

[Python] 파이썬 기초 4 - 딕셔너리(Dictionary)에 대한 정의와 기본적인 함수 사용법

노트북: [TIL-MY]

만든 날짜: 2020-08-12 오후 10:36

URL: <https://continuous-development.tistory.com/64>

나무늘보의 개발 블로그

홈

태그



분류 전체보기 

Python 

Database

ASP.NET

Algorithm

Machine learning
| Deep lear..



파이썬 딕셔너리

- # 범용적으로 가장 많이 사용되는 타입이다.
- # 딕셔너리의 특징은 key 와 value의 대응 관계를 가지고 있는 type이다.
- # 순서x, 키중복 x, 수정 O, 삭제 O
- # {} 이와 같은 형태를 가진다.

※상식

oop라는 것은 현실에 존재하는 object(모든 객체들을)를 객체로서 프로그램에 넣어주는 작업을 한다.

클래스라는 것은 현실에 존재하는 명사적 특징과 동사적 특징을 한 틀에 넣어준다.

클래스는 프로그램상의 객체는 아니고 객체

AWS

ETC..

R 

공지사항

글 보실 때 주의사항

: 최근글 : 인기글

[Python] 파이썬 기초 8

2020.08.12

[Python] 파이썬



2020.08.11

[Python] 파이썬



2020.08.11

[Python] 파이썬



2020.08.11

[Python] 파이썬



2020.08.11

최근댓글

태그

Oracle SQL,
python for문,
행렬, 설정,

자체를 만들어 낼 수 있는 템플릿이다.
dict이라는 클래스로 부터 생겨난 객체이다.

선언

```
temp = {}  
print(type(temp)) # <class 'dict'>
```

dict 생성 방법 1

```
dic01 = {'name': 'seop',  
        'age': 48,  
        'address': 'seoul',  
        'birth': '730910',  
        'gender': True}
```

dict 생성 방법 2

```
dic02 = dict([('name', 'seop'),  
             ('age', 48),  
             ('address', 'seoul'),  
             ('birth', '730910'),  
             ('gender', True)])
```

dict 생성 방법 3

```
dic03 = dict(name='seop',  
            age=48,
```

python if else,
python if문,
python while문,
Python,
파이썬 함수,
파이썬 교집합,
날짜함수,
파이썬 while문,
Oracle,
python date type,
python if elif else,
SQL, 인스턴스,
사용법, AWS,
파이썬 if문, DDL,