



문자열 자료형

Index

[문자열을 만드는 법](#)

[따옴표 안에 따옴표 포함](#)

[여러 줄인 문자열을 변수에 대입하고 싶을 때](#)

[문자열 연산하기](#)

[문자열 인덱싱과 슬라이싱](#)

[문자열 인덱싱](#)

[문자열 인덱싱 활용](#)

[문자열 슬라이싱](#)

[슬라이싱으로 문자열 나누기](#)

[문자열 포매팅](#)

[포맷 코드와 숫자 함께 사용](#)

[format 함수를 사용한 포매팅](#)

[f 문자열 포매팅](#)

[문자열 관련 함수들](#)

[문자 개수 세기 - count](#)

[위치 알려주기1 -find](#)

[위치 알려주기2 -index](#)

[문자열 삽입 -join](#)

[소문자를 대문자로 바꾸기 -upper](#)

[대문자를 소문자로 바꾸기 -lower](#)

[왼쪽 공백 지우기 -lstrip](#)

[오른쪽 공백 지우기 -rstrip](#)

[양쪽 공백 지우기 -strip](#)

[문자열 바꾸기 -replace](#)

[문자열 나누기 -split](#)

문자열을 만드는 법

- 큰따옴표로 양쪽 둘러싸기
- 작은따옴표로 양쪽 둘러싸기
- 큰따옴표 3개를 연속으로 써서 양쪽 둘러싸기
- 작은따옴표 3개를 연속으로 써서 양쪽 둘러싸기

따옴표 안에 따옴표 포함

- 문자열에 작은따옴표 포함
- 문자열에 큰따옴표 포함
- 역슬래시를 사용

여러 줄인 문자열을 변수에 대입하고 싶을 때

- \n 삽입
- 연속된 작은따옴표 3개 또는 큰따옴표 3개 사용

문자열 연산하기

파이썬만의 장점

- 문자열 더해서 연결하기

```
head = "Python"
tail = " is fun!"
head + tail
>>> 'Python is fun!'
```

- 문자열 곱하기

```
a = "Python"
a * 2
>>> 'pythonpython'
```

- 문자열 곱하기 응용

```
print("=" * 50)
>>> =====...
```

- 문자열 길이 구하기

```
a = "Life is too short"
len(a)
>>> 17
```

문자열 인덱싱과 슬라이싱

문자열 인덱싱

```
a = "Life is too short, You need Python"
a[3]
>>> 'e'
```

문자열 인덱싱 활용

```
a = "Life is too short, You need Python"
a[0]
>>> 'L'
a[12]
>>> 's'
a[-1]
>>> 'n'
```

-를 붙이면 뒤에서부터다. (-0은 0과 똑같다)

문자열 슬라이싱

```
a = "Life is too short, You need Python"
b = a[0] + a[1] + a[2] + a[3]
b
>>> 'Life'
```

간단하게

```
a = "Life is too short, You need Python"
a[0:4]
>>> 'Life'
```

a[시작_번호:끝_번호]를 지정할 때 끝 번호의 문자는 포함하지 않는다.

a[:]는 모든 문자열, -도 사용 가능

슬라이싱으로 문자열 나누기

```
a = "20230331Rainy"
date = a[:8]
weather = a[8:]
date
>>> '20230331'
weather
>>> 'Rainy'
```

```
a = "20230331Rainy"
year = a[:4]
day = a[4:8]
weather = a[8:]
year
>>> '2023'
day
>>> '0331'
weather
>>> 'Rainy'
```

문자열 포매팅

- 숫자 바로 대입

```
"I eat %d apples." % 3
>>> 'I eat 3 apples'
```

- 문자열 바로 대입

```
"I eat %s apples." % "five"
>>> 'I eat five apples'
```

- 숫자 값을 나타내는 변수로 대입

```
number = 3
"I eat %d apples." % number
>>> 'I eat 3 apples'
```

- 2개 이상의 값 넣기

```
number = 10
day = "three"
"I ate %d apples. so I was sick for %s days." % (number, day)
>>> 'I ate 10 apples. so I was sick for three days.'
```

코드	설명
%s	문자열 (String)
%c	문자 1개 (character)
%d	정수 (Integer)
%f	부동소수 (floating-point)
%o	8진수
%x	16진수
%%	Literal % (문자 % 자체)

```
"I have %s apples " % 3
>>> 'I have 3 apples'
"rate is %s" % 3.234
>>> 'rate is 3.234'
```

