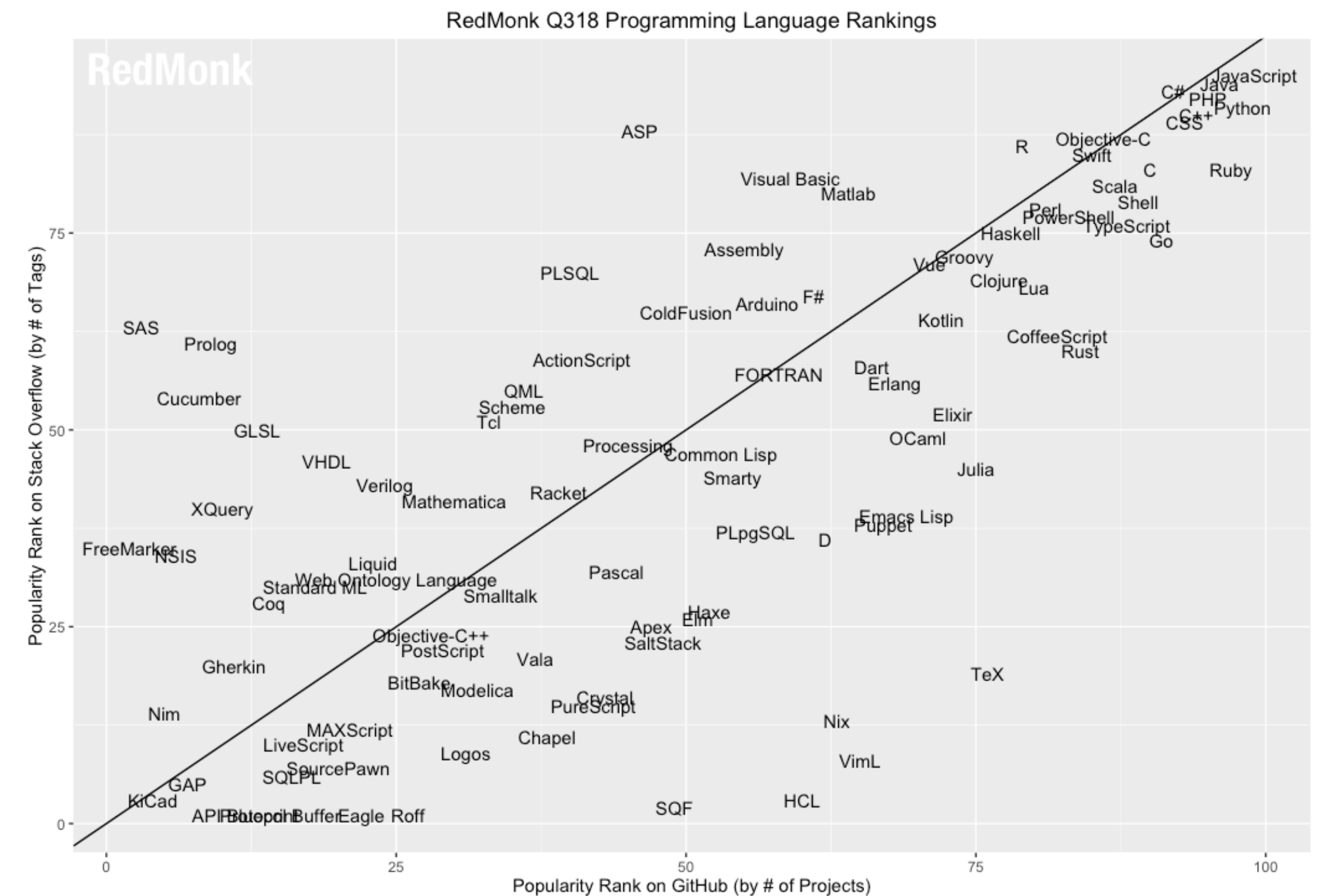


Web Studio 2019

2. Python, flask(1)

Contents

1. Python
2. Pip
3. Virtualenv
4. Flask



Python

프로그래밍 언어를 배울 땐

1. Variable, types, list, dictionary
2. if
3. for
4. function
5. class (or structure, interface)
6. File read, write

Python

Variable, types

```
a = 5
c = a ** 3 # c: 125
b = 'Sangkeun'
print(b[0:3:2]) # 'Sn'
print(b[1:5]) # 'angk'
print(b[:3]) # 'San'
print(b + ' Kim') # 'Sangkeun Kim'
d = 5.123
e = 987654321123456789123456
```

