

빠르게 활용하는 파이썬3 프로그래밍

Chapter 1. 파이썬3 개요 및 설치

파이썬이란...



- 1991년 귀도 반 로섬(Guido van Rossum)이 발표.
- 다른 개발언어에 비해 문법이나 구조가 간결하여 생산성이 좋음.
- 사용자 층이 점점 넓어지는 중.



파이썬의 특징

- 가독성
 - 간결하고 가독성이 좋습니다.
 - 코드블럭을 들여쓰기(indentation)로 구분.
- 풍부한 라이브러리
- 접착성
- 무료
- 유니코드
- 동적타이핑



파이썬의 특징

- 가독성
- **풍부한 라이브러리**
 - 매우 광범위한 라이브러리가 기본으로 포함되어 있음.
 - 외부 라이브러리가 풍부하며 확장이 쉬움.
- 접착성
- 무료
- 유니코드
- 동적타이핑



파이썬의 특징

- 가독성
- 풍부한 라이브러리
- **접착성**
 - 쉽게 라이브러리를 추가할 수가 있습니다.
 - 파이썬에서 C로 구현된 부분을 사용할 수도 있으며, C에서 파이썬을 사용할 수도 있습니다.
- 무료
- 유니코드
- 동적타이핑



파이썬의 특징

- 가독성
- 풍부한 라이브러리
- 접착성
- **무료**
 - 파이썬은 파이썬 소프트웨어 재단(Python Software Foundation)에서 관리하고 있음.
 - 무료와 다름없는 Python Software Foundation License.
- 유니코드
- 동적타이핑



파이썬의 특징

- 가독성
- 풍부한 라이브러리
- 접착성
- 무료
- **유니코드**
 - 파이썬에서의 문자열들은 모두 유니코드입니다.
 - 우리가 한글, 한자 등을 표현하려고 특별한 노력을 할 필요가 없습니다.
- 동적타이핑



파이썬의 특징

- 가독성
- 풍부한 라이브러리
- 접착성
- 무료
- 유니코드
- **동적타이핑**
 - 런타임 시에 타입 체크를 하는 동적타이핑을 지원하며,
 - 메모리 관리 자동으로 합니다.



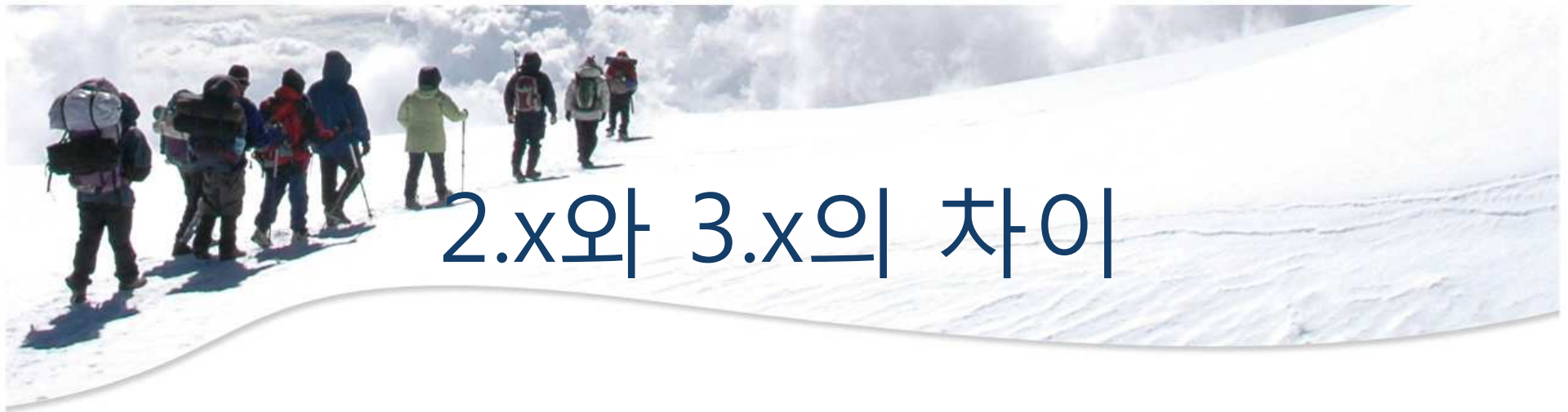
파이썬의 종류

- Cpython
- Jython
- IronPython
- PyPy



파이썬이 쓰이는 프로젝트들

- 알게 모르게 파이썬이 사용되는 프로젝트들이 많습나다만, 유명한 것들만 예를 들어 보겠습니다.
 - BitTorrent, MoinMoin, Scons, Trac, Yum
 - CherryPy, Django
 - GIMP, Maya, Paint Shop Pro
 - Youtube.com, Google Groups, Google Maps, Gmail

