



SW멘토링 프로그램

비전공자를 위한 파이썬 기초
한림대학교 영어영문학과 정민



파이썬 기본문법 - 자료형

- 숫자열
 - ex) $1+3$ / $10 * 7$ / etc...
- 문자열
 - ex) 'Hello world!' / $a + b$ / etc...
- 리스트
 - ex) $[1, 3, 5, 7, 9]$ / etc...
- 튜플
 - ex) $(1, 2, 'a', 'b')$ / etc...
- 딕셔너리
 - ex) $\{ 'name' : 'Jeong', 'birth' : '0618' \}$ / etc...

숫자형

- 정수형
 - $a = 123$
 - $b = 45$
 - $c = -6$
- 실수형
 - $a = 1.2$
 - $b = -3.45$

숫자형의 사칙연산

- 사칙연산

```
In [5]: a = 10  
        b = 5
```

```
In [6]: a + b
```

```
Out [6]: 15
```

```
In [7]: a - b
```

```
Out [7]: 5
```

```
In [8]: a * b
```

```
Out [8]: 50
```

```
In [9]: a / b
```

```
Out [9]: 2.0
```

숫자형의 사칙연산

- **a**의 **b** 제곱을 나타내는 연산자

```
In [10]: a ** b
```

```
Out [10]: 100000
```

숫자형의 사칙연산

- 나눗셈 후 나머지를 반환하는 연산자

```
In [11]: a % b
```

```
Out [11]: 0
```

- 나눗셈 후 몫을 반환하는 연산자

```
In [12]: a // b
```

```
Out [12]: 2
```

숫자형 실습

1. 국어 : 90, 수학: 80, 영어: 70 의 평균을 구하세요.
2. 2의 12제곱을 구하세요.

문자열

문자열 만들기

In [15]: "큰따옴표 안에 문자열 넣기"

Out [15]: '큰따옴표 안에 문자열 넣기'

In [16]: '작은따옴표 안에 문자열 넣기'

Out [16]: '작은따옴표 안에 문자열 넣기'

In [17]: ""큰따옴표 3개 안에 문자열 넣기""

Out [17]: '큰따옴표 3개 안에 문자열 넣기'

In [18]: '''작은따옴표 3개 안에 문자열 넣기'''

Out [18]: '작은따옴표 3개 안에 문자열 넣기'

문자열 안에 따옴표 넣기

문자열 안에 따옴표 넣기

```
In [19]: "'큰따옴표'를 넣으려면 작은따옴표 안에 넣기'
```

```
Out [19]: "'큰따옴표"를 넣으려면 작은따옴표 안에 넣기'
```

```
In [20]: "'작은따옴표'를 넣으려면 큰따옴표 안에 넣기"
```

```
Out [20]: "'작은따옴표'를 넣으려면 큰따옴표 안에 넣기"
```

```
In [21]: '따옴표 앞에 #'백슬래쉬#'를 넣어도 됩니다.'
```

```
Out [21]: "따옴표 앞에 '백슬래쉬'를 넣어도 됩니다."
```

문자열 여러줄 넣기

```
In [22]: multilines = '''작은따옴표를  
이용한  
여러줄의  
글쓰기  
입니다.'''
```

```
In [23]: multiline = """큰따옴표를  
이용한  
여러줄의  
글쓰기  
입니다."""
```

```
In [24]: multi = "이스케이프코드를\n이용한\n여러줄의\n글쓰기\n입니다."
```

이스케이프 코드

- 프로그래밍할 때 사용할 수 있도록 미리 정의해 둔 “문자조합”
 - `\n`: 줄바꿈
 - `\t`: 수평탭
 - `\\`: 문자 `\`
 - `\'`: 문자열 안의 작은따옴표
 - `\"`: 문자열 안의 큰따옴표

문자열의 더하기 / 곱하기

- 문자열 합치기

```
In [25]: a = '문자열을'  
        b = '합쳐봅시다.'
```

```
In [26]: a + b
```

```
Out [26]: '문자열을합쳐봅시다.'
```

- 문자열 반복하기

```
In [27]: a = '문자열을 반복해보아요!'
```

```
In [28]: a * 3
```

```
Out [28]: '문자열을 반복해보아요!문자열을 반복해보아요!문자열을 반복해보아요!'
```

문자열 반복 응용

- 문자열 반복하기를 이용하여 사각형을 만들어보세요.



문자열 인덱싱

- 각 문자의 번호를 붙인 것.
- 파이썬의 첫번째 숫자는 '0'

```
In [32]: a = 'Life is too short, you need python'
```

```
In [33]: a[3]
```

```
Out [33]: 'e'
```

```
In [34]: a[-0]
```

```
Out [34]: 'L'
```

```
In [35]: a[-1]
```

```
Out [35]: 'n'
```

문자열 슬라이싱

- 문자열 인덱싱처럼 단순히 한 문자만을 뽑아내는 것이 아니라 단어를 뽑아낼 수 있는 것.
- ex) `a[0:4]` = a에서 0,1,2,3을 뽑으세요

문자열 인덱싱 VS. 문자열 슬라이싱

- 문자열 인덱싱

```
In [ ]: a = 'Life is too short, you need python'
```

```
In [37]: word = a[0] + a[1] + a[2] + a[3]
```

```
In [38]: word
```

```
Out [38]: 'Life'
```

- 문자열 슬라이싱

```
In [40]: a[0:4]
```

```
Out [40]: 'Life'
```


문자열 슬라이싱

- 원하는 부분을 뽑아낼 수 있다.

```
In [42]: a[19:]
```

```
Out [42]: 'you need python'
```

```
In [43]: a[:17]
```

```
Out [43]: 'Life is too short'
```

