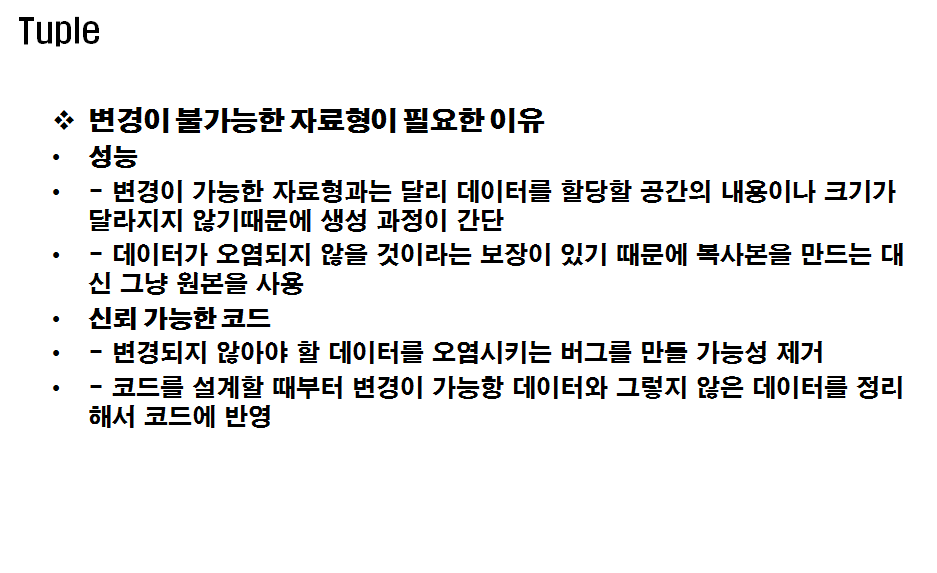
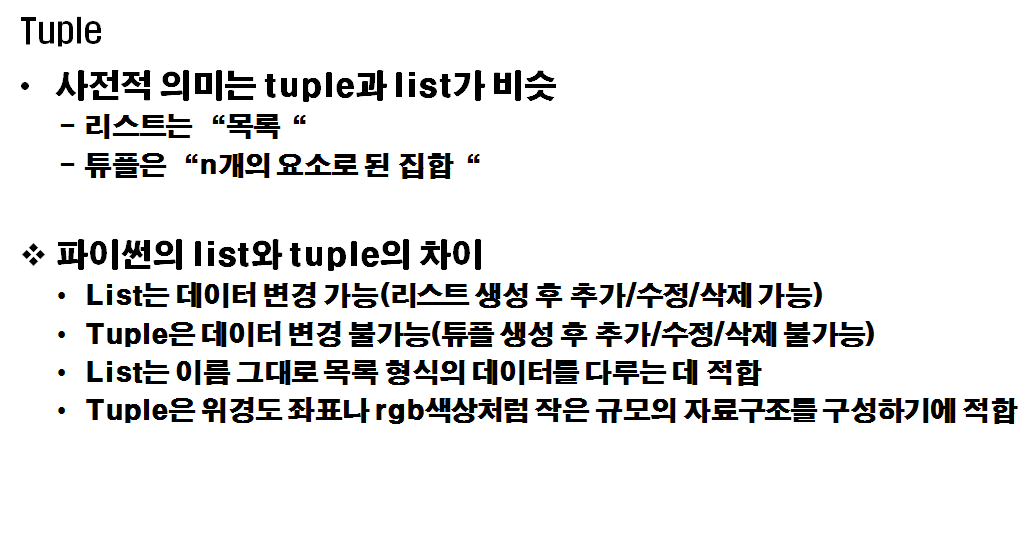
**Python기초 (Day 7)**

※ 튜플 : 바뀌면 안 되는 자료를 만질 때 굉장히 유용하다 , ★절대로 내부 자료변경이 불가능하다.

■ 튜플의 생성 : 리스트는 대괄호[]로 생성하고 튜플은 괄호()로 생성 , 하나의 값을 저장 할 때는 반드시 , (콤마)를 붙여줘야 한다 , 괄호 생략이 가능하다.

