Python Basic 4

튜플

19. 튜플의 특징

- 시퀀스 자료형
- 수정, 추가, 삭제가 불가능한 리스트
- 읽기만 가능하기 때문에 데이터 손실 염려가 없음

20. 튜플 만들기

튜플 = (데이터, 데이터, 데이터)

튜플 = 데이터, 데이터, 데이터

① a = (3, 4, 5)

a = "DCU캠퍼스", 3, True

20. 튜플 만들기

튜플 = (데이터, 데이터, 데이터)

튜플 = 데이터, 데이터, 데이터

② a = (30,)

a = 30,

21. 리스트를 튜플로 만들기

① a = tuple([5, 6, 7])

② x = list(range(10))

a = tuple(x)

22. 튜플을 리스트로 만들기

③ x = 5, 6, 7

a = list(x)

23. 패킹과 언패킹

- 패킹 : 여러개의 데이터를 하나의 변수에 할당하는 것
- 언패킹 : 컬렉션의 각 데이터를 각각의 변수에 할당하는 것

23. 패킹과 언패킹

```
numbers = 3, 4, 5          #패킹

a, b, c = numbers          #언패킹
```

23. 패킹과 언패킹

```
numbers = [3, 4, 5]       #패킹

a, b, c = numbers         #언패킹
```

24. 튜플 함수

a = 10, 20, 30, 40, 30

①	특정값의 인덱스 구하기	a.index(20)
②	특정값의 개수	a.count(30)
③	최대값, 최소값	max(a), min(a)
④	합계	sum(a)

튜플

실습

딕셔너리

25. 딕셔너리의 특징

- 시퀀스 자료형
- 키와 데이터를 가지고 있는 사전형 자료형
- 사전형태의 자료를 만들 때 편리

26. 딕셔너리 만들기

딕셔너리 = {키1 : 데이터1, 키2 : 데이터2}

① stock = {"삼성전자" : 82000, "LG전자" : 150000}

26. 딕셔너리 만들기

② stock = {

"삼성전자" : [81000, 81500, 82000, 81500, 82000]

"LG전자" : (150000, 149000, 148000, 151000, 152000)

}

27. 데이터 접근하기

딕셔너리["키"]

① stock["삼성전자"]

28. 데이터 할당하기

```
딕셔너리["키"] = 데이터
```

```
stock["삼성전자"] = 85000
```

29. 데이터 삭제하기

```
del 딕셔너리["키"]
```

```
del stock["삼성전자"]
```

30. 딕셔너리 함수

```
stock = {
        "삼성전자" : 82000,
        "SK하이닉스" : 123000,
        "NAVER" : 370000,
        "카카오" : 133000
}
```

30. 딕셔너리 함수

①	키와 데이터 쌍	stock.items()
②	키	stock.keys()
③	데이터	stock.values()

딕셔너리

실습

감사합니다.