

파이썬 101

무한한 라이브러리, 저 너머로!

September 18, 2019

51일 전: 왜 파이썬?

WHY Python?

Python lets you

- ▶ care less about the details.
- ▶ think at the high level.

공감 되시나요?

오늘: 왜 파이썬?

WHY Python?

Python lets you

- ▶ care less about the details.
- ▶ think at the high level.

그리고 다양한 라이브러리들

파이선이 슈퍼히어로라면...

edufy.me

Python

Has a superpower for everything.

```
import lasers  
import x_ray_vision  
import flight  
import super strength  
import super hearing  
import time travel  
... and the list goes on.
```



근데 진짜임

```
import antigravity
```

유용한 모듈, 라이브러리와 프로그램들 일부

- ▶ pdb: 디버깅
- ▶ PyPy: 빠른 파이썬
- ▶ pip: 파이썬 패키지 관리자
- ▶ Numpy: 빠른 계산
- ▶ Pillow: 이미지 처리
- ▶ OpenCV: 컴퓨터 비전
- ▶ Request: HTTP 요청 주고받기
- ▶ Scrapy: 웹크롤링
- ▶ Selenium: 웹브라우저 자동화
- ▶ KoNLPY: 한국어 처리
- ▶ tossi: 한국어 조사처리
- ▶ Ren'Py: 비주얼 노벨 파연사 만들기
- ▶ Pygame: 2D 게임 만들기

pdb

The Python Debugger

지금까지의 디버깅

```
def wrong_function(n):  
    if n == 2 and n == 3:  
        print("n is: ", n)  
        return True  
    if n % 2 == 0 or n < 2:  
        print("n is now: ", n)  
        return False  
    for i in range(3, int(n**0.5+1), 2):  
        print(n, i)  
        if n % i == 0:  
            return False  
        print("should not print if n is prime", i)  
    if n == 3:  
        print("if I can read this, I am XXXXed")  
    return True
```


좋은 디버깅

```
~/wor../kim code > pdb gbach.py
> /home/indiofish/workspace/kim_code/gbach.py(1)<module>()
-> n = int(input(""))
(Pdb) list
1  -> n = int(input(""))
2
3      import math
4
5      def is_prime(n):
6          if n == 2 or n == 3:
7              return True
8          if n % 2 == 0 or n < 2:
9              return False
10         for i in range(3,int(n**0.5+1),2):
11             if n % i == 0:
(Pdb) next
10
> /home/indiofish/workspace/kim_code/gbach.py(3)<module>()
-> import math
(Pdb) next
> /home/indiofish/workspace/kim_code/gbach.py(5)<module>()
-> def is_prime(n):
(Pdb) print(n)
10
(Pdb) step
> /home/indiofish/workspace/kim_code/gbach.py(16)<module>()
-> for k in range(1, n//2+1):
(Pdb) step
> /home/indiofish/workspace/kim_code/gbach.py(24)<module>()
-> if is_prime(k) and is_prime(n-k):
(Pdb) print(k)
1
(Pdb) !n=30
(Pdb) print(n)
30
(Pdb) is_prime(100)
False
```

좋은 디버깅

- ▶ `help`
가능한 명령어를 보여준다.
- ▶ `break` (줄번호|함수이름)
n번째 줄이나 함수에 도달하면 일시정지
- ▶ `print()`
우리가 아는 출력함수
- ▶ `!x = 3`
값 바꾸기
- ▶ `next`
현재 함수의 다음줄로 넘어간다.(중간의 함수는 그냥 실행)
- ▶ `step`
`step into`: 함수 안으로 들어간다.

좋은 디버깅

장점들

- ▶ 프로그램을 켜다켰다하지 않고도 테스트 가능
- ▶ 프로그램을 바꾸지 않고도 테스트 가능
- ▶ 멋있음

PyPy



PyPy

Python으로 실행하는 Python

```
python program.py == pypy program.py
```

JIT 컴파일과 파이썬 코드를 C로 번역하는 과정 등을 거쳐 표준 python 구현체보다 2-10배 정도 빠를 때도 있고, 조금 느릴 때도 있습니다.

pip install you-name-it

파이썬에 기본적으로 포함되어 있는, 패키지 관리자. 일일이 홈페이지에 들어가서 설치파일을 받지 않아도 되는 장점이 있습니다.

Numpy

pip install numpy

Anaconda 등에 포함되어 있어 써보셨을 것 같은 Numpy

왜 Numpy?

딥러닝 등의 근간이 되는 행렬¹을 빠르게 계산 해줍니다. 물론 그외에도 많은 수학기능이 있습니다.

¹김모 교수님 "미적분학은 한물갔고 이젠 선형대수학의 시대입니다 음하하!"

Numpy

```
import numpy as np
SIZE = 500
a = np.array([[randint(-100,100) for _ in range(SIZE)] for _ in
               range(SIZE)])
b = np.array([[randint(-100,100) for _ in range(SIZE)] for _ in
               range(SIZE)])
c = np.dot(a, b)
```

500 x 500 행렬의 곱셈

Numpy vs 저

```
singlecore took: 19.164047956466675  
multicore took: 6.122005462646484  
numpy took: 0.12259554862976074
```

코드는 깃 저장소의 `mat_mul.py`에서 확인 가능합니다.

Pillow

pip install Pillow

Python Image Library (PIL)의 변형판
사진을 늘리고, 줄이고, 변형하는데 유용합니다.
<https://pillow.readthedocs.io/en/stable/>

Pillow

```
from PIL import Image, ImageFilter
import numpy as np

im = Image.open("homer.png")
im.show()
im = im.convert('RGB')
out = im.filter(ImageFilter.FIND_EDGES)
out.show("edges only")
out = im.transpose(Image.FLIP_LEFT_RIGHT)
out.show("homer in the mirror")
out = im.convert('LA')
out.show("noire homer")

# images are actually arrays!
pix = np.array(im)
print(pix)
```