1. **환경 설정**

1.1. python 설치

- python.org 접속

- 상단의 Downloads 메뉴에 마우스 커서 놓기

- 3.0 이상의 python 설치 파일 다운 (18.03.06 기준 Windows latest release version : Python 3.6.4)

- 설치 파일 실행

- 설치 시 환경 변수 체크

1.2. sublimetext 3 설치

- https://www.sublimetext.com/3 접속 > sublimetext3 download

1.3. SublimeText 단축키

- sublime 전체 주석 단축키 : Ctrl + /

- 이전 명령 재실행 > Command Palette : Ctrl + Shift + P

|  |
| --- |
| print("Hello World!") #test code |

**2. Python 기본**

2.1. Python

- 컴파일 없이 실행 가능한 스크립트 언어

- 인터프리터 사용

# 컴파일러와 인터프리터

|  |  |
| --- | --- |
| **컴파일러** | **인터프리터** |
| 프로그램 단위 번역 | 명령 줄 단위 번역 |
| 번역 속도 느림 | 번역 속도 빠름 |
| 실행 속도 빠름 | 실행 속도 느림 |
| 큰 메모리 필요 | 적은 메모리 필요 |

2.2. Python 기초문법

2.2.1 자료형

- 데이터를 변수에 담는 순간 자료형이 정해진다.

- Python에서는 자료형마다 연산을 지원한다.

# 출력

>>> a=1

>>> a

1

>>> print(a)

1

>>> print(a,"test")

1test

# 연산자

정수와 실수의 연산 => 실수

음수를 나누었을 때 나머지가 있는 경우

ex) -11 // 2 => -6 (-1이 추가된 값)

# 문자열 선언 방법

> a="asdf" // "" 포함

> a='asdf' // '' 포함

> a='''asdf'''

> a="""asdf"""

요즘 기능이 좋아졌으니 그냥 """~~~"""로 묶어도 충분히 문자열이 내가 원하는대로 따옴표도 잘 표시되고 개행도 그냥 하면 되고 탭도 잘먹혀서 저장됨

(탭: Ctrl + TAB)

# 인덱싱

> 문자열에서 특정 문자를 가리키는 것을 의미

> 인덱스값은 0부터 시작

> 배열처럼 생각

ex)

>>> a="2018-03-13"

>>> print(a[0])

2

# 슬라이싱

> 문자열에서 특정 문자열을 자르는 것

ex)

>>> a="2018-03-13"

>>> print(a[2:7])

18-03

>>> print(a[:7])

2018-03

>>> print(a[5:])

03-13

=================================================================================================================================

# 리스트

> 배열과 비슷한 개념.

>>> a=[0, 123, "aaa",111, 1.1]

>>> a

[0, 123, 'aaa', 111, 1.1]

>>> print(a[2:])

['aaa', 111, 1.1]

>>> print(a[1]," and",a[3:5])

123 and [111, 1.1]

# 딕셔너리

> 대응 관계를 나타낼 수 있는 자료형

> key와 value가 연결된 자료형

> 사전과 비슷함

> 딕셔너리 자체의 연산은 없음

> 인덱싱 O, 슬라이싱 X

ex)

>>> dic={'1':'aaaa','2':'bbbb','3':'cccc'}

>>> dic

{'1': 'aaaa', '2': 'bbbb', '3': 'cccc'}

>>> dic['1']

'aaaa'

=================================================================================================================================

3. Python 함수

# 리스트 함수

@ count(x)

리스트 내에 x의 개수를 반환

@ index

찾고자 하는 내용이 없으면 error 출력

@ find

찾고자 하는 내용이 없으면 -1 출력

@ join

리스트를 문자열로 반환

ex)

>>> a='.'

>>> r=a.join("DSM")

>>> print(r)

D.S.M

# 문자열 함수

@ upper

> 소문자를 대문자로 변환

@ lower

> 대문자를 소문자로 변환

@ replace

> 문자열 바꾸기

ex)

>>> a="aabbcc"

>>> a.replace("bb","dd")

'aaddcc'

@ split

> 문자열 나누기

ex)

>>> a = "Life is too short"

>>> a.split()

['Life', 'is', 'too', 'short']

>>> a = "a:b:c:d"

>>> a.split(':')

['a', 'b', 'c', 'd']

@ strip

> 양쪽 공백 지우기

ex)

>>> a=" asdf "

>>> a.strip

'asdf'

@ lstrip

> 왼쪽 공백 지우기

@ rstrip

> 오른쪽 공백 지우기

# 내장 함수

@ type

> 자료형 반환

ex)

>>> a="DSM"

>>> type(a)

<class 'str'>

@ int

> 문자열 형태의 소수점이 있는 숫자 등을 정수 형태로 반환

> 정수를 입력으로 넣으면 그대로 반환

ex)

>>> int(3)

3

>>> int(3.141592653589793238462643383279)

3

@ str

> 문자열 형태로 객체를 변환하여 반환

ex)

>>> str(555)

'555'

@ ord

> 문자의 아스키 코드값 반환

@ chr

> 아스키 코드값에 해당하는 문자 반환