파이썬 다루기!

1.자료형 : 파이썬에서 다루는 데이터가 어떤 형태가 있는지



네모 하나가 셀(Cell)이라고 부름

초록색 : 셀 안에 뭔가를 입력할수 있다

파란색 : 셀의 상태를 제어할 수 있는 상태 (입력이 아닌 설정을 변경할수 있는 단계) 여기서 m을 입력하면 In[]: 이 사라짐 -> 마크다운을 입력할 수 있음

마크다운 : 메모를 작성할 수 있는 형식 (C++에서 ## 이거랑 비슷한듯?)

M을 누르면 마크다운, Y를 누르면 마크다운 해제 가능

Shift + Enter : 셀을 실행함

마크다운에서 #은 제목을 의미, 한칸을 띄워야 작동하고 # 4개까지 붙여서 소제목 단계로 내려갈수 있음

파란색 상태일 때 A(Above)를 누르면 위에 셀이 생기고 B(Bottom)을 누르면 아래에 셀이 생김

지우고 싶으면 X 지운거 실행취소는 Z

텍스트이(가) 표시된 사진

자동 생성된 설명

a, b, c라는 변수에 숫자를 정의

print(a/b/c) : 각 a,b,c를 출력

print(type(a/b/c)) : 각 a,b,c가 어떤 종류의 데이터인지 출력

class ‘int’ (integer) 이것은 정수 타입의 숫자 데이터다 라는 의미 3/21 수업

텍스트이(가) 표시된 사진

자동 생성된 설명

숫자는 에디터 내에서 코드 실행 순서임 (별로 큰 의미는 읎다 어짜피 커널 재실행하면 위에서부터 다시 하는듯?)

함수 : 명령어 세트

print 함수 : 데이터 출력 함수

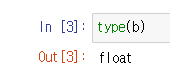
type 함수 : 데이터의 형태, 종류를 나타내주는 함수

int (integer) : 정수형 데이터를 의미

텍스트이(가) 표시된 사진

자동 생성된 설명

float : 실수형 데이터를 의미



print 안쳐도 일단은 출력되는데… 주피터의 편의기능임 실상 차이는 없음

텍스트이(가) 표시된 사진

자동 생성된 설명

주석 : 코드에 #을 붙여서 코드로 취급되지 않는 부연설명 기능

사칙연산 : 사칙연산은 다 알고…

a \*\* b : a^b를 나타내는 제곱

a // b : a를 b로 나눴을때의 몫

a % b : a를 b로 나눈 나머지 (엑셀 mod함수)

텍스트이(가) 표시된 사진

자동 생성된 설명

1과 1.0은 수학적으로는 같지만… 프로그래밍 적으로는 int와 float이기 때문에 다르다

텍스트, 테이블이(가) 표시된 사진

자동 생성된 설명

문자열(string, str) 은 텍스트 형태의 데이터 (그것이 숫자라도)

따옴표(“) 내에 집어넣어서 문자열임을 나타냄

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

문자열에 따옴표 자체를 포함해야 한다면 큰따옴표와 작은따옴표를 섞어서 사용 둘다 쓰고싶으면 따옴표 3개짜리 쓰면 됨

텍스트이(가) 표시된 사진

자동 생성된 설명

역슬래시(폰트상 \)을 앞에써서 텍스트 취급도 가능함 : 정확히는 특수한 의미를 가진 문자가 아닌 문자 그 자체로서 의미를 가지도록 설정함

텍스트이(가) 표시된 사진

자동 생성된 설명

여러줄 문자열은 \n 으로 줄바꿈 가능

텍스트이(가) 표시된 사진

자동 생성된 설명

미리 정의된 이스케이프 코드가 있는데 캐리지 리턴이랑 폼피드는 쓸일 없으니 그런게 있다 정도만 알면 된다

텍스트이(가) 표시된 사진

자동 생성된 설명

작은/큰 따옴표 3개짜리 쓰면 그냥 엔터로도 줄바꿈 삽가능

텍스트이(가) 표시된 사진

자동 생성된 설명

더하기 연산자로 문자열끼리 붙여주거나 숫자로 곱하면 반복이 가능함

텍스트이(가) 표시된 사진

자동 생성된 설명

인덱싱(대괄호) : 문자열의 특정한 자리의 문자(인덱스)를 데려옴, 파이썬은 문자 순서를 0부터 셈



마이너스 인덱스 : 오른쪽에서부터 문자를 데려옴, 근데 이건 0부터가 아니네 시1발



0: 이면 0 생략해도 알아서 인식함 (마이너스 인덱스도 가능)



