



Overview

Pythonic Code

최성철 교수  
Director of TEAMLAB

# 파이썬 스타일 코드 (Pythonic Code)

# Pythonic Code

예시: 여러 단어들을 하나로 붙일 때

Python shell

```
>>> colors = ['red', 'blue', 'green', 'yellow']
>>> result = ''
>>> for s in colors:
    result += s
```

# Pythonic Code

예시: 여러 단어들을 하나로 붙일 때

```
>>> colors = ['red', 'blue', 'green', 'yellow']  
>>> result = ''.join(colors)  
>>> result  
'redbluegreenyellow'
```

Python shell

Good Case

Source: <http://python.net/~goodger/projects/pycon/2007/idiomatic/handout.html>

---

# Pythonic Code

- 파이썬 스타일의 코딩 기법
- 파이썬 특유의 문법을 활용하여 효율적으로 코드를 표현함
- 고급 코드를 작성 할 수록 더 많이 필요해짐

---

# Pythonic Code

- **Split & Join**
- **List Comprehension**
- **Enumerate & Zip**

---

# Why Pythonic Code?

남 코드에 대한 이해도

많은 개발자들이 python 스타일로 코딩한다

효율

단순 for loop append보다 list가 조금 더 빠르다  
익숙해지면 코드도 짧아진다.

간지

쓰면 웬지 코드 잘 짜는 것처럼 보인다

# Supplement Materials



**Fluent Python – 전문가를 위한 파이썬**

**(한빛미디어, 2016)**

**Part 2 부분 – 시퀀스 & 딕셔너리**



**Effective Python - 파이썬 코딩의 기술**

**(길벗, 2016)**



# Supplement Materials

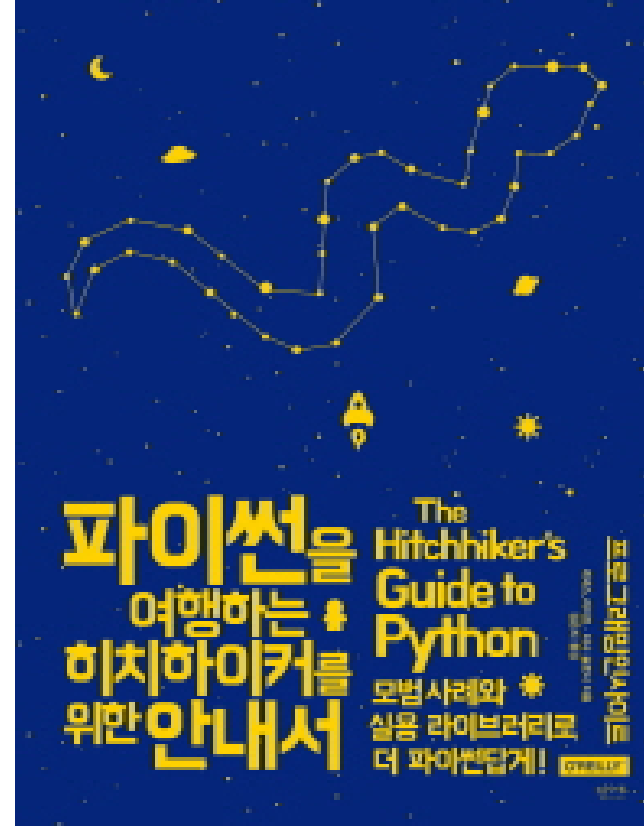


## 10 Tips for Pythonic Code (M. Kennedy, 2016)

[https://www.youtube.com/watch?v=\\_O23jlXsshs](https://www.youtube.com/watch?v=_O23jlXsshs)

# Supplement Materials

## The Hitchhiker's Guide to Python



<http://python-guide-kr.readthedocs.io/ko/latest/>

<http://docs.python-guide.org/en/latest/writing/style/>



**Human knowledge belongs to the world.**

Split & Join

Pythonic Code

최성철 교수  
Director of TEAMLAB

01100  
00110

A 3D rendering of a white robot head with large, expressive eyes and a hand holding a glowing blue binary code '01100' and '00110'. The robot is white with black accents on its joints and fingers. The background is white.

Split

# Split 함수

## - String Type의 값을 나눠서 List 형태로 변환

Python shell

```
>>> items = 'zero one two three'.split() # 빈칸을 기준으로 문자열 나누기
>>> print (items)
['zero', 'one', 'two', 'three']
>>> example = 'python,jquery,javascript' # ","을 기준으로 문자열 나누기
>>> example.split(",")
['python', 'jquery', 'javascript']
>>> a, b, c = example.split(",")
# 리스트에 있는 각 값을 a,b,c 변수로 unpacking
>>> example = 'cs50.gachon.edu'
>>> subdomain, domain, tld = example.split('.')
# "."을 기준으로 문자열 나누기 → Unpacking
```

# Join 함수

## - String List를 합쳐 하나의 String으로 반환할 때 사용

```
>>> colors = ['red', 'blue', 'green', 'yellow']
>>> result = ''.join(colors)
>>> result
'redbluegreenyellow'
>>> result = ' '.join(colors) # 연결 시 빈칸 1칸으로 연결
>>> result
'red blue green yellow'
>>> result = ', '.join(colors) # 연결 시 ", "으로 연결
>>> result
'red, blue, green, yellow'
>>> result = '-'.join(colors) # 연결 시 "-"으로 연결
>>> result
'red-blue-green-yellow'
```

Python shell



**Human knowledge belongs to the world.**



The background of the slide is a 3D rendering of a white robot head. The robot has large, circular eyes with green and yellow lenses. Its right hand is raised, holding a cluster of blue binary digits (0s and 1s). The robot's head is split vertically: the left side is dark grey, and the right side is white. The text is overlaid on the dark grey side.