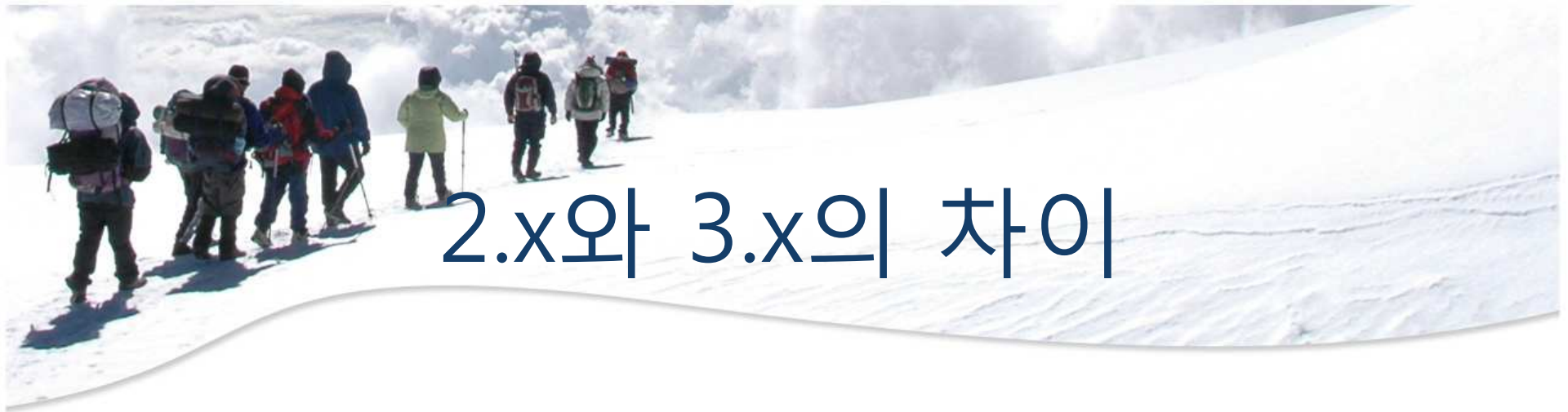


2.x와 3.x의 차이

- **print가 함수로 변경.**
 - 2.x style :

```
>>> print "welcome to", "python3k"  
welcome to python3k
```
 - 3.0 style :

```
>>> print("welcome to", "python3k")  
welcome to python3k
```
- long 자료형이 없어지고 int로 통일.
- 'int / int'의 결과가 float으로 처리.
- String, Unicode 체계 변경.



2.x와 3.x의 차이

- print가 함수로 변경.
- **long** 자료형이 없어지고 int로 통일.
 - 2.x style :

```
>>> type(2**31)
<type 'long'>
>>> sys.maxint
2147483647
```
 - 3.0 style :

```
>>> type(2**31)
<class 'int'>
>>> type(2**40)
<class 'int'>
```
- 'int / int'의 결과가 float으로 처리.
- String, Unicode 체계 변경.



2.x와 3.x의 차이

- print가 함수로 변경.
- long 자료형이 없어지고 int로 통일.
- **'int / int'의 결과가 float으로 처리.**
 - 2.x style :

```
>>> 3/2  
1
```
 - 3.0 style :

```
>>> 3/2  
1.5  
>>> type(2/2)  
<class 'float'>
```
- String, Unicode 체계 변경.



