Ch1. Meet Python: *Everyone Loves Lists*

- summarized by 전승일(2014/9/14)
- source code: http://www.headfirstlabs.com/books/hfpython/
- support site: http://python.itcarlow.ie

What's to like about Python? (python의 매력)

- PC, 맥, 휴대기기, 휴대폰, 웹, 대형서버에서 빠르고 쉽게 코드 작성, GUI 생성 가능
- Programming Languages Ranking: http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

Install Python 3

- www.python.org에서 download 및 설치

```
python3 -V                  # for OS-X, Linux
c:\Python31\python.exe -V   # for Windows
```

Use IDLE to help learn Python

- IDLE은 python 구문 및 내장함수를 알고 보기쉽게 표시 해 줌
- **BIF**(내장함수) : built-in function

```
dir(__builtins__)
help(input)
```

Work effectively with IDLE

- 탭 완성, 코드 문장 기억, 이전 문장 불러와서 편집하기, IDLE 환경 설정 바꾸기
- **Eclipse** + **PyDev** 사용하는 방법도 있음

```
1. www.eclipse.org/downloads 에서 Standard version 선택
2. 압축 풀고 eclipse 실행
3. http://pydev.org/manual_101_install.html 참고하여 pydev 추가
   * Help > Install New Software
   * Location: http://pydev.org/updates
   * PyDev 선태 후 화면에 따라 진행
   * 환경설정: General > Editors > Text Editors : Tab을 space로 변경, Tab = 4
     PyDev > Interpreters > Python Interpreter : Advanced Auto-Config
```

Deal with complex data

- 영화광 데이터

  ```
  영화제목 및 개요
      주연배우
          조연배우
  ```

- list 사용하여 관리

Create simple Python lists

- **list** : 데이터의 집합체로 콤마로 구분하고 대괄호로 시작과 끝을 둘러 싼다

  ```
  movies = ["The Holy Grail", "The Life of Brian", "The Meaning of Life"]
  ```

Lists are like arrays

- list는 항목으로 서로다른 유형의 데이터를 포함할 수 있지만, array는 동일한 유형의 데이터만 포함할 수 있음

```
movies = ["The Holy Grail", 1975, "Terry Jones & Terry Gilliam", 91]
```

Add more data to your list

```
cast = ["Cleese", 'Palin', 'Jones', "Idle"]
len(cast)
cast.append("Gilliam")
cast.pop()
cast.extend(["Gilliam", "Chapman"])
cast.remove("Chapman")
cast.insert(0, "Chapman")
```

Work with your list data

```
movies = ["The Holy Grail", "The Life of Brian"]

print(movies[0])
print(movies[1])
```

- list의 항목이 늘어나면? 코드도 늘어나야...

For loops work with lists of any size

```
movies = ["The Holy Grail", "The Life of Brian"]

for each_item in movies:
    print(each_item)

또는

count = 0
while count < len(movie):
    print(movies[count])
    count = count + 1
```

- list의 크기에 무관하게 코드가 늘어나지 않음

Store lists within lists

- 배우 데이터를 저장하기 위해 list 사용, 조연 배우를 위해 또 다른 list 사용: **nested list**

```
movies = ["The Holy Grail", 1975, "Terry Jones & Terry Gilliam", 91,
                ["Graham Chapman",
                        ["Michael Palin", "John Cleese", "Terry Gilliam", "Eric

for each_item in movies:
    print(each_item)
```

Check a list for a list

- data가 list인지 아닌지 확인하는 function

```
isinstance(data, list)
```

Complex data is hard to process

```
for each_item in movies:
    if isinstance(each_item, list):
        for nested_item in each_item:
            print(nested_item)
    else:
        print(each_item)
```

Handle many levels of nested lists

```
for each_item in movies:
    if isinstance(each_item, list):
```

```
        for nested_item in each_item:
            if isinstance(nested_item, list):
                for deeper_item in nested_item:
                    print(deeper_item)
            else:
                print(nested_item)
    else:
        print(each_item)
```

Don’t repeat code; create a function

- code가 반복되면 function을 만든다

Create a function in Python

```
    def function_name(arguments):
        code_suites
```

Recursion to the rescue!

- define a recursive function

```
def print_lol(the_list):
    for each_item in the_list:
        if isinstance(each_item, list):
            print_lol(each_item)
        else:
            print(each_item)

print_lol(movies)
```

Additional Information

- 파이썬 2.7 vs 파이썬 3 차이 (added by 좌승룡) https://wikidocs.net/743

- Install Python3 using brew on Mac (added by 표준영) http://brew.sh/

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master
brew install python3
```