개수상 배정은 안되어있지만 끝 문자열 넣고 싶으면 글자열 수 이상의 숫자 넣으면 됨



생략해도 되니 끝까지라면 생략을 하자



슬라이싱(대괄호내부 :) : 어디서부터 어디까지 텍스트를 가져옴

a:b 로 슬라이싱하면 b에 해당하는 인덱스는 안뽑힘(b 미만 개념으로 이해하면 됨)

텍스트이(가) 표시된 사진

자동 생성된 설명

마이너스 인덱스 타입으로도 가능함 근데 뒤에 :를 하나 더 붙여서 연산 조건을 붙여야 함 기본은 (:1)임 : (:-1)이면 -1을 계속 더해줌(좌측으로 1칸씩) (:2)이면 2를 계속 더해줌 (우측으로 2칸씩) 조건 자체가 안맞으면 문자열 출력이 안됨 (-1에서 1을 더해서 -4를 만들라던가?) 연산조건은 정수만 가능

도표이(가) 표시된 사진

자동 생성된 설명

마이너스 인덱스도 생략할건 생략가능

달력이(가) 표시된 사진

자동 생성된 설명

문자열 포맷 코드 : 문자열 내에 %d (digit)를 넣고 이후 %()에서 해당 %d의 값을 지정해줄 수 있다

텍스트이(가) 표시된 사진

자동 생성된 설명

%d는 정수?

텍스트이(가) 표시된 사진

자동 생성된 설명

소수점 짤라서 나타내기

0.4 = 소수점 4째 자리까지 표현 (반올림)

f = Float 타입

%pi : 해당 값에 pi를 대입

테이블이(가) 표시된 사진

자동 생성된 설명

포맷 함수 : .format() 형태로 format 내에서 앞부터 0 순으로 순서를 정하고 문자열 내에서는 순서 상관없이 활용 가능, 재사용도 가능함

테이블이(가) 표시된 사진

자동 생성된 설명

포맷 함수 내에서 변수 정의 및 변수를 중괄호 내에 입력해 바로 꼬라박는것도 가능함

텍스트이(가) 표시된 사진

자동 생성된 설명

중괄호 내부

0 : 포맷함수내 첫번째 숫사

:0.4 : 소수점 4째자리까지 반올림

f : float 타입

f-string 함수 (3.6버전부터 가능)

텍스트이(가) 표시된 사진

자동 생성된 설명

f를 문자열 시작 전에 붙이면 중괄호 내에 변수를 집어넣을 수 있음. 중괄호 내에서 연산도 가능함

텍스트이(가) 표시된 사진

자동 생성된 설명

소수점 표현도 ㅈㄴ 똑같네

텍스트이(가) 표시된 사진

자동 생성된 설명

.count 함수 : 특정 문자 내에 어떤 문자가 몇번 들어갔는지 세기

텍스트이(가) 표시된 사진

자동 생성된 설명

.find 함수 : 특정 문자 내에 어떤 문자가 처음으로 몇번째 위치에서 나왔는가? (이것도 0부터 셈)

없으면 -1 반환

텍스트이(가) 표시된 사진

자동 생성된 설명

.index 함수 : find 함수와 같으나 없으면 에러 반환

차트이(가) 표시된 사진

자동 생성된 설명

.join 함수 : 함수 앞에 입력한 텍스트를 각 문자 사이에 삽입해줌

차트이(가) 표시된 사진

자동 생성된 설명

리스트(나중에 배움)에도 활용 가능

3/23 강의

텍스트이(가) 표시된 사진

자동 생성된 설명

대소문자 변환 (upper, lower) : 컴퓨터는 대문자와 소문자를 완전히 다른 것으로 인지함



공백 제거 (strip) : 문자열 양쪽 끝에 남아있는 공백을 제거함 (trim) 공백도 문자 데이터의 하나임

텍스트이(가) 표시된 사진

자동 생성된 설명

문자열 치환 (replace) : 특정 문자를 다른 문자로 변환 (감지되는 전부 다), 공백으로 바꾸면 사실상 삭제가 가능

텍스트이(가) 표시된 사진

자동 생성된 설명

문자열 나누기 (split) : 문자열을 나눌수 있음 함수 내부를 공백으로 두면 띄어쓰기 기준으로 나누고 특정 문자를 넣어서 특정 문자 기준으로 나눌수도 있음

