



8장. 문자열 관리

파이썬 정복



Contents

❖ 목차

- 1. 문자열 분리
- 2. 문자열 메서드
- 3. 포매팅



1. 문자열 분리

❖ 첨자

- 문자의 위치
- 대괄호와 첨자 적어 문자열 구성하는 개별 문자 읽음
- 앞뒤 양쪽에서 읽을 수 있음

0	1	2	3	4	5
p	y	t	h	o	n
-6	-5	-4	-3	-2	-1

index

```
s = "python"
print(s[2])
print(s[-2])
```

실행결과

t
o

- for문으로 순회하여 개별 문자 순서대로 꺼냄

stringindex

```
s = "python"
for c in s:
    print(c, end = ',')
```

실행결과

p,y,t,h,o,n,



1. 문자열 분리

❖ 슬라이스

- 범위 지정하여 부분 문자열을 추출
- **[begin:end:step]**
 - 시작, 끝, 중간값을 지정

slice

```
s = "python"
print(s[2:5])
print(s[3:])
print(s[:4])
print(s[2:-2])
```

실행결과

```
tho
hon
pyth
th
```



1. 문자열 분리

- 일정 형식 가진 문자열에서 원하는 정보만 추출할 수 있음

slice2

```
file = "20171224-104830.jpg"
print("촬영 날짜 : " + file[4:6] + "월 " + file[6:8] + "일")
print("촬영 시간 : " + file[9:11] + "시 " + file[11:13] + "분")
print("확장자 : " + file[-3:])
```

실행결과

촬영 날짜 : 12월 24일
촬영 시간 : 10시 48분
확장자 : jpg



2. 문자열 메서드

❖ 메서드

- 클래스에 소속된 함수
- 객체에 대해 특화된 작업 수행

❖ 검색

- **len** 함수
 - 문자열의 길이를 조사

find

```
s = "python programming"
print(len(s))
print(s.find('o'))
print(s.rfind('o'))
print(s.index('r'))
print(s.count('n'))
```

실행결과

18
4
9
8
2



2. 문자열 메서드

- **find 메서드**

- 인수로 지정한 문자 또는 부분 문자열의 위치 조사

- **rfind 메서드**

- 뒤에서 검색 시작

- **count 메서드**

- 특정 문자 개수



2. 문자열 메서드

❖ 조사

■ in 구문

- 특정 문자 유무 여부 조사

in

```
s = "python programming"  
print('a' in s)  
print('z' in s)  
print('pro' in s)  
print('x' not in s)
```

실행결과

```
True  
False  
True  
True
```



2. 문자열 메서드

함수	설명
isalpha	모든 문자가 알파벳인지 조사한다.
islower	모든 문자가 소문자인지 조사한다.
isupper	모든 문자가 대문자인지 조사한다.
isspace	모든 문자가 공백인지 조사한다.
isalnum	모든 문자가 알파벳 또는 숫자인지 조사한다.
isdecimal	모든 문자가 숫자인지 조사한다.
isdigit	모든 문자가 숫자인지 조사한다.
isnumeric	모든 문자가 숫자인지 조사한다.
isidentifier	명칭으로 쓸 수 있는 문자로만 구성되어 있는지 조사한다.
isprintable	인쇄 가능한 문자로만 구성되어 있는지 조사한다.



2. 문자열 메서드

❖ 변경

■ lower 메서드 / upper 메서드

- 각기 영문자를 전부 소문자 / 대문자로 바꿈

lower

```
s = "Good morning. my love KIM."  
print(s.lower())  
print(s.upper())  
print(s)  
  
print(s.swapcase())  
print(s.capitalize())  
print(s.title())
```

실행결과

```
good morning. my love kim.  
GOOD MORNING. MY LOVE KIM.  
Good morning. my love KIM.  
gOOD MORNING. MY LOVE kim.  
Good morning. my love kim.  
Good Morning. My Love Kim.
```



2. 문자열 메서드

- 문자열 자체를 변경하는 것은 아님
- **lstrip** / **rstrip** / **strip** 메서드
 - 왼쪽 / 오른쪽 / 양측 공백을 제거

strip

```
s = " angel  "
print(s + "님")
print(s.lstrip() + "님")
print(s.rstrip() + "님")
print(s.strip() + "님")
```

