

PYTHON

0.설치

1)python

www.python.org

python은 2.x 와 3.x 버전을 함께 제공한다.

- 자료형(long type이 int type에 포함, str에 Unicode 포함 등) 및 문법 일부가 변경됨
- 2.x 에서 만들어진 패키지들이 많았음 (3.x로 많이 변경되었다.)
- 3.x 버전 다운로드

0.설치

1)python

- 1) add python to path 체크
- 2) customize installation -> next -> install location을
c:\python으로 변경 후 install
- 3) cmd에서 python 입력하면 실행되는지 확인

0.설치

2)pyDev

- 1) help - marketplace - PyDev 설치
- 2) window - preferences - pydev
- interpreters - python interpreter – Browse for python
- 3) C:\python\python.exe 경로 잡아주고, 이름은 편한걸로

0.설치

2)pyDev

4) PyDev Project

5) interpreter를 3번에서 정해준 이름으로 변경(default)

6) Additional syntax validation에서 설치 버전 선택

7) 프로젝트 우클릭 - new - pydev module

1.python

python이란

Guido van Rossum (네덜란드) 이 심심해서 만든 언어.

특징

- 쉽다 : `print('hello world')`
- 들여쓰기가 필수이다 : `tab` or `whitespace*4`
- 인터프리터이다.
- `co-routine`을 지원한다
(`sub-routine` : `main routine`에 종속, `co-routine` : `main routine`과 대등)

1.python

python library

c:\python\Lib\json

- 해당 폴더를 보면 *.jar 안에 있는 class 처럼
decoder.py 등이 있는 것을 볼 수 있다. (library / module 이라고 한다.)
- beautiful soup, numpy, scipy, tensorflow 등의 library가 있으며
웹 크롤링, 공학계산, 머신러닝 등의 많은 분야에서 사용된다.

1.python

실행

IDLE 찾아서 실행 (cmd에서 python 입력한 결과와 같음)

```
>>100
```

```
100
```

```
>>'a'
```

```
'a'
```

```
>>"a"
```

```
'a'
```


2.type

1)숫자형

정수형(int)

>>a = 100

실수형(float)

>>b = 100.0

* 8진수 (0o) / 16진수 (0x)

2.type

2)문자형

single quotation*1

```
> > > print('abc')
```

single quotation*3

```
> > > print("""abc  
def""")
```

*여러줄 주석으로도 사용

double quotation*1

```
> > > print("def")
```

double quotation*3

```
> > > print("""def  
abc""")
```

2.type

3)list

변수명 = [값 (index 0), 값 (index 1), 값 (index 2), ...]

```
a=list()
```

```
a.append(1)
```

```
a[0]='a'
```

```
a=[1,2,3,[4,5,6]]
```

```
a[3][1] = 7
```

2.type

4)tuple

변수명 = (값, 값, 값, ...)

- 값을 변경할 수 없는 리스트
- 값을 변경할 수 없다는 것을 빼면 list와 같다

2.type

5)set

변수명 = set(값)

- 중복 X, 순서 X

set01=set([1,2,3])

set02=set('hello')

- 합집합 : a.union(b), a | b

- 교집합 : a.intersection(b), a & b

2.type

6)dictionary

변수={'key':'value', 'key':'value', 'key':'value', ... }

- key를 통해 value를 가져올 수 있다.
- key는 중복 X, value는 중복 O

2.type

7)연산자

산술연산

$a + b$

$a - b$

$a * b$

a / b : 나누기 (몫.나머지)

a / b : 나누기 (몫)

$a \% b$: 나누기 (나머지)

*증감연산자(++a, b--) 는 없다.

2.type

7)연산자

비교연산

`a == b`

`a != b`

`a > b`

`a >= b`

`a < b`

`a <= b`

`a is b`

`a is not b`

`True and False`

`True or False`

`not True`

2.type

7)연산자

범위연산

- list[index] = 해당 index의 값
- list[start index : end index] = start index ~ end index-1까지
- list[start index : end index : step] = step만큼씩 건너서

2.type

7)연산자

멤버연산

- a in list/dict...
- b not in list/dict...

2.type

8)내장함수

자주 사용하는 내장함수

int() #정수

float() #실수

complex() #복소수

str() #문자열

repr() #'문자열'객체

chr() #문자

ord() #아스키코드

oct() #16진수

hex() #8진수

eval() #'숫자인 문자열'을 숫자로

tuple() #대상을 tuple로

list() #대상을 list로

set() #대상을 set으로

dict() #대상을 dict로

frozenset() #대상을 고정, 중복 제거

zip() #자료형을 묶어준다

3.제어문

1)조건문

if ~ elif ~ else

if 조건 :

 명령문

elif 조건 :

 명령문

else

 명령문

3.제어문

2)반복문

while ~ else

while 조건 :

 명령문1

else

 명령문2

- 조건이 true이면 명령문 1을 반복수행 후 else의 명령문 2 수행
false이면 명령문 2 수행

3.제어문

2)반복문

for ~ else

for 변수 in 자료형(순서) :

 명령문1

else :

 명령문2

*명령문 2는 반복이 완료된 후 수행된다.

3.제어문

2)반복문

`range(start, stop, step)`

start부터 stop전까지 step만큼 건너뛰며 sequence 생성

```
for i in range(1,100):  
    print(i)
```

4.함수

1)함수 선언

함수 선언 방법

```
def 함수이름(파라미터)  
    명령문  
    return
```

* if __name__=="__main__": #프로그램의 주 진입점(main함수)

4. 함수

1) 함수 선언

함수 선언 방법

```
def 함수이름(파라미터)  
    명령문  
    return
```

* `if __name__=="__main__":` #프로그램의 주 진입점(main함수)

4. 함수

2) 익명함수

lambda 함수

lambda 파라미터, 파라미터,... : 표현식

5.모듈

numPy : 수학, 과학 연산을 위한 library

- cmd -> pip install numpy
- numpy 자체로만 사용기도 하지만,
보통 scipy 등 다른 library와 함께 사용된다.

5.모듈

matplotlib : 차트, 플롯 등의 시각화 library

- cmd -> pip install matplotlib
- 다른 수치해석 모듈에서 연산된 결과를 시각화하는데 사용한다.

5. 모듈

beautiful soup : web parsing library

- cmd -> pip install beautifulsoup4
- web page의 html을 parsing 해 와서, 보통 webcrawling 할 때 사용

6.io

```
변수 = open('파일명', '옵션')
```

```
변수.write('내용')
```

```
변수.close()
```

```
with open('파일명', '옵션') as 변수:
```

```
    코드
```

```
...
```

6.io

<옵션>

r : 읽기

w : 쓰기 (기존 내용 덮어쓰기)

a : 파일 끝에 쓰기

x : 파일 없을 때 파일 만들어서 쓰기

t / b : text / binary (default : t)

0.챕터

소제목

내용