불 자료형 : 참과 거짓을 나타내는 자료형

텍스트이(가) 표시된 사진

자동 생성된 설명

True, False는 첫글자 대문자 지킬 것

텍스트이(가) 표시된 사진

자동 생성된 설명

bool 타입

텍스트이(가) 표시된 사진

자동 생성된 설명

== : 같은 데이터인지 판단

부등호로 대소 구분 가능

Bool() : 데이터가 값이 있으면 (텍스트, 숫자 구분 없이) True, 값이 없거나 0이면 False

텍스트이(가) 표시된 사진

자동 생성된 설명

Int 함수 : 정수형으로 변환. True는 1, False는 0으로 변환됨

텍스트이(가) 표시된 사진

자동 생성된 설명

컬렉션

도표이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

마크다운에서 | (\ + shift) 로 표 생성 가능

첫번째 줄은 컬럼 헤더 설정

:-- 왼쪽정렬

--: 오른쪽정렬

:--: 양쪽정렬

리스트 (list) : 데이터 세트 함수중 가장 많이 씀

튜플 (tuple) : 데이터 처리에 리소스가 적게 드므로 데이터 전달이나 수정되면 안될 때 사용

세트 (set) : 중복값을 넣어도 중복값들이 제거됨 데이터 정리 가능

딕셔너리 (dict) : 데이터의 1:1 관리에 사용 키-값이 다른 키-값과 중복되지 않음

텍스트이(가) 표시된 사진

자동 생성된 설명

리스트

텍스트이(가) 표시된 사진

자동 생성된 설명

리스트도 데이터 단위로 인덱싱 슬라이싱이 가능함

텍스트이(가) 표시된 사진

자동 생성된 설명

다중리스트 : 리스트 자체도 리스트 데이터 단위가 될 수 있음 이 리스트도 단독으로 뽑으면 리스트에 대한 연산 가능

텍스트이(가) 표시된 사진

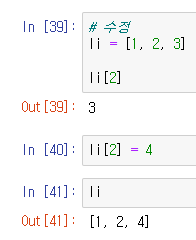
자동 생성된 설명

문자열처럼 리스트는 리스트 병합과 반복 연산이 가능함

텍스트이(가) 표시된 사진

자동 생성된 설명

리스트 내부 특정 데이터를 삭제 가능 (인덱스 값으로)



수정도 인덱스값 기반으로 가능

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

Remove 함수는 리스트에서 첫 번째로 나오는 특정 요소를 삭제하는 함수

텍스트이(가) 표시된 사진

자동 생성된 설명

Pop 함수 : 리스트 내 마지막 요소를 지우고 난 후 해당 값을 반환함

텍스트이(가) 표시된 사진

자동 생성된 설명

인덱스 값을 이용하여 특정 위치를 반환 및 삭제하는 것도 가능하다

텍스트이(가) 표시된 사진

자동 생성된 설명

Append : 리스트의 마지막에 요소를 추가함 (리스트 자체도 요소로서 들어감)

Insert(a, b) : 인덱스 값 기반으로 a 번째 위치에 4가 들어오도록 삽입

텍스트이(가) 표시된 사진

자동 생성된 설명

Extend(리스트) : 리스트를 뒤에다가 합침 (+랑 거의 같음. +가 함수 대상이 바뀌는건 아님… extend = 리스트 = 리스트+리스트2)

텍스트이(가) 표시된 사진

자동 생성된 설명

리스트 오름차순/내림차순 정렬 sort 함수

같은 타입의 대소 or 순서를 구분 가능한 요소끼리만 가능함

텍스트이(가) 표시된 사진

자동 생성된 설명

reverse 함수 : 리스트의 순서를 거꾸로 뒤집음

차트이(가) 표시된 사진

자동 생성된 설명

슬라이싱으로도 뒤집을수 있으나 함수 대상이 실제로 바뀌지는 않음

텍스트이(가) 표시된 사진

자동 생성된 설명

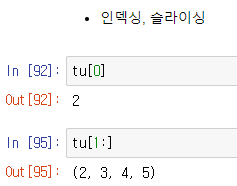
문자열에 사용되는 index, count 함수도 그대로 사용 가능

튜플

텍스트이(가) 표시된 사진

자동 생성된 설명

튜플은 데이터의 수정이 불가능함



인덱싱, 슬라이싱은 튜플을 조작하는 것이 아닌 데이터를 추출하는 것이므로 적용 가능함

텍스트이(가) 표시된 사진

자동 생성된 설명

튜플을 건드는 것이 아닌 연산은 실행 가능함 (append 같은건 안됨)