실행결과

```
angel  님
angel  님
angel님
angel님
```



2. 문자열 메서드

❖ 분할

■ split 메서드

- 구분자를 기준으로 문자열을 분할

split

```
s = "짜장 짬뽕 탕숙"
print(s.split())

s2 = "서울->대전->대구->부산"
city = s2.split("->")
print(city)
for c in city:
    print(c, "찍고", end = ' ')
```

실행결과

```
['짜장', '짬뽕', '탕숙']
['서울', '대전', '대구', '부산']
서울 찍고 대전 찍고 대구 찍고 부산 찍고
```

■ splitlines 메서드

- 개행 문자나 파일 구분자 등 기준으로 문자열 잘라 리스트로 만들



2. 문자열 메서드

■ join 메서드

- 문자열의 각 문자 사이에 다른 문자열 끼워넣음

join

```
s = "._."  
print(s.join("대한민국"))
```

실행결과

대._.한._.민._.국



2. 문자열 메서드

❖ 대체

■ replace 메서드

- 특정 문자열을 찾아 다른 문자열로 대체
- 첫 번째 인수 : 검색할 문자열 지정
- 두 번째 인수 : 바꿀 문자열 지정

replace

```
s = "독도는 일본땅이다. 대마도도 일본땅이다."  
print(s)  
print(s.replace("일본", "한국"))
```

실행결과

```
독도는 일본땅이다. 대마도도 일본땅이다.  
독도는 한국땅이다. 대마도도 한국땅이다.
```

■ just 메서드

- 문자열을 특정 폭에 맞추어 정렬



2. 문자열 메서드

■ center 메서드

- 중앙 정렬

center

```
traveler = """강나루 건너서\n밀밭 길을\n\n구름에 달 가듯이\n가는 나그네\n\n길은 외줄기\n남도 삼백리\n\n술 익는 마을마다\n타는 저녁놀\n\n구름에 달 가듯이\n가는 나그네"""  
poet = traveler.splitlines()  
for line in poet:  
    print(line.center(30))
```

실행결과

```
강나루 건너서  
밀밭 길을  
  
구름에 달 가듯이  
가는 나그네  
  
길은 외줄기  
남도 삼백리  
  
술 익는 마을마다  
타는 저녁놀  
  
구름에 달 가듯이  
가는 나그네
```



3. 포매팅

❖ 포매팅

- 문자열 사이사이에 다른 정보 삽입하여 조립하는 기법

stringcat

```
price = 500  
print("궁금하면 " + str(price) + "원!")
```

실행결과

궁금하면 500원!



format

```
price = 500  
print("궁금하면 %d원!" % price)
```



3. 포매팅

표식	설명
%d	정수
%f	실수
%s	문자열
%C	문자 하나
%h	16진수
%O	8진수
%%	% 문자

format2

```
month = 8
day = 15
anni = "광복절"
print("%d월 %d일은 %s이다." % (month, day, anni))
```

실행결과

8월 15일은 광복절이다.



3. 포매팅

■ 서식

- %[-]폭[.유효자리수]서식

align

```
price = [30, 13500, 2000]
for p in price:
    print("가격:%d원" % p)
for p in price:
    print("가격:%7d원" % p)
```

실행결과

```
가격:30원
가격:13500원
가격:2000원
가격:   30원
가격: 13500원
가격:  2000원
```



3. 포매팅

■ 폭에 – 지정하여 왼쪽 정렬

numalign

```
price = [30, 13500, 2000]
for p in price:
    print("가격:%-7d원" % p)
```

실행결과

```
가격:30      원
가격:13500   원
가격:2000    원
```

■ . 기호로 소수점 이하 표시 자리수 설정

precision

```
pie = 3.14159265
print("%10f" % pie)
print("%10.8f" % pie)
print("%10.5f" % pie)
print("%10.2f" % pie)
```

실행결과

```
3.141593
3.14159265
3.14159
3.14
```





Thank You !

파이썬 정복

