(1, 13):  
    print(f'2200년 {mon} 월')
```

{mon} → 숫자
(변수 값)

```
2200년 1 월  
2200년 2 월  
2200년 3 월  
2200년 4 월  
2200년 5 월  
2200년 6 월  
2200년 7 월  
2200년 8 월  
2200년 9 월  
2200년 10 월  
2200년 11 월  
2200년 12 월
```

f-string, 중괄호 출력

for mon in range(1, 13):
 print(f'2200년 {{mon}} 월')

2200년 {1} 월
2200년 {2} 월
2200년 {3} 월
2200년 {4} 월
2200년 {5} 월
2200년 {6} 월
2200년 {7} 월
2200년 {8} 월
2200년 {9} 월
2200년 {10} 월
2200년 {11} 월
2200년 {12} 월

f-string, 함수 호출

```
word = "PYTHON f-string"
```

```
print(f"{word}는 {len(word)}글자입니다.")
```

```
print(f"대문자로는 {word.upper()}이고, 소문자로는 {word.lower()}입니다.")
```

```
print(f"swapcase {word.swapcase()}입니다.")
```

스왑케이스

PYTHON f-string는 15글자입니다.

대문자로는 PYTHON F-STRING이고, 소문자로는 python f-string입니다.

swapcase python F-STRING입니다.

Format identifier

8

Format identifier	using	example
%s	String	%10s
%d	Number, integer	%5d %5.3d
%f	Number, float	%.3f %7.2f

f-string, 소수점 표현

```
pi = 3.141592653589
```

```
print(f'{pi:10.10f}')
```

```
print(f'{pi:10.4f}')
```

```
print(f'{pi:10.2f}')
```

```
print(f'{pi:12.7f}')
```

```
print(f'{pi:.4f}')
```

3.1415926536

3.1416

3.14

3.1415927

3.1416

100

space

Time	State	Count	Frequency
2200	T ₁	1	0.1111
2200	T ₁	1	0.1111
2200	T ₁	1	0.1111
2200	T ₁	2	0.2222
2200	T ₁	2	0.2222
2200	T ₁	2	0.2222
2200	T ₁	3	0.3333
2200	T ₁	3	0.3333
2200	T ₁	3	0.3333

f-string, 정렬 기호

11

```
print('왼쪽 정렬')  
s1 = 'left'  
print(f' | {s1:<12} | \n' ) ➔4
```

```
print('가운데 정렬')  
s2 = 'middle'  
print(f' | {s2:^12} | \n' ) ➔6
```

```
print('오른쪽 정렬')  
s3 = 'right'  
print(f' | {s3:>12} | ') ➔5
```

왼쪽 정렬
| left 8 |

가운데 정렬
| 3 middle 3 |

오른쪽 정렬
| 2 right |

f-string, 정렬 예제

```
ch_a = '5'  
int_a = 5
```

```
print('1234567890')
```

```
print(f'{ch_a:>5}') # >는 오른쪽정렬
```

```
print(f'{ch_a:<5}') # <는 왼쪽정렬
```

```
print(f'{ch_a:^5}') # ^는 가운데정렬
```

★

```
print(f'{int_a:0<5}')
```

 # <는 왼쪽정렬, 빈자리를 0으로 채운다

1234567890

5

5

5

50000

□

강의 요약

13

- ▶ f-string 사용 시 장점 이해하기
- ▶ f-string 활용하기
 - ▶ 문자열 맨 앞에(따옴표앞) f를 붙인다
 - ▶ 사용하고 싶은 변수, 값, 함수등을 중괄호 안에 넣는다

목표 달성 질문

14

- ▶ f-string을 사용하면 어떤 장점이 있는지 말해보세요
- ▶ Format identifier, 실수를 표현할 때 어떤 기호를 쓰나요?

감사합니다