딕셔너리

텍스트이(가) 표시된 사진

자동 생성된 설명

딕셔너리는 각 값에 이름표를 붙인 것

리스트에서 인덱스 값도 수동으로 지정을 한것으로 이해하면 편함 (name = 인덱스 0)

Key 값은 immutable만 가능! (문자열, 튜플 등)

테이블이(가) 표시된 사진

자동 생성된 설명

리스트같은것도 들어감

텍스트이(가) 표시된 사진

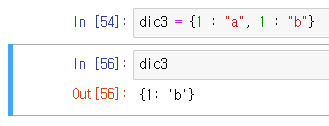
자동 생성된 설명

딕셔너리에 주소와 값을 지정하여 추가 가능

텍스트이(가) 표시된 사진

자동 생성된 설명

삭제를 할때도 주소만 써주면 값도 세트로 삭제됨



딕셔너리는 동일한 key (주소)가 존재할 수 없음 동일한 주소를 세팅하면 나중에 입력한것으로 덮어씌워짐

텍스트이(가) 표시된 사진

자동 생성된 설명

Keys 함수 : 딕셔너리의 key 리스트를 모두 출력

Values 함수 : 딕셔너리의 value 리스트를 모두 출력

Items 함수 : 딕셔너리의 key – value 매칭을 튜플 형식으로 출력 딕셔너리가 딕셔너리 형태로 출력되면 안될 때 사용함

텍스트이(가) 표시된 사진

자동 생성된 설명

Get 함수 : 지정한 키에서 값을 찾아 오되 해당 키가 없으면 지정한 값을 반환 (엑셀 IFERROR 같은거)

텍스트이(가) 표시된 사진

자동 생성된 설명

In 함수 : 딕셔너리에 키 값이 있는지 여부를 True / False로 반환함

텍스트이(가) 표시된 사진

자동 생성된 설명

Clear 함수 : 딕셔너리 내의 키-값을 몽땅 지움

3/28 강의

텍스트이(가) 표시된 사진

자동 생성된 설명

세트는 순서가 없고 중복을 허용하지 않음

집합 개념을 사용할수 있음

테이블이(가) 표시된 사진

자동 생성된 설명

세트의 교집합 & or intersection

텍스트, 테이블이(가) 표시된 사진

자동 생성된 설명

세트의 합집합 | or union

테이블이(가) 표시된 사진

자동 생성된 설명

세트의 차집합 – or difference

텍스트이(가) 표시된 사진

자동 생성된 설명

세트에서 값을 추가하거나 제거 가능

세트는 리스트에서 중복값을 제거할 때 많이 쓰임 (리스트 -> 세트 -> 리스트 변환)

텍스트이(가) 표시된 사진

자동 생성된 설명

Mutable과 immutable

텍스트이(가) 표시된 사진

자동 생성된 설명

Mutable은 변형이 일어나도 메모리 주소가 변하지 않는다 (변경이 가능하므로 해당 메모리 내에서 변형이 일어남)

텍스트이(가) 표시된 사진

자동 생성된 설명

Immutable은 값이 변하지 않음. 그러므로 해당 변수를 바꾸었을 경우 새로운 id로 배정되어서 메모리에 저장됨

텍스트이(가) 표시된 사진

자동 생성된 설명

파이썬은 메모리를 효율적으로 사용하기 위해서 같은 데이터를 지정할 경우 다른 변수라도 id가 같아짐 (별도로 똑같이 설정하면 id가 달라질 수 있음)

텍스트이(가) 표시된 사진

자동 생성된 설명

같은 id에 배정된 mutable들은 하나만 변경이 일어나면 다른 mutable도 모두 변경됨

텍스트이(가) 표시된 사진

자동 생성된 설명

리스트의 경우 슬라이싱을 처음부터 끝까지 하는 것으로 다른 id에 리스트를 복사시킬수 있다

텍스트이(가) 표시된 사진

자동 생성된 설명

Id가 다르므로 한쪽의 리스트가 변형되어도 다른 리스트는 변형되지 않는다

텍스트이(가) 표시된 사진

자동 생성된 설명

Copy 함수로도 복사 가능

도표이(가) 표시된 사진

자동 생성된 설명

마크다운에는 이미지 첨부 가능

절대 경로 : ?드라이브 포함한 모든 경로를 다 입력

상대 경로 : 기본 작업 폴더까지 들어온 상태에서 나머지 주소 입력 ex.) “./img/control.png”

