



강좌 검색

홈 » 내 강좌 » 파이썬 코딩 도장 » 핵심 정리 » 핵심 정리

◀ 26.9 심사문제: 공약수 구하기

Q & A ▶



요즘 핫한 개발직군 연봉

희망연봉 작성 전 원티드에서 직군
연봉 확인하기

원티드

핵심 정리

딕셔너리 메서드

▼ 표 26-1 딕셔너리 메서드

메서드	설명
setdefault(키, 기본값)	딕셔너리에 키-값 쌍을 추가, 키만 지정하면 값에 None 을 저장
update(키=값) update(딕셔너리) update(리스트) update(튜플) update(반복가능한객체)	딕셔너리에서 키의 값을 수정, 딕셔너리에 키가 없으면 키-값 쌍을 추가
pop(키) pop(키, 기본값)	딕셔너리에서 특정 키-값 쌍을 삭제한 뒤 값을 반환, del 딕셔너리[키] 와 같음 기본값을 지정하면 키가 없을 때 기본 값을 반환

popitem()	딕셔너리에서 임의의 키-값 쌍을 삭제하고 해당 키-값 쌍을 튜플로 반환
clear()	딕셔너리의 모든 키-값 쌍을 삭제
get(키) get(키, 기본값)	딕셔너리에서 특정 키의 값을 가져옴 기본값을 지정하면 키가 없을 때 기본값을 반환
items() keys() values()	items는 딕셔너리의 키-값 쌍을 모두 가져옴 keys는 딕셔너리의 키를 모두 가져옴 values는 딕셔너리의 값을 모두 가져옴
copy()	딕셔너리를 복사하여 새 딕셔너리 생성
dict.fromkeys(키리스트) dict.fromkeys(키리스트, 값)	리스트(튜플)로 딕셔너리를 만들, 키 리스트만 지정하면 값은 모두 None 이 저장됨 키 리스트와 값을 지정하면 해당 값이 딕셔너리의 값으로 저장됨

딕셔너리와 반복문

for 변수 in 뒤에 딕셔너리를 지정하면 키만 꺼내옵니다. 그리고 for in 뒤에 딕셔너리를 지정하고 items를 사용하면 반복하면서 모든 키와 값을 꺼내오고, keys는 키, values는 값만 꺼내옵니다.

```

for 변수 in 딕셔너리:           # 모든 키를 꺼내옴
    반복할 코드

for 키, 값 in 딕셔너리.items():  # 모든 키와 값을 꺼내옴
    반복할 코드

for 키 in 딕셔너리.keys():       # 모든 키를 꺼내옴
    반복할 코드

for 값 in 딕셔너리.values():     # 모든 값을 꺼내옴
    반복할 코드

```

딕셔너리 표현식

딕셔너리 표현식은 딕셔너리 안에 식, for 반복문, if 조건문 등을 지정하여 딕셔너리를 생성합니다.

```

{키: 값 for 키, 값 in 딕셔너리}
{key: value for key, value in dict.fromkeys(['a', 'b', 'c', 'd']).items()}
dict({키: 값 for 키, 값 in 딕셔너리})

{키: 값 for 키, 값 in 딕셔너리 if 조건식}
{key: value for key, value in {'a': 10, 'b': 20, 'c': 30, 'd': 40}.items() if value != 20}
dict({키: 값 for 키, 값 in 딕셔너리 if 조건식})

```

딕셔너리 안에 딕셔너리 사용하기

딕셔너리는 값 부분에 딕셔너리를 넣을 수 있습니다. 딕셔너리 안에 들어있는 딕셔너리에 접근하려면 [](대괄호)를 단계만큼 붙이고 키를 지정해줍니다.

```
딕셔너리 = {키1: {키A: 값A}, 키2: {키B: 값B}}    # 딕셔너리 안에 딕셔너리 넣기

딕셔너리[키][키]          # 딕셔너리 안에 있는 딕셔너리에서 값에 접근
딕셔너리[키][키] = 값     # 딕셔너리 안에 있는 딕셔너리에서 키에 값 할당
```

세트

세트는 집합 형태의 자료형입니다. { }(중괄호) 안에 값을 저장하며 각 값은 ,(콤마)로 구분해줍니다. 단, 세트는 [](대괄호)로 특정 요소를 출력할 수 없습니다.

```
세트 = {값1, 값2, 값3}          # 세트 만들기
세트 = set(반복가능한객체)      # 세트 만들기

값 in 세트                      # 세트에 특정 값이 있는지 확인
값 not in 세트                  # 세트에 특정 값이 없는지 확인

len(세트)                      # 세트의 요소 개수(길이) 구하기
```

세트의 메서드와 집합 연산

▼ 표 26-2 세트의 메서드와 집합 연산

메서드	집합 연산자	설명
set.union(세트1, 세트2)		두 세트의 합집합
set.intersection(세트1, 세트2)	&	두 세트의 교집합
set.difference(세트1, 세트2)	-	두 세트의 차집합
set.symmetric_difference(세트1, 세트2)	^	두 세트의 대칭차집합
update(다른세트)	=	현재 세트에 다른 세트를 더함
intersection_update(다른세트)	&=	현재 세트와 다른 세트 중에서 겹치는 요소만 현재 세트에 저장
difference_update(다른세트)	-=	현재 세트에서 다른 세트를 뺌
symmetric_difference_update(다른세트)	^=	현재 세트와 다른 세트 중에서 겹치지 않는 요소만 현재 세트에 저장

issubset(다른세트)	<=	현재 세트가 다른 세트의 부분집합인지 확인
	<	현재 세트가 다른 세트의 진부분집합인지 확인
issuperset(다른세트)	>=	현재 세트가 다른 세트의 상위집합인지 확인
	>	현재 세트가 다른 세트의 진상위집합인지 확인
isdisjoint(다른세트)		현재 세트가 다른 세트와 겹치지 않는지 확인
add(요소)		세트에 요소를 추가
remove(요소)		세트에서 특정 요소를 삭제, 없으면 에러 발생
discard(요소)		세트에서 특정 요소를 삭제, 요소가 없으면 그냥 넘어감
pop()		세트에서 임의의 요소를 삭제하고 해당 요소를 반환
clear()		세트에서 모든 요소를 삭제
copy()		세트를 복사하여 새로운 세트 생성

세트와 반복문

for 변수 in 뒤에 세트를 지정하면 반복하면서 모든 요소를 꺼내옵니다.

```
for 변수 in 세트:
    반복할 코드
```

세트 표현식

세트 표현식은 세트 안에 식, for 반복문, if 조건문 등을 지정하여 세트를 생성합니다.

{식 for 변수 in 반복가능한값}	{식 for 변수 in 세트 if 조건식}
{i for i in 'apple'}	{i for i in 'pineapple' if i not in 'apl'}
set(식 for 변수 in 반복가능한값)	set(식 for 변수 in 세트 if 조건식)