

Lesson 4

문자열 포매팅



문자열 포매팅(Formatting)?

데이터를 원하는 형태의 문자열로 만드는 것.

1000 → "1,000"

11.234 → "11.2"

- 너비 지정, 정렬 설정, 콤마 표시, 소수 자릿수 지정, 백분율 표시, 공백 채우기,...

%타입기호

format()

f-strings

코드

```
name="손우주"  
age=25
```

원하는 출력 결과=>

손우주는 25세입니다.

```
print(name+"는 "+str(age)+"세입니다.")
```

```
print("{}는 {}세입니다.".format(name,age))
```



나는 Elsa입니다.

[illegible]

2	3								
---	---	--	--	--	--	--	--	--	--

[illegible]

코드

```
# 손우주는 25세입니다.
```

```
name="손우주"
```

```
age=25
```

"%타입기호" % 변수

"{}".format(변수)

f"{변수}"

```
print("%s는 %d세입니다." % (name, age))
```

```
print("{}는 {}세입니다.".format(name, age))
```

```
print(f"{name}는 {age}세입니다.")
```

- %타입 기호 : "%채울문자 정렬 너비 .소수자릿수 타입기호"%**변수**
- format() : "{:채울문자 정렬 너비 .소수자릿수}".format(**변수**)
- f-strings : f"{**변수**:채울문자 정렬 너비 .소수자릿수}"



타입 기호: %s(문자열), %d(정수), %f(실수)

```
format_spec ::= [[fill]align][sign][#][0][width][grouping_option][.precision][type]
fill        ::= <모든 문자>
align       ::= "<" | ">" | "=" | "^"
sign        ::= "+" | "-" | " "
width       ::= digit+
grouping_option ::= "_" | ","
precision   ::= digit+
type        ::= "b" | "c" | "d" | "e" | "E" | "f" | "F" | "g" | "G" | "n" | "o" | "s"
```

<https://python.flowdas.com/library/string.html#formatspec>

코드

```
name="손우주"
```

```
age=25
```

```
# 너비 지정
```

```
print("---너비---")
```

```
print("%10s" % name) # 문자는 기본 오른쪽 정렬
```

```
print("{:10}".format(name))
```

```
print(f"{name:10}")
```

```
# 정렬 지정 : <, >, ^
```

```
print("---정렬---")
```

```
print("{:<10}".format(name))
```

```
print("{:>10}".format(name))
```

```
print("{:^10}".format(name))
```

```
---너비---  
          손우주  
손우주  
손우주
```

```
---정렬--  
손우주  
          손우주  
          손우주
```


코드

소수 자릿수 지정~~~> 타입기호 f를 입력해야 함

```
num = 3.141
```

```
print("---자릿수---")
```

```
print("%f" % num) 3.141000
```

```
print("%.2f" % num) 3.14
```

```
print("%10.2f" % num) 3.14
```

```
print("{:.2f}".format(num)) 3.14
```

```
print(f"{num:.2f}") 3.14
```

코드

```
name="손우주"
```

```
age=25
```

```
# 빈자리 채움 지정
```

```
print("%010d" % age)
```

```
print("{:>10} ".format(age))
```

```
print("{:_>10} ".format(name))
```

```
0000000025
+++++++25
_____손우주
```

```
# 콤마, 백분율
```

```
print("{:,} ".format(123456))
```

```
print("{:.1%} ".format(0.015))
```

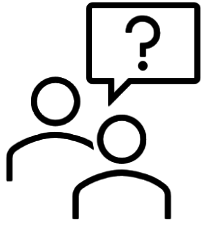
```
123,456
1.5%
```



%는 숫자 데이터에만 사용 가능하고 0으로만 채움 가능

format, f-strings는 반드시 너비와 정렬이 설정되어 있어야 채움 가능

단, 숫자 0으로 채울 경우 정렬은 지정하지 않아도 됨.



1. 포매팅은 데이터를 원하는 형태의 []로 만드는 것이다.
2. "{:03}".format(5)의 결과는?
3. 아래 코드의 결과는?
`result=1.2345`
`f"{result:5.1f}"`