List Comprehension

Pythonic Code

최성철 교수  
Director of TEAMLAB

---

# List comprehensions

- 기존 List 사용하여 간단히 다른 List를 만드는 기법
- 포괄적인 List, 포함되는 리스트라는 의미로 사용됨
- 파이썬에서 가장 많이 사용되는 기법 중 하나
- 일반적으로 for + append 보다 속도가 빠름

# List comprehensions (1/4)

```
>>> result = []
>>> for i in range(10):
...     result.append(i)
...
>>> result
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Python shell

General Style

```
>>> result = [i for i in range(10)]
>>> result
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> result = [i for i in range(10) if i % 2 == 0]
>>> result
[0, 2, 4, 6, 8]
```

Python shell

List Comprehension

# List comprehensions (2/4)

Python shell

```
>>> word_1 = "Hello"
>>> word_2 = "World"
>>> result = [i+j for i in word_1 for j in word_2]
           # Nested For loop
>>> result
['HW', 'Ho', 'Hr', 'Hl', 'Hd', 'eW', 'eo', 'er',
 'el', 'ed', 'lW', 'lo', 'lr', 'll', 'ld', 'lW',
 'lo', 'lr', 'll', 'ld', 'oW', 'oo', 'or', 'ol', 'od']
```

# List comprehensions (3/4)

Python shell

```
>>> case_1 = ["A", "B", "C"]
>>> case_2 = ["D", "E", "A"]
>>> result = [i+j for i in case_1 for j in case_2]
>>> result
['AD', 'AE', 'AA', 'BD', 'BE', 'BA', 'CD', 'CE', 'CA']
>>> result = [i+j for i in case_1 for j in case_2 if not(i==j)]
# Filter: i랑 j과 같다면 List에 추가하지 않음
>>> result
['AD', 'AE', 'BD', 'BE', 'BA', 'CD', 'CE', 'CA']
>>> result.sort()
>>> result
['AD', 'AE', 'BA', 'BD', 'BE', 'CA', 'CD', 'CE']
```

# List comprehensions (4/4)

```
>>> words = 'The quick brown fox
jumps over the lazy dog'.split()
# 문장을 빈칸 기준으로 나눠 list로 변환

>>> print (words)
['The', 'quick', 'brown', 'fox',
'jumps', 'over', 'the', 'lazy', 'dog']
>>>
>>> stuff = [[w.upper(), w.lower(),
len(w)] for w in words]

# list의 각 element들을 대문자, 소문자, 길이
로 변환하여 two dimensional list로 변환
```

```
>>> for i in stuff:
...     print (i)
...
['THE', 'the', 3]
['QUICK', 'quick', 5]
['BROWN', 'brown', 5]
['FOX', 'fox', 3]
['JUMPS', 'jumps', 5]
['OVER', 'over', 4]
['THE', 'the', 3]
['LAZY', 'lazy', 4]
['DOG', 'dog', 3]
```

# Two dimensional vs One dimensional

Python shell

```
>>> case_1 = ["A","B","C"]
>>> case_2 = ["D","E","A"]
['AD', 'AE', 'AA', 'BD', 'BE', 'BA', 'CD', 'CE', 'CA']
>>> result = [i+j for i in case_1 for j in case_2]
>>> result
>>> result = [ [i+j for i in case_1] for j in case_2]
>>> result
```



**Human knowledge belongs to the world.**





**Enumerate & Zip**

**Pythonic Code**

**최성철 교수**  
**Director of TEAMLAB**

**Enumerate**

# Enumerate

## - List의 element를 추출할 때 번호를 붙여서 추출

```
>>> for i, v in enumerate(['tic', 'tac', 'toe']):  
# list의 있는 index와 값을 unpacking  
...     print (i, v)  
...  
0 tic  
1 tac  
2 toe  
  
>>> mylist = ["a","b","c","d"]  
>>> list(enumerate(mylist)) # list의 있는 index와 값을 unpacking하여 list로 저장  
[(0, 'a'), (1, 'b'), (2, 'c'), (3, 'd')]  
>>> {i:j for i,j in enumerate('Gachon University is an academic institute  
located in South Korea.'.split())}  
# 문장을 list로 만들고 list의 index와 값을 unpacking하여 dict로 저장  
{0: 'Gachon', 1: 'University', 2: 'is', 3: 'an', 4: 'academic', 5: 'institute',  
6: 'located', 7: 'in', 8: 'South', 9: 'Korea.'}
```

Python shell

Zip

# Zip

## 두 개의 list의 값을 병렬적으로 추출함

Python shell

```
>>> alist = ['a1', 'a2', 'a3']
>>> blist = ['b1', 'b2', 'b3']
>>> for a, b in zip(alist, blist): # 병렬적으로 값을 추출
...     print (a,b)
...
a1 b1
a2 b2
a3 b3

>>> a,b,c =zip((1,2,3), (10,20,30), (100,200,300)) #각 tuple의 같은 index 끼리 묶음
(1, 10, 100) (2, 20, 200) (3, 30, 300)

>>> [sum(x) for x in zip((1,2,3), (10,20,30), (100,200,300))]
# 각 Tuple 같은 index를 묶어 합을 list로 변환
[111, 222, 333]
```

# Enumerate & Zip

Python shell

```
>>> alist = ['a1', 'a2', 'a3']
>>> blist = ['b1', 'b2', 'b3']
>>>
>>> for i, (a, b) in enumerate(zip(alist, blist)):
...     print (i, a, b) # index alist[index] blist[index] 표시
...
0 a1 b1
1 a2 b2
2 a3 b3
```



**Human knowledge belongs to the world.**