포맷 코드와 숫자 함께 사용

- 정렬과 공백

```
"%10s" % "hi"
>>> '          hi'
```

- 소수점 표현

```
"%0.4f" % 3.42134234
>>> '3.4213'
```

format 함수를 사용한 포매팅

- 숫자 바로 대입

```
"I eat {0} apples".format(3)
>>> 'I eat 3 apples'
```

- 문자열 바로 대입

```
"I eat {0} apples".format("five")
>>> 'I eat five apples'
```

- 숫자 값을 가진 변수로 대입

```
number = 3
"I eat {0} apples".format(number)
>>> 'I eat 3 apples'
```

- 2개 이상의 값 넣기

```
number = 10
day = "three"
"I ate {0} apples. so I was sick for {1} days.".format(number, day)
>>> 'I ate 10 apples. so I was sick for three days.'
```

- 이름으로 넣기

```
"I ate {number} apples. so I was sick for {day} days.".format(number=10, day=3)
>>> 'I ate 10 apples. so I was sick for 3 days.'
```

- 인덱스와 이름을 혼용해서 넣기

```
"I ate {0} apples. so I was sick for {day} days.".format(10, day=3)
>>> 'I ate 10 apples. so I was sick for 3 days.'
```

- 왼쪽 정렬

```
"{0:<10}".format("hi")
>>> 'hi          '
```

- 오른쪽 정렬

```
"{0:>10}".format("hi")
>>> '          hi'
```

- 가운데 정렬

```
"{0:^10}".format("hi")
>>> '      hi      '
```

- 공백 채우기

```
"{0:=^10}".format("hi")
>>> '====hi===='
"{0:!
```

- 소수점 표현하기

```
y = 3.42134234
"{0:0.4f}".format(y)
>>> '3.4213'
"{0:10.4f}".format(y)
>>> '      3.4213'
```

- { 또는 } 문자 표현하기

```
"{{ and }}".format()  
>>> '{ and }'
```

f 문자열 포매팅

```
name = '홍길동'  
age = 30  
f'나의 이름은 {name}입니다. 나이는 {age}입니다.'  
>>> '나의 이름은 홍길동입니다. 나이는 30입니다.'
```

```
age = 30  
f'나는 내년이면 {age + 1}살이 된다.'  
>>> '나는 내년이면 31살이 된다.'
```

```
d = {'name': '홍길동', 'age': 30}  
f'나의 이름은 {d["name"]}입니다. 나이는 {d["age"]}입니다.'  
>>> '나의 이름은 홍길동입니다. 나이는 30입니다.'
```

```
f'{"hi":<10}' # 왼쪽 정렬  
>>> 'hi      '  
f'{"hi":>10}' # 오른쪽 정렬  
>>> '      hi'  
f'{"hi":^10}' # 가운데 정렬  
>>> '    hi    '
```

```
f'{"hi"::=^10}' # 가운데 정렬하고 '='문자로 공백 채우기  
>>> '====hi===='  
f'{"hi":!<10}' # 왼쪽 정렬하고 '!'문자로 공백 채우기  
>>> 'hi!!!!!!!!'
```

```
y = 3.42134234  
f'{y:0.4f}' # 소수점 4자리까지만 표현
```



```
>>> '3.4213'
f'{y:10.4f}' # 소수점 4자리까지 표현하고 총 자리수를 10으로 맞춤
>>> '      3.4213'
```

```
f'{{ and }}'
>>> '{ and }'
```

문자열 관련 함수들

문자 개수 세기 - count

```
a = "hobby"
a.count('b')
>>> 2
```

위치 알려주기1 -find

```
a = "Python is the best choice"
a.find('b')
>>> 14
a.find('k')
>>> -1 # 존재하지 않음
```

위치 알려주기2 -index

```
a = "Life is too short"
a.index('t')
>>> 8
a.index('k')
>>> Error # 찾는 문자가 없음
```

문자열 삽입 -join

```
", ".join('abcd')
>>> 'a,b,c,d'
```

소문자를 대문자로 바꾸기 -upper

```
a = "hi"  
a.upper()  
>>> 'HI'
```

대문자를 소문자로 바꾸기 -lower

```
a = "HI"  
a.lower()  
>>> 'hi'
```

왼쪽 공백 지우기 -lstrip

```
a = " hi "  
a.lstrip()  
>>> 'hi '
```

오른쪽 공백 지우기 -rstrip

```
a = " hi "  
a.rstrip()  
>>> ' hi'
```

양쪽 공백 지우기 -strip

```
a = " hi "  
a.strip()  
>>> 'hi'
```

문자열 바꾸기 -replace

```
a = "Life is too short"  
a.replace("Life", "Your leg")
```

```
>>> 'Your leg is too short'
```

문자열 나누기 -split

```
a = "Life is too short"
a.split()
>>> ['Life', 'is', 'too', 'short']
b = "a:b:c:d"
b.split(':')
>>> ['a', 'b', 'c', 'd']
```

모두 변수의 값이 변하는 게 아니라 변하게 리턴한 것이다.