텍스트이(가) 표시된 사진

자동 생성된 설명

조건문

If (조건코드):

(들여쓰기 = Tab) 코드1

Else:

(들여쓰기 = Tab) 코드2

조건문의 True, False 여부에 따라 실행하는 코드가 달라짐 (코드1 : True, 코드2 : False)

If / else 뒤 들여쓰기에 있는 코드들 = 코드블럭, 조건문에서 실행할 코드 단위를 설정하는 것으로 매우 중요함, 들여쓰기가 없는 줄로 넘어가게 되면 if / else의 코드블럭이 끝난 것으로 간주

콜론(:) 뒤의 Tab이 코드블럭을 정의함

텍스트, 테이블이(가) 표시된 사진

자동 생성된 설명

조건식 : 비교연산자… !=은 다를경우 True 반환

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트, 편지이(가) 표시된 사진

자동 생성된 설명

And, or, not

텍스트이(가) 표시된 사진

자동 생성된 설명

or 사용

텍스트이(가) 표시된 사진

자동 생성된 설명

In, not in 조건문

텍스트이(가) 표시된 사진

자동 생성된 설명

사용 예시

텍스트이(가) 표시된 사진

자동 생성된 설명

Elif를 이용하면 다중 if 사용 가능함

텍스트이(가) 표시된 사진

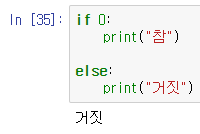
자동 생성된 설명

Elif와 다중 if 사용

텍스트이(가) 표시된 사진

자동 생성된 설명

None : 값이 없음을 나타내는 데이터 형태 (거짓 취급)

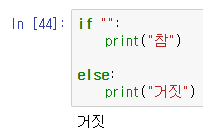


0도 거짓 취급 (False의 int형, 1은 참)

텍스트이(가) 표시된 사진

자동 생성된 설명

데이터가 있는 값도 참 (문자열을 넣어도 데이터가 있는 것이므로 참)



빈 문자열은 거짓

텍스트이(가) 표시된 사진

자동 생성된 설명

조건부 표현식으로 if를 줄일수도 있음

이 식은 단일 값에 대한 것이므로 조건에 따라 지정하는 변수가 다를 경우에는 조건부 표현식을 사용할 수 없음 (참일 경우 a 변수에 부여, 거짓일 경우 b 변수에 부여 등) 일반 if문을 사용해야됨

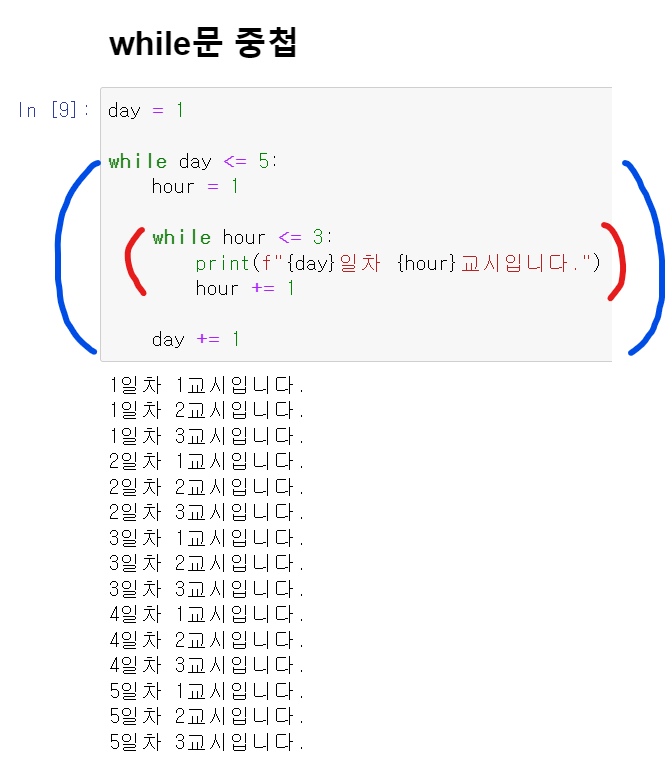
3/30 강의

텍스트이(가) 표시된 사진

자동 생성된 설명

While문 : 조건을 만족하는동안 특정 코드블럭을 반복하여 실행, 조건을 만족해야 실행하므로 경우에 따라 코드블럭이 한번도 실행되지 않을수도 있음. 무한루프도 가능하니 주의