Python

List

```
l = [1, 2, 3, 4, 3]
print(len(l)) # 5
l.append(0)    # [1, 2, 3, 4, 3, 0]
l.sort()      # [0, 1, 2, 3, 3, 4]
print(l[3])   # 3
l = l + [4, 2, 1]
print(l)      # [0, 1, 2, 3, 4, 4, 4, 2, 1]
```

Python

List 2d

```
l = [[1, 2], [3, 4, 5]]  
print(l[0][1])      # 2  
print(l[1][1])      # 4  
print(len(l))       # 2  
print(len(l[1]))    # 3
```

Python

Dictionary

```
d = {  
    'my_name': 'Sangkeun Kim',  
    'age': 30,  
    'emails': [{  
        'comment': 'private mail',  
        'email': 'sisobus@naver.com'  
    }, {  
        'comment': 'public mail',  
        'email': 'sisobus1@gmail.com'  
    }]  
}  
  
print(d['my_name'])           # Sangkeun Kim  
print(d['emails'][1]['comment']) # public mail
```


Python

If

```
a = 3.1415927584
if a > 3 and a < 4:
    print('a is in range({}, {})' .format(3, 4))
elif a <= 3 or a >= 4:
    print('a is not in range(%s, %s)' % (3, 4))
else:
    print('It is impossible')

# a is in range(3, 4)
```

Python

For (range)

```
for i in range(3):  
    print(i)
```

```
# 3
```

```
# 2
```

```
# 1
```

```
l = []
```

```
for i in range(3):  
    l.append(i)
```

```
# l: [0, 1, 2]
```

Python

For (enumerate)

```
l = [0, 1, 2]
for i in range(len(l)):
    print(l[i])
# 0\n1\n2\nl = [3, 4, 5]
for value, index in enumerate(l):
    print(value, index)
"""
0, 3
1, 4
2, 5
"""
```

Python

Function

```
def f(a, b, c):  
    a = a + 1  
    b.append(5)  
    c['test'] = 'T_T'  
_a = 5  
b = [4, 3]  
c = {}  
f(_a, b, c)  
print(_a, b, c) # 5 [4, 3, 5] {'test': 'T_T'}
```

Python

Class

```
class Wine:
    def __init__(self):
        self.is_alcohol = True
    def color(self):
        return 'Red or White or Sparkling'

class RedWine(Wine):
    def color(self):
        return 'Red'

wine = Wine()
print(wine.is_alcohol, wine.color())
# True Red or White or Sparkling
red_wine = RedWine()
print(red_wine.is_alcohol, red_wine.color()) # True Red
```

Python

File write

```
s = 'hey guys, please don\'t sleep. i\'d like to sleep too.'  
with open('Filename.txt', 'w') as fp:  
    fp.write(s)
```

Python

File read

```
with open('Filename.txt', 'r') as fp:  
    s = fp.read()  
print(s)  
# hey guys, please don't sleep. i'd like to sleep too.
```

Python 3.6

Installation (OS X)

1. <https://www.python.org/ftp/python/3.6.8/python-3.6.8-macosx10.6.pkg>
2. 실행은 \$ python3.6 <sourceFileName>
3. 어차피 우린 이렇게 실행 안할거임



pip is the package installer for Python

1. 파이썬으로 작성한 패키지 (라이브러리)를 설치, 관리하는 패키지 관리 시스템
2. 보통 PyPI(Python Package Index)로 연결되어 있음
3. `$ pip install <package_name>` 으로 설치할 수 있음
4. ~~Kakao mirror로 속도를 빠르게 할 수 있음~~



Installation (osx)

1. `$ sudo easy_install pip`

Virtualenv

is a tool to create isolated Python environments

1. Ian Bicking이 만들었음 (<http://www.ianbicking.org/>)
2. 독립적인 python package 관리 환경을 만들어줌
3. 프로젝트마다 virtualenv로 환경을 독립적으로 만들 수 있음
 - A 프로젝트는 flask 0.11을 쓰는데 B 프로젝트는 flask 1.0.2를 쓰는 경우
4. 여긴 이게 생긴 역사를 얘기하고 구렁이 담 넘어가듯 넘어가겠음

Virtualenv

Installation

1. `pip install virtualenv`

Virtualenv

Usage

1. `$ virtualenv venv`
2. `$. venv/bin/activate` (applying virtualenv)
3. `(venv)$` (applied virtualenv)

Flask

The Python micro framework for building web applications

1. 정말 배우기 쉽고 작은 웹 프레임워크
2. Python 2.x 에는 django와 라이벌
3. Python 3.x 에는 sanic, tornado, twisted, aiohttp와 라이벌
4. 사용자 수가 django와 함께 top 1,2위를 다툼