- 시작, 끝 번호를 생략하면 처음부터 뽑아내거나, 끝까지 뽑아낸다.

```
In [44]: a[:]
```

```
Out [44]: 'Life is too short, you need python'
```

- 기호를 사용할 수 있다.

```
In [45]: a[19:-7]
```

```
Out [45]: 'you need'
```

슬라이싱으로 문자열 나누기

```
In [46]: a = '20190324sunny'
```

```
In [47]: date = a[:8]
```

```
In [48]: weather = a[8:]
```

```
In [49]: date
```

```
Out [49]: '20190324'
```

```
In [50]: weather
```

```
Out [50]: 'sunny'
```

(슬라이싱을 이용한) 문자열 바꾸기

```
In [57]: a = "Jupyternotebook"
```

```
In [58]: a[:3] + "y" + a[4:]
```

```
Out [58]: 'Jupyternotebook'
```

문자열 포매팅

- 문자열 내의 특정한 값을 바꿔야 할 경우가 있을 때 이것을 가능하게 해주는 것이 바로 문자열 포매팅이다.

문자열 포매팅

- 숫자 대입

```
In [59]: "나는 %d 개의 파이를 먹었다" % 5
```

```
Out [59]: '나는 5 개의 파이를 먹었다'
```

- 문자열 대입

```
In [60]: "나는 %s 하는 것을 좋아한다." % "파이썬 공부"
```

```
Out [60]: '나는 파이썬 공부 하는 것을 좋아한다.'
```

- 변수 대입

```
In [61]: temp = 5  
"오늘의 온도는 %d 도 입니다." % temp
```

```
Out [61]: '오늘의 온도는 5 도 입니다.'
```

문자열 포매팅

- 2개 이상의 값 넣기

```
In [62]: task = 3  
        sle = 2  
        "나는 %d 개의 과제를 끝내고, %d 일 동안 잠을 잤다." % (task, sle)
```

```
Out [62]: '나는 3 개의 과제를 끝내고, 2 일 동안 잠을 잤다.'
```

- % 문자 사용하기

```
In [63]: hu = 74  
        "오늘의 습도는 %d%% 입니다." % hu
```

```
Out [63]: '오늘의 습도는 74% 입니다.'
```

.format() 함수 사용하기

- 포맷 함수 사용하여 포매팅하기

```
In [64]: "나는 {0} 개의 사과를 먹었다.".format(3)
```

```
Out [64]: '나는 3 개의 사과를 먹었다.'
```

- 2개 이상의 값 넣기

```
In [65]: "나는 {0}개의 귤을 먹고, {1}일 동안 잤다.".format(3,5)
```

```
Out [65]: '나는 3개의 귤을 먹고, 5일 동안 잤다.'
```

문자열 관련 함수

- 문자열 길이(len())

```
In [30]: a = 'Hello World! I love python world!'
```

```
In [31]: len(a)
```

```
Out [31]: 33
```

- 특정 문자 개수 세기(.count())

```
In [67]: a = "shoot"  
a.count('o')
```

```
Out [67]: 2
```


문자열 관련 함수

- 특정 문자 위치 찾기 (find, index)

```
In [68]: a = 'Python is powerful'
```

```
In [69]: a.find('i')
```

```
Out [69]: 7
```

```
In [73]: a.find('z')
```

```
Out [73]: -1
```

```
In [71]: a.index('i')
```

```
Out [71]: 7
```

```
In [72]: a.index('z')
```

```
ValueError                                Traceback (most recent call last)
<ipython-input-72-c8e03cca-fal c> in <module>
----> 1 a.index('z')
```

```
ValueError: substring not found
```

문자열 관련 함수

- 문자열 삽입하기(.join())

```
In [75]: ','.join('abcd')
```

```
Out [75]: 'a,b,c,d'
```

- 대문자 만들기(.upper())

```
In [77]: a = 'abcde'  
a.upper()
```

```
Out [77]: 'ABCDE'
```

- 소문자 만들기(.lower())

```
In [78]: b = 'ABCDE'  
b.lower()
```

```
Out [78]: 'abcde'
```

문자열 관련 함수

- 공백 지우기(.strip())

```
In [80]: a = '  good  '  
a.lstrip()
```

```
Out [80]: 'good  '
```

```
In [81]: a = '  good  '  
a.rstrip()
```

```
Out [81]: '  good'
```

```
In [82]: a = '  good  '  
a.strip()
```

```
Out [82]: 'good'
```

문자열 관련 실습

- 문자열 바꾸기(.replace())

```
In [83]: a = 'Hello world. Python is fun'  
         a.replace('Python', 'Game')
```

```
Out [83]: 'Hello world. Game is fun'
```

- 문자열 나누기(.split())

```
In [84]: a.split()
```

```
Out [84]: ['Hello', 'world.', 'Python', 'is', 'fun']
```

```
In [85]: a.split('.')
```

```
Out [85]: ['Hello world', ' Python is fun']
```

문자열 실습

1. 전화번호에서 하이픈을 제거하고 출력하세요
 - a. 010-1111-2222
2. 전화번호의 마지막 4자리만 출력하세요
 - a. 010-1234-5678
3. 소문자 a를 대문자 A로 변경하세요
 - a. abcdef1234aa32a
4. a:b:c:d 를 a#b#c#d로 바꾸세요
 - a. replace 함수 사용
 - b. split과 join 함수 사용