+=는 특정 변수에 덧셈을 하는 식



While 문 중첩 : 빨간 코드블럭을 while 문 안에서 돌리고 나서 파란 코드블럭을 마저 실행하여 while문 반복



Break문 : 반복문을 강제로 종료시킬 수 있음 break가 실행되는 순간 즉시 코드블럭이 종료되므로 코드가 남았어도 실행하지 않음 (n = 10)

텍스트이(가) 표시된 사진

자동 생성된 설명

Continue 문 : 실행 시 반복문의 시작 지점(조건 체크)로 즉시 이동, 남은 코드는 실행되지 않으므로 원하는 코드를 생략하는 식으로 사용 가능 (a = 10)

중첩 while문에서 Break나 continue문이 실행되면 가장 가까운 while 문에 대하여 해당 코드가 실행됨 (전체에 대해서 break를 걸고싶다면 break를 2번 이상 사용하면 됨)

텍스트이(가) 표시된 사진

자동 생성된 설명

For 문 : 반복 가능 개체에서 요소를 하나씩 가져와서 코드 실행, 반복 가능 개체가 반복 회수를 결정, for에서 지정되는 변수가 개체에서 요소를 값으로서 할당받는다

텍스트이(가) 표시된 사진

자동 생성된 설명

For 문 사용

테이블이(가) 표시된 사진

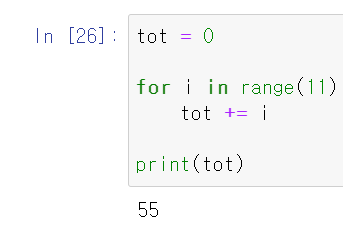
자동 생성된 설명

for문에도 continue 적용 가능

텍스트이(가) 표시된 사진

자동 생성된 설명

Range 함수 : 수열을 만들어주는 함수, 시작인덱스와 증감값은 초기값 0, 1로서 생략 가능, 종료 인덱스는 생략됨



활용

테이블이(가) 표시된 사진

자동 생성된 설명

구구단 2단 출력

텍스트이(가) 표시된 사진

자동 생성된 설명

구구단 전체 출력

텍스트이(가) 표시된 사진

자동 생성된 설명

리스트 내포 : 리스트 작성에 (입력값) for (변수) in (리스트)

코드도 가독성이 있고 실제 실행 속도도 빠름

텍스트이(가) 표시된 사진

자동 생성된 설명

뒤에 if문을 붙여서 조건에 맞는 식만 리스트에 넣을 수도 있다

4/4 강의노트

텍스트이(가) 표시된 사진

자동 생성된 설명

함수 정의 def : 여러 번 사용할 코드는 함수로 정의해서 반복 사용 가능

텍스트이(가) 표시된 사진

자동 생성된 설명

매개변수가 있는 함수 정의 : return을 이용하여 결과를 반환해야 결과가 남겨진다.