2.x와 3.x의 차이

- print가 함수로 변경.
- long 자료형이 없어지고 int로 통일.
- 'int / int'의 결과가 float으로 처리.
- **String, Unicode 체계 변경.**
 - 2.x style :

```
>>> type('가')  
<type 'str'>  
>>> type(u'가')  
<type 'unicode'>
```
 - 3.0 style :

```
>>> type('가')  
<class 'str'>  
>>> type('가'.encode('cp949'))  
<class 'bytes'>
```




들여쓰기

- 들여쓰기(indentation)는 파이썬 문법의 가장 큰 특징입니다.
 - 가독성을 높이지만, 오류가 일어나지 않도록 조심해야 합니다.
 - 코드블럭1
 - [TAB]코드블럭2
 - 코드블럭3
 - [TAB]코드블럭4



소스코드 인코딩

- 파이썬에서는 # 이후는 주석으로 인식합니다.
- 그러나 다음과 같이,
소스코드 부분에서 사용될 경우,
소스코드 인코딩을 지정하는 용도로 사용됩니다.
 - # coding: latin-1
 - # -*- coding: utf-8 -*-

기타 파이썬 문법

- 한 라인에 여러 구문이 올 경우에는 세미콜론(;)을 사용해야 합니다.
 - `>>> a = 1; b = 2`
- 들여쓰기(indentation)이 중요하지만, 문장이 아직 안 끝난 경우에는 들여쓰기를 안 해도 문법오류가 나지 않습니다.
 - `>>> a = (1 +
[TAB]2 +
3 +
[TAB][S][S]4)
>>> a
10`

자료형 및 연산자

변수명

- 문자, 숫자, 밑줄(_)로 구성됩니다. 숫자는 처음에 나올 수 없습니다.
- 대소문자를 구분합니다.
- 예약어 사용 불가.

ex.

```
>>> friend = 10
>>> Friend = 1
>>> friend
10
>>> Friend
1
```



수치

- int
 - `>>> 10, 0x10, 0o10, 0b10`
`(10, 16, 8, 2)`
- float
 - `>>> type(3.14), type(314e-2)`
`(<class 'float'>, <class 'float'>)`
- complex
 - `>>> x=3-4j`
`>>> type(x), x.imag, x.real, x.conjugate()`
`(<class 'complex'>, -4.0, 3.0, (3+4j))`
- 연산자
 - `+, -, *, /, //, %, **, =`



문자

- 'string', "string",
""줄바꿈도
그대로 적용됩니다""
- Escape 문자

Escape 문자 사용예	의미
Wn	개행(줄바꿈)
Wt	탭
Wr	캐리지 리턴
W0	널(Null)
WW	문자 'W'
W'	단일 인용부호(')
W"	이중 인용부호(")



문자

- +, * 연산자
 - >>> 'py' 'thon'
'python'
 - >>> 'py' * 3
'pypypy'
- 인덱싱 & 슬라이싱
 - >>> 'python'[0]
'p'
 - >>> 'python'[5]
'n'
 - >>> 'python'[1:4]
'yth'
 - >>> 'python'[-2:]
'on'

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



리스트

- 리스트는 쉽게 값들의 나열이라고 생각하시면 됩니다.
- 또한, 인덱싱과 슬라이싱도 가능합니다.
 - ```
>>> colors = ['red', 'green', 'gold']
```

```
>>> colors
```

```
['red', 'green', 'gold']
```
- `append`, `insert`, `extend`, + 연산자를 이용해 값(들)을 추가할 수 있습니다.

```
>>> colors.append('blue')
>>> colors
['red', 'green', 'gold', 'blue']
```

```
>>> colors.insert(1, 'black')
>>> colors
['red', 'black', 'green', 'gold', 'blue']
```

```
>>> colors.extend(['white', 'gray'])
>>> colors
['red', 'black', 'green', 'gold', 'blue',
'white', 'gray']
```



- Index를 이용해 어디에 원하는 값이 있는지 확인 가능합니다.
  - ```
>>> colors.index('red')
```

```
0
```
- Count를 이용해 원하는 값의 개수를 알 수 있으며, pop을 이용해 값을 뽑아낼 수도 있습니다.
 - ```
>>> colors
```

```
['red', 'black', 'green', 'gold', 'blue', 'white', 'gray', 'red']
```

```
>>> colors.count('red')
```

```
2
```

```
>>> colors.pop()
```

```
'red'
```

```
>>> colors
```

```
['red', 'black', 'green', 'gold', 'blue', 'white', 'gray']
```
- Remove로 값을 삭제하거나, sort로 정렬을 할 수도 있습니다.