Flask

Installation

1. `$ pip install flask`

Flask

Usage (main.py)

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route('/')  
def hello():
```

```
    return 'Hello, World!'
```

```
if __name__ == '__main__':  
    app.run()
```


Flask

Usage

1. `$ virtualenv --python=python3.6 venv`
2. `$. venv/bin/activate`
3. `(venv)$ pip install flask`
4. `(venv)$ python main.py`

Flask

실습1

1. `$ cd WebStudio2019`
2. `$ git checkout -b 2-flask-1-〈여러분영어이름〉`
3. `$ mkdir 2_flask_1/practice/〈여러분영어이름〉`
4. `$ cd 2_flask_1/practice/〈여러분영어이름〉`
5. `$ virtualenv --python=python3.6 venv`
6. `$. venv/bin/activate`
7. `(venv) $ pip install flask`
8. `(venv) $ vi main.py`
(위에 위에 페이지 코드 복사 붙여넣기)
9. `(venv) $ python main.py`

Flask

실습1

11. (venv)\$ git add .
12. (venv)\$ git commit -m “Add flask simple script”
13. (venv) \$ git push origin 2-flask-1-〈여러분영어이름〉
14. Pull requests

Flask

URI에 변수 이용하기

```
from flask import Flask

@app.route('/variable/<input_string>')
def variable(input_string):
    return '<h1>{}</h1>'.format(input_string)

if __name__ == '__main__':
    app.run(debug=True)
```

Flask

방문횟수 기록하기

```
from flask import Flask
import os
```

```
@app.route('/counter')
def counter():
    filename = 'visit_count.txt'
    count = 0
    if not os.path.exists(filename):
        with open(filename, 'w') as fp:
            fp.write(str(count))
    with open(filename, 'r') as fp:
        count = int(fp.read())
    count += 1
    with open(filename, 'w') as fp:
        fp.write(str(count))
    return str(count)
```

```
if __name__ == '__main__':
    app.run(debug=True)
```

Flask

현재 시각 보여주기

```
...
from datetime import datetime
...

@app.route('/current_date')
def current_date():
    current = datetime.now()
    return current.strftime("%Y-%m-%d %H:%M:%S")

if __name__ == '__main__':
    app.run(debug=True)
```

Flask

URI과 URL

1. URI (Uniform Resource Identifier) vs URL (Uniform Resource Locator)
2. 현대의 웹은 Redirection을 통해 웹 주소와 서버의 파일 주소와 대부분 다르다.
 - 예를들어 api.sisobus.com/share/syllabus_web_studio.pdf 는 사실 share 디렉토리 밑에 존재하지 않는다.

Q & A

Appendix

Bash 기본 명령어

1. ls : 현재 위치의 파일 리스트
 - ls -al : 숨겨진 파일까지 전부 보기
2. cd : 디렉토리 이동
 - .은 현재 디렉토리, ..은 바로 전 디렉토리를 의미함
 - 예를들어 cd .. 이 명령어는 바로 전 디렉토리로 이동
3. rm : 파일 지우기
 - rm -rf : 디렉토리도 지울 수 있음
4. vi <filename> : vim 에디터를 이용한 파일 작성하기
5. mkdir <directoryname> : 디렉토리(폴더) 만들기
6. pwd : 현재 위치 출력하기

Appendix

Vim

1. 최고의 Text editor
2. 이것만 잘써도 코딩 생산성이 상당히 많이 올라감
3. 대부분의 IDE에는 Vim Plugin이 존재함
4. Vim 쓰세요
5. 두번쓰세요
6. 세번쓰세요
7. 평생쓰세요

Appendix

Vim

1. 명령어 모드와 에디터 모드로 나뉨
2. 맨처음 들어갈 때에는 명령어 모드임
3. 에디터 모드에서 명령어 모드로 바꾸는 것은 esc로 함
4. 명령어 모드에서 에디터 모드로 바꾸는 것은 i, o, a로 함 (i만 알아도 상관 없습니다.)

Appendix

Vim 명령어 모드

1. i는 현재 커서에서 에디터모드로 전환
2. yy는 현재 줄 복사
3. p는 복사한 것을 붙여넣기
4. dd는 현재줄 삭제
5. u는 undo
6. ctrl+r는 redo
7. :wq 는 저장후 종료를 의미함
8. /<search string> 으로 검색할 수 있음
9. :%s/<stringA>/<stringB>/g 는 stringA를 stringB로 대체함
 - :%s/^/₩/₩//g : 맨 앞에 //를 붙임
 - :3,12s/ha/hi/g : 3번줄부터 12번째줄의 ha를 hi로 바꿈