텍스트이(가) 표시된 사진

자동 생성된 설명

매개변수와 인수의 정의, 구분하기

텍스트이(가) 표시된 사진

자동 생성된 설명

함수에 매개변수가 없을수도 있고 그러한 경우에도 결괏값을 변수에 반환받을 수 있다

텍스트이(가) 표시된 사진

자동 생성된 설명

결괏값이 없는 함수도 있으며, 함수 프로세스에 따라 특정 값을 출력만 할 뿐 아무 값을 반환하지 않을 수 있다

데이터가 없는 ‘None” 이라는 데이터 타입도 있다

텍스트이(가) 표시된 사진

자동 생성된 설명

입력값도 결괏값도 없는 함수 지정

텍스트이(가) 표시된 사진

자동 생성된 설명

매개변수가 다양하면 함수 사용에 순서가 헷갈릴수도 있으므로 순서 상관없이 매개변수를 직접 지정하여 인수로서 입력할 수도 있다

텍스트이(가) 표시된 사진

자동 생성된 설명

함수의 반환값을 여러 개로 지정할수도 있으며, 이를 하나의 변수로 받으면 튜플 형식으로 반환됨

텍스트이(가) 표시된 사진

자동 생성된 설명

반환값이 여러 개로 지정하고 이를 각기 변수에 나누어 받을 경우

텍스트이(가) 표시된 사진

자동 생성된 설명

함수는 return을 만나는 순간 함수 실행이 종료됨

텍스트이(가) 표시된 사진

자동 생성된 설명

해당 특성을 이용하여 함수에서 break 대용이나 원하는 값을 반환시키고 종료시킬수도 있다

텍스트이(가) 표시된 사진

자동 생성된 설명

주피터 기능으로 함수(기본 + 커스텀 포함)에서 인수 입력자리에서 Shift + Tab을 누르면 매개변수 가이드가 뜸

함수에 따라 아무 값을 입력하지 않았을 때 디폴트로 입력되는 매개변수가 있음 (ex. Print 함수에서 디폴트 매개변수에 따라 기본적으로 \n (줄바꿈)을 하게 됨 end = 를 정의함으로서 줄바꿈을 하지 않고 다른 명령을 실행할 수도 있음)

커스텀 함수에서도 이를 정의할 수 있음

텍스트이(가) 표시된 사진

자동 생성된 설명

디폴트 매개변수 정의는 함수 생성시 뒤쪽에 들어가야 함 (ex. 위에서 27이 man에 들어가야 할지 old에 들어가야 할지 구분할 수 없다) 안지키면 함수 자체가 에러 뜨면서 생성이 안됨

텍스트이(가) 표시된 사진

자동 생성된 설명

함수 내에서 생성된 변수는 함수 외부에서 사용할 수 없음(지역변수)

텍스트이(가) 표시된 사진

자동 생성된 설명

반대로 함수 외부에서 생성한 변수는 함수 내부에서도 정상적으로 사용 가능(전역변수)

텍스트이(가) 표시된 사진

자동 생성된 설명

전역변수를 함수 내에서 값을 변경하고자 하면 global을 사용하여 해당 변수 값을 변경하겠다는 선언을 할 수 있음 (값의 변질을 추적하기 어렵기 때문에 일반적으로 비추천)

4/6 강의

Class : 객체를 만들어내는 틀의 개념. 생성된 객체간에 형태는 다르지만 기본은 같음 (와플틀 – 여러 종류의 와플, 게임 캐릭터들은 동일한 부류의 스텟을 사용하지만 그 스텟의 값은 캐릭터별로 다르다)

텍스트이(가) 표시된 사진

자동 생성된 설명

함수로는 하나의 프로세스만 실행 가능

텍스트이(가) 표시된 사진

자동 생성된 설명

Class 생성 : \_\_init\_\_ 문법이므로 언더바 양쪽 2개는 고정명 (노란색), self는 class의 고유 객체 개념으로 마찬가지로 문법 취급, 다른 입력은 불가(초록색)

텍스트이(가) 표시된 사진

자동 생성된 설명

Pass : 아무 기능이 없으나 최소한의 형태만 갖추게 해줌 (입력 안했다는 오류를 안뜨게 해줌)

\_\_main\_\_ : 파이썬이 실행되는 영역 (큰 의미는 없음)

텍스트이(가) 표시된 사진

자동 생성된 설명

Type를 찍으면 어느 class에서 만들어진 객체인지 확인 가능

텍스트이(가) 표시된 사진

자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명

Cal1, cal2 = 클래스로 만들어진 인스턴스 (= 객체)