# 세트

- 세트(set)는 수학시간에 배운 집합과 동일합니다.
- 세트는 리스트와 마찬가지로 값들의 모임이며, 순서가 없습니다.
- 제공되는 메소드는 리스트와 유사하며, 추가적으로 교집합(intersection)과 합집합(union)이 제공됩니다.
  - ```
>>> a = {1, 2, 3}
```
 - ```
>>> b = {3, 4, 5}
```
  - ```
>>> a.union(b)
```

합집합

```
{1, 2, 3, 4, 5}
```
 - ```
>>> a.intersection(b)
```

# 교집합  

```
{3}
```



# 튜플

- 튜플(tuple)은 리스트와 유사하나, 읽기전용입니다.
- 읽기 전용인 만큼 제공되는 함수도 리스트에 비해 적지만, 속도는 그만큼 빠릅니다.
- 튜플에서 제공되는 메소드는 count, index 정도입니다.
- 튜플을 이용하면 'C'와 같은 언어에서는 변수가 하나 더 필요한 swap 예제를 다음과 같이 간단하게 해결할 수도 있습니다.
  - ```
>>> a, b = 1, 2
>>> print(a, b)
1 2
>>> a, b = b, a
>>> print(a, b)
2 1
```




부울

- 부울(bool)은 참과 거짓을 나타내는 자료형으로, 가능한 값은 True와 False 뿐입니다.
- 주로 부울은 부울 값들 간의 논리연산이나, 수치들간의 비교연산의 결과로 사용됩니다.
- 비교연산자의 종류는 '크다(>)', '작다(<)', '같다(==)', '다르다(!=)', '같거나크다(>=)', '같거나작다(<=)' 가 있습니다.
- 논리연산자는 'and(&)', 'or(|)', 'not' 이 있습니다.
'and'는 두 값이 모두 참이어야만 참을 반환하고, 'or'는 둘 중 하나의 값만 참이면 참을 반환합니다. 또한 'not'은 반대의 값을 반환합니다.
논리연산자 역시 비교연산자와 함께 제어문의 조건에서 주로 사용됩니다.



과제

1. 학과에 대한 리스트를 만들어서 변수명 dept에 넣고 과목명에 대한 리스트를 만들어서 변수명 subject에 넣는다. 이 두 개의 리스트를 결합하고 그 결과를 university 라고한다. 마지막으로 university를 출력한다.



과제

2. 다음과 같은 리스트를 생성한 후 예제와 같은 리스트로 변경합니다.

```
>>> animals = ['tiger', 'dog', 'cat', 'cow', 'rabbit']
>>> ?
>>> animals
['tiger', 'dog', 'cat', 'cow', 'rabbit', 'lion']
>>> ?
>>> animals
['tiger', 'dog', 'cat', 'cow', 'rabbit', 'lion', ['dove', 'eagle', 'parrot']]
>>> ?
>>> animals
['tiger', 'dog', 'ant', 'cat', 'cow', 'rabbit', 'lion', ['dove', 'eagle', 'parrot']]
>>> ?
'ant'
>>> animals
['tiger', 'dog', 'cat', 'cow', 'rabbit', 'lion', ['dove', 'eagle', 'parrot']]
```



과제

3. `>>> stu = "My student number is 20161234"` 다음과 같이 변수에 본인의 학번값을 넣어 저장한 후에 슬라이싱하여 'student number' 와 학번을 출력한다.(출력결과는 변수에 넣지말고 슬라이싱만 이용하시오)



과제 제출 방법

- ▶ 과제캡처 후 워드or한글파일에 첨부/정리하여 제출
- ▶ 파일형식 : [과제번호1]_이름(조이름)_학번
 - ▶ 제출 형식 어길 시 감점처리
- ▶ 제출 : dbcyy1@gmail.com로 제출
- ▶ 제출기간 : 3월15일 화요일 23시59분까지



수고하셨습니다.