☞ 입력) tp = (10,20,30);

print("tp:",tp);

print("tp type:",type(tp));

print("tp len:",len(tp));

☞ 출력) tp: (10, 20, 30)

tp type: <class 'tuple'>

tp len: 3

☞ 입력) tp1 = 10,20,30;

print("tp1:",tp1);

print("tp1 type:",type(tp1));

print("tp1[0]type:",type(tp1[0]));

#tp1[0] = 100 #에러 발생 [tuple내부 데이터 변경 불가능하기 때문에 오류]

☞ 출력) tp1: (10, 20, 30)

tp1 type: <class 'tuple'>

tp1[0]type: <class 'int'>

☞ 입력) tpType = "문자열",100,1.111;

print("tpType:",tpType);

print("type:",type(tpType));

print("tpType[0] type:",type(tpType[0]));

print("tpType[1] type:",type(tpType[1]));

print("tpType[2] type:",type(tpType[2]));

☞ 출력) tpType: ('문자열', 100, 1.111)

type: <class 'tuple'>

tpType[0] type: <class 'str'>

tpType[1] type: <class 'int'>

tpType[2] type: <class 'float'>

☞ 입력) tpint = (10); [ , 콤마 없으니까 int형으로][int형이기 때문에 내부데이터 변경 가능][, 콤마 넣는거 주의하기]

print("tpint:",type(tpint));

tpT1 = (10,); [ , 콤마 있으니까 tuple형으로][tuple형이기 때문에 내부 데이터 변경 불가능][이러한 이유 때문에 내부데이터 변경 하지 말아야 할 때 tuple 사용]

print("tpT1:",type(tpT1));

tpT2 = 10,; [ , 콤마가 있으니까 변수가 아니라 tuple로 인식]

print("tpT2:",type(tpT2));

☞ 출력) tpint: <class 'int'>

tpT1: <class 'tuple'>

tpT2: <class 'tuple'>

☞ 입력) tt1 = (10,20,30,40);

tt2 = tt1[0]+tt1[1]+tt1[2]+tt1[3];

print("튜플 합:%d"%tt2);

print("tt1[1:3]:",tt1[1:3]);

print("tt1[1:]:",tt1[1:]);

print("tt1[:3]:",tt1[:3]);

☞ 출력) 튜플 합:100

tt1[1:3]: (20, 30)

tt1[1:]: (20, 30, 40)

tt1[:3]: (10, 20, 30)

☞ 입력) a = 1,2,3;

b = 4,5,6;

c = a+b;

print("a:",a);

print("b:",b);

print("c:",c);

☞ 출력) a: (1, 2, 3)

b: (4, 5, 6)

c: (1, 2, 3, 4, 5, 6)

☞ 입력) pack = 1,2,"패킹";

print("packing\npack:",pack); [packing은 하나로 묶는 것]

a,b,c = pack; [unpacking 작업][tuple의 개수에 맞춰서 사용]

print("unpacking\na:",a,"b:",b,"c:",c); [unpacking은 하나로 묶은 것을 쪼개는 것][unpacking하면 내부데이터 변경가 능]

print("packing\npack:",pack);

a=10; [내부 데이터 변경]

pack = a,b,c; [내부데이터 변경 끝나면 다시 packing으로 묶어 주어야 한다][packing 작업]

print("packing\npack:",pack);

☞ 출력) packing

pack: (1, 2, '패킹')

unpacking

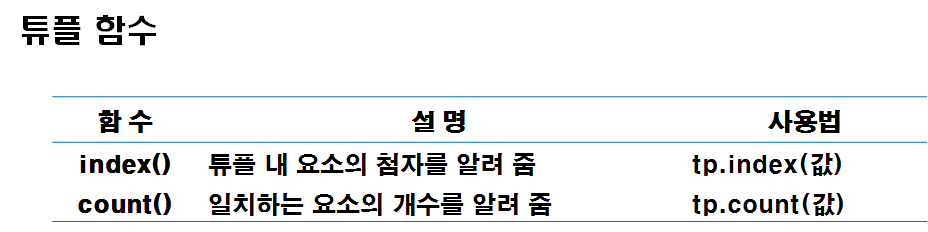
a: 1 b: 2 c: 패킹

packing

pack: (1, 2, '패킹')

packing

pack: (10, 2, '패킹')



☞ 입력) tp = 100,200,"함수",100,"개수";

print("tp안의 200의 위치:",tp.index(200),"번째 인덱스");

print("tp안의 100의 개수:",tp.count(100),"개");

☞ 출력) tp안의 200의 위치: 1 번째 인덱스

tp안의 100의 개수: 2 개

예시 코딩 문제 : 여러 개의 값을 패킹시킨 후 저장되어 있는 값을 빼내어 출력 하시오

(저장 되어 있는 값은 몇 개나 있는지 알 수 없는 상태이다)(5개 값 저장)