self.result = 인스턴스 변수 (self.는 인스턴스 변수를 선언하는 문법

Add와 같이 (self가 들어간) 인스턴스 변수를 사용하는 함수가 인스턴스 메소드. 첫번째 매개변수로 self가 반드시 들어가야함

텍스트이(가) 표시된 사진

자동 생성된 설명

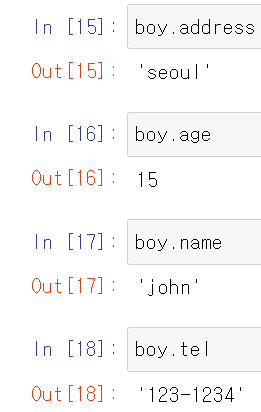
Person 클래스의 boy 객체(인스턴스) 생성

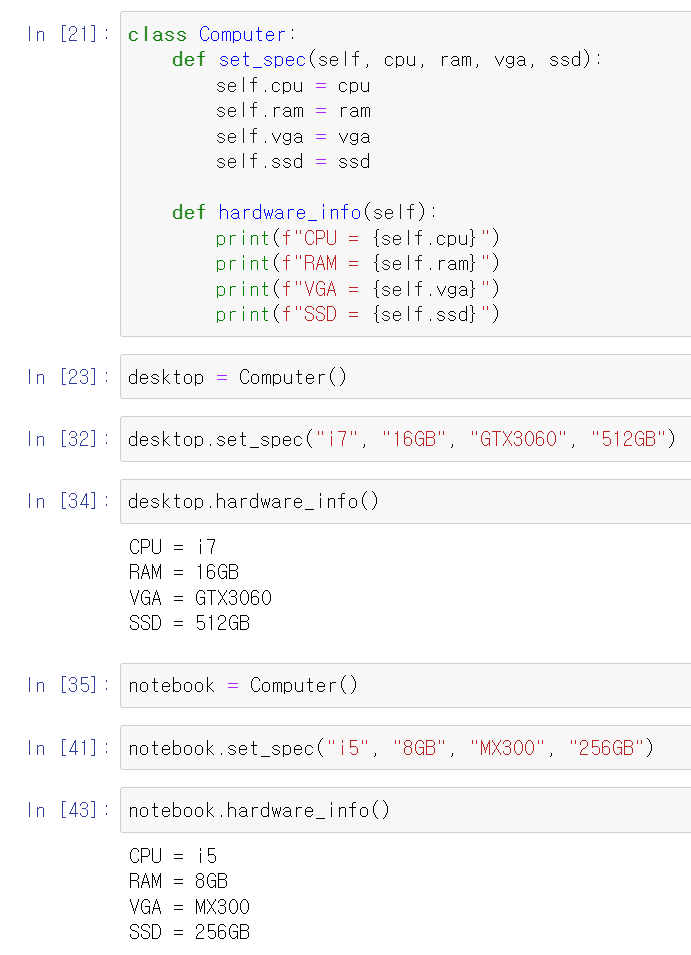
이 단계에서 boy는 who\_am\_i 메소드가 실행되지 않았으므로 name ~ address 변수가 없는 상태

텍스트이(가) 표시된 사진

자동 생성된 설명

Who\_am\_i 메소드를 실행시켜야 변수가 생김 (객체) + (.) 까지 입력하고 tab을 누르면 해당 객체가 사용할 수 있거나 가지고 있는 메소드와 인스턴스 변수를 선택할 수 있음 (주피터 기능)





객체별로 인스턴스 변수값이 다르므로 같은 메소드를 실행했을 때 나타나는 값도 다르다

텍스트이(가) 표시된 사진

자동 생성된 설명

add 함수의 result는 add 함수의 지역변수

텍스트이(가) 표시된 사진

자동 생성된 설명

클래스 내의 지역변수 result는 클래스내 다른 함수나 클래스 외부에서 불러올 수 없다

텍스트이(가) 표시된 사진

자동 생성된 설명

사칙연산을 수행하는 클래스 생성

텍스트이(가) 표시된 사진

자동 생성된 설명

생성자 : 객체가 생성될 때 즉시 실행되는 메소드 (\_\_init\_\_ 명의 함수) 즉시 실행되므로, 객체 생성 시에 생성자의 매개변수가 있을 경우 이를 입력해야만 객체가 생성됨

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

모듈 : 함수나 변수, 클래스를 모아둔 파일 다른 파이썬 프로그램에서 불러와서 사용할 수 있음

텍스트이(가) 표시된 사진

자동 생성된 설명

모듈을 불러와 해당 모듈 내의 함수 실행 from으로 불러오지 않으면 직접 실행은 불가함

텍스트이(가) 표시된 사진

자동 생성된 설명

From 모듈 import 함수를 사용하면 함수를 직접 사용할 수 있다

텍스트이(가) 표시된 사진

자동 생성된 설명

Import \*을 쓰면 모듈 내 모든 함수, 변수 등을 불러옴 (추천은 안함)

텍스트이(가) 표시된 사진

자동 생성된 설명

모듈명이 길 경우 as로 별명을 붙일수도 있음

텍스트이(가) 표시된 사진

자동 생성된 설명

모듈 내 함수도 별명을 붙일 수 있음