

---

Chapter 1

# 문자열 다루기



# 문자열 다루기

파이썬을 문자열을 다루는 다양한 방법을 제공합니다.

1. 문자열 인덱싱, 슬라이싱
2. 문자열 함수(미리 만들어진 기능)

## 인덱싱

- 문자열[index] 형식으로 문자열 이상~지정한 위치(index)의 미만 문자를 뽑아낼 수 있음
- 첫 문자의 인덱스가 0
- 음수는 맨 뒤의 문자부터 의미(-1은 마지막 문자)

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

1	"Python"[0]
	'p'
1	"Python"[3]
	'h'

# 문자열 슬라이싱

## 슬라이싱

- [start:stop]를 이용하면 부분 문자열을 빼 낼 수 있음
  - Start이상~ stop미만
  - start 위치의 문자는 포함하지만 stop 위치의 문자는 포함하지 않음
  - [:]형식에서도 인덱스를 음수로 지정할 수 있음

```
1 str_ = 'Python'
```

```
1 str_[0:6]
```

'Python'

```
1 str_[:]
```

'Python'

```
1 str_[0:]
```

'Python'

```
1 str_[:6]
```

'Python'

-1은 맨 마지막 문자  
(뒤에서)를 의미

```
1 str_[-6:-1]
```

'Pytho'

```
1 str_[-6:]
```

'Python'

```
1 str_[:-1]
```

'Pytho'

# 문자열 슬라이싱

## 슬라이싱

- [start:stop:step] 형식은 매 step번째 아이템을 추출해 줌
  - Start이상~ stop미만
  - start, stop, step 은 생략될 수 있음
  - 만일 start와 stop이 생략되면 [::step] 형식이 됨
  - 이 형식은 문자열 뿐만 아니라 리스트, 튜플 등에서도 사용할 수 있음

1	'0123456789' [::2]
---	--------------------

'02468'

1	'0123456789' [::3]
---	--------------------

'0369'

1	'0123456789' [2:8:2]
---	----------------------

'246'

1	'0123456789' [9::-2]
---	----------------------

'97531'

-1은 맨 마지막 문자  
(뒤에서)를 의미

1	'0123456789' [::-2]
---	---------------------

'97531'

1	'0123456789' [-1::-2]
---	-----------------------

'97531'

1	'0123456789' [-1:-7:-2]
---	-------------------------

'975'

# 문자열 함수

방법	설명
<code>" " 또는 ''</code>	문자열을 만듭니다.
<code>+</code>	문자열을 연결합니다.
<code>len("문자열")</code>	문자열의 길이를 반환합니다.
<code>[start : stop]</code>	문자열을 start 위치부터 stop 위치까지 자릅니다. stop은 포함 안 됩니다.
<code>[start : stop : step]</code>	문자열을 start 위치부터 stop 위치까지 매 step 마다 반환합니다. stop은 포함 안 됩니다.
<code>split('delimiter')</code>	문자열을 delimiter로 잘라 리스트로 반환합니다.
<code>'delimiter'.join(["str1", ...])</code>	문자열 리스트를 delimiter로 연결합니다.
<code>capitalize()</code>	첫 문자를 대문자로, 나머지 문자를 소문자로 바꿔줍니다.
<code>upper, lower()</code>	문자열을 모두 대문자(upper) 또는 소문자(lower)로 바꿉니다.
<code>startswith(), endswith()</code>	특정 문자로 시작하는지와 끝나는지를 식별해서 논리(True/False)값을 반환합니다.
<code>find(), index()</code>	특정 문자의 인덱스를 반환합니다.
<code>isalnum(), isalpha(), isnumeric(), isdecimal()</code>	이 문자열이 숫자인지, 문자인지 판별해 줍니다.
<code>replace(old, new)</code>	old 문자를 new 문자로 치환합니다.



# Chapter 5

## 수고하셨습니다