튜플의 값을 리스트에 저장하시오

☞ 입력) tp = 10,20,"test",3.3333,40;

ls = [];

for i in range(len(tp)):

ls.append(tp[i]);

print("tp :",tp);

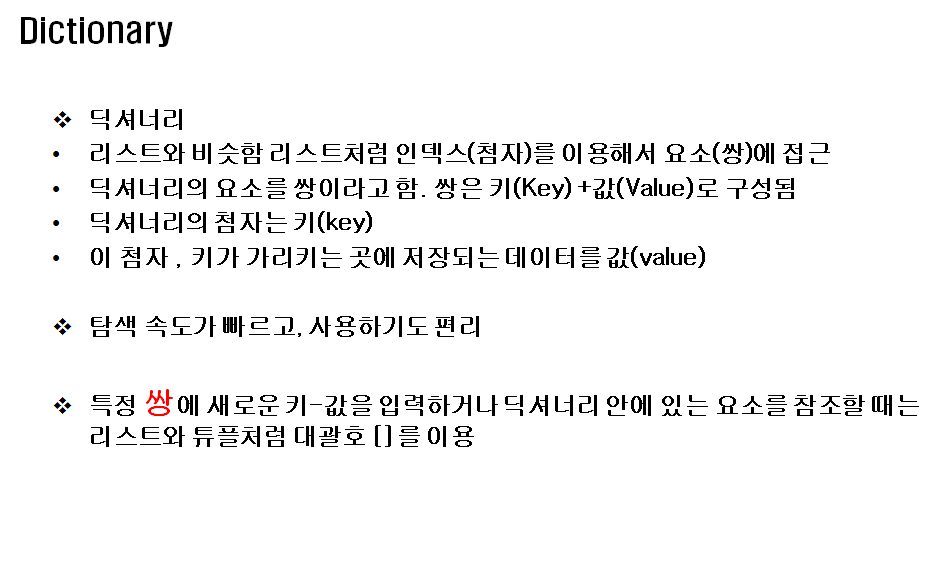
print("ls :",ls);

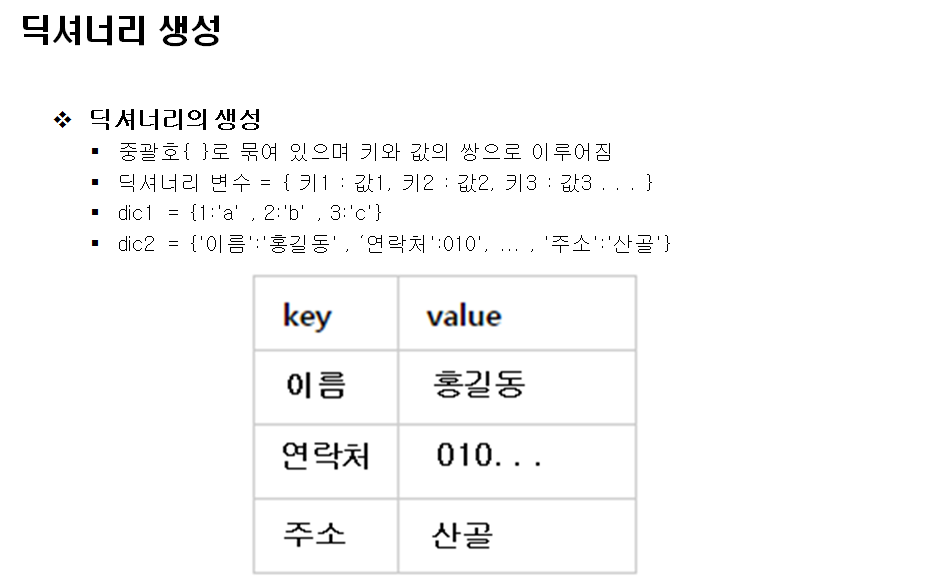
print(type(ls));

☞ 출력) tp : (10, 20, 'test', 3.3333, 40)

ls : [10, 20, 'test', 3.3333, 40]

<class 'list'>

※ 딕셔너리(Dictionary) :

■ 딕셔너리 생성 :

☞ 입력) student = {"학번":1234 , "이름":"홍길동" , "학과":"it학과"};

print(student);

mobile = {"품명":"겔럭시" , "가격":100 , "크기":10};

print(mobile);

print(student["학번"]); [list tuple처럼 개별적으로 호출 가능하지만 키(Key)로 호 출]

☞ 출력) {'학번': 1234, '이름': '홍길동', '학과': 'it학과'}

{'품명': '겔럭시', '가격': 100, '크기': 10}

1234

☞ 입력) impo = {}; [impo라는 딕셔너리 생성]

impo["파이썬"] = "www.python.org"; [키값 , 값 개별적으로 넣는 것 가능!!!]

impo["네이버"] = "www.naver.com"; [키값 , 값 개별적으로 넣는 것 가능!!!]

impo["구글"] = "www.google.com"; [키값 , 값 개별적으로 넣는 것 가능!!!]

print("파이썬:",impo["파이썬"]);

print("네이버:",impo["네이버"]);

print("구글:",impo["구글"]);

print(impo);

☞ 출력) 파이썬: www.python.org

네이버: www.naver.com

구글: www.google.com

{'파이썬': 'www.python.org', '네이버': 'www.naver.com', '구글': 'www.google.com'}

☞ 입력) impo = {};

name = input("키값 입력:");

val = input("값 입력:");

impo[name] = val;

print(name,":",impo[name]);

☞ 출력) 키값 입력:김태영

값 입력:1234

김태영 : 1234

예시 코딩 문제 : 하나의 대상을 선택하여 킷값과 값을 입력받아 저장 후 출력 하시오.

이름(key) 입력:겐지

값(value) 입력:수리검

이름(key) 입력:맥크리

값(value) 입력:섬광탄

이름(key) 입력:파라

값(value) 입력:로켓런처

이름(key) 입력:리퍼

값(value) 입력:샷건

이름(key) 입력:솔저

값(value) 입력:나선로켓

{“솔저”:“나선로켓” , “맥크리”:“섬광탄” , “리퍼”:“샷건” , “파라”:“로켓런처” , “겐지”:“수리검”}

☞ 입력) overwatch = {};

for i in range(0,5,1):

name = input("이름(key) 입력:");

val = input("값(value) 입력:");

overwatch[name] = val;

print(overwatch);

☞ 출력) 이름(key) 입력:겐지

값(value) 입력:수리검

이름(key) 입력:맥크리

값(value) 입력:섬광탄

이름(key) 입력:파라

값(value) 입력:로켓런처

이름(key) 입력:리퍼

값(value) 입력:샷건

이름(key) 입력:솔저

값(value) 입력:나선로켓

{'겐지': '수리검', '맥크리': '섬광탄', '파라': '로켓런처', '리퍼': '샷건', '솔저': '나선로켓'}

☞ 출력) 이름(key) 입력:1

값(value) 입력:1

이름(key) 입력:1

값(value) 입력:2

이름(key) 입력:1

값(value) 입력:3

이름(key) 입력:1

값(value) 입력:4

이름(key) 입력:1

값(value) 입력:5

{'1': '5'} [키값이 중복되면 지정한 value값을 가져와서 새로입력하기(수정) 때문에 마지막에 넣은 value값이 호출된다.]





☞ 입력) num = {1:"일" , 2:"이" , 3:"삼" , 4:"사"};

print("keys()키:",num.keys());

print();

print("values()값:",num.values());

☞ 출력) keys()키: dict\_keys([1, 2, 3, 4]) [dict\_keys라는 List로 뽑아냄]

values()값: dict\_values(['일', '이', '삼', '사'])

☞ 입력) num = {1:"일" , 2:"이" , 3:"삼" , 4:"사"};

print("num.keys():",num.keys());

print("list(num):",list(num)); [list로 딕셔너리 출력하면 key값만 출력]

print("list(num.keys()):",list(num.keys()));

print();

print("num.values():",num.values());

print("list(num.values()):",list(num.values())); [list로 value뽑고 싶을 때]

☞ 출력) num.keys(): dict\_keys([1, 2, 3, 4])

list(num): [1, 2, 3, 4]

list(num.keys()): [1, 2, 3, 4]

num.values(): dict\_values(['일', '이', '삼', '사'])

list(num.values()): ['일', '이', '삼', '사']

☞ 입력) singer = {};

singer["이름"]=input("가수 이름 입력:");

singer["구성원"]=input("인원수 입력:");

singer["대표곡"]=input("대표곡 입력:");

for i in singer.keys(): [i가 내부의 키 값“이름”,“구성원”,“대표곡”에 한번 씩 들 어감]

print("%s:%s"%(i,singer[i]));

print(singer);

☞ 출력) 가수 이름 입력:원

인원수 입력:4

대표곡 입력:원 투 쓰리

이름:원

구성원:4

대표곡:원 투 쓰리

{'이름': '원', '구성원': '4', '대표곡': '원 투 쓰리'}

☞ 입력) menu = {}; bl = True; num = 0;

while bl:

print("1.메뉴 등록");

print("2.메뉴별 가격 보기");

print("3.가격 수정");

print("4.종료");

num = int(input(">>>"));

if num == 1:

name = input("메뉴 입력:");

menu[name] = input("가격 입력:");

elif num == 2:

for i in menu.keys(): [i가 내부에 키 값들을 하나씩 찍음]

print(i,":",menu[i]);

elif num == 3:

name = input("수정할 목록 입력:");

menu[name] = input("수정가격:");

elif num == 4:

print("프로그램 종료 합니다");

bl = False;

☞ 출력) 1.메뉴 등록

2.메뉴별 가격 보기

3.가격 수정

4.종료

>>>1

메뉴 입력:김밥

가격 입력:1000원

1.메뉴 등록

2.메뉴별 가격 보기

3.가격 수정

4.종료

>>>1

메뉴 입력:떡볶이

가격 입력:2000원

1.메뉴 등록

2.메뉴별 가격 보기

3.가격 수정

4.종료

>>>2

김밥 : 1000원

떡볶이 : 2000원

1.메뉴 등록

2.메뉴별 가격 보기

3.가격 수정

4.종료

>>>3

수정할 목록 입력:김밥

수정가격:1500원

1.메뉴 등록

2.메뉴별 가격 보기

3.가격 수정

4.종료

>>>2

김밥 : 1500원

떡볶이 : 2000원

1.메뉴 등록

2.메뉴별 가격 보기

3.가격 수정

4.종료

>>>4

프로그램 종료 합니다

☞ 입력) num = {1:"일" , 2:"이" , 3:"삼" , 4:"사" , 5:"오"};

print(num);

print("num.get(3):",num.get(3)); [밑에 #붙은 코드와 똑같음]

#print("num.get(3):",num[3]);

print("num.get(9):",num.get(9,“test”));

#print("num.get(9):",num[9]); [9라는 킷값이 없기 때문에 오류 뜸]

☞ 출력) {1: '일', 2: '이', 3: '삼', 4: '사', 5: '오'}

num.get(3): 삼

num.get(9): None

num.get(9): test

☞ 입력) student = {"학번":1234 , "이름":"홍길동" , "학과":"it학과"};

print(student["학번"]);

print(student["이름"]);

print(student["학과"]);

print();

print(student.items());

print();

print(student);

☞ 출력) 1234

홍길동

it학과

dict\_items([('학번', 1234), ('이름', '홍길동'), ('학과', 'it학과')])

{'학번': 1234, '이름': '홍길동', '학과': 'it학과'}

☞ 입력) name = {"이순신":"거북선" , "세종대왕":"훈민정음" , "파이썬":"프로그래밍 언어"};

for key,value in name.items():

print(key,":",value);

print("삭제:",name.clear());

print("삭제 후 값:",name);

☞ 출력) 이순신 : 거북선

세종대왕 : 훈민정음

파이썬 : 프로그래밍 언어

삭제: None

삭제 후 값: {}

☞ 입력) num = {1:"일" , 2:"이" , 3:"삼" , 4:"사" , 5:"오"};

print("변경전 값:",num);

print();

print("num.setdefault(9):",num.setdefault(9,"구~우"));

print();

print("변경전 후:",num);

print();

print("num.get(9)번째 value:",num.get(9));

☞ 출력) 변경전 값: {1: '일', 2: '이', 3: '삼', 4: '사', 5: '오'}

num.setdefault(9): 구~우

변경전 후: {1: '일', 2: '이', 3: '삼', 4: '사', 5: '오', 9: '구~우'}

num.get(9)번째 value: 구~우

☞ 입력) dic = {};

ls = [];

ls.append(input("등록할 키값 입력:"));

ls.append(input("등록할 키값 입력:"));

ls.append(input("등록할 키값 입력:"));

dic = dic.fromkeys(ls); [list항목들을 모두 key값으로 받음]

print("dic키 설정:",dic);

dic=dic.fromkeys(ls,0); [뒤에 0을 입력하기 때문에 key값에 모두 0의 값 넣음]

print("dic 키&값 설정:",dic);

☞ 출력) 등록할 키값 입력:10

등록할 키값 입력:20

등록할 키값 입력:30

dic키 설정: {'10': None, '20': None, '30': None}

dic 키&값 설정: {'10': 0, '20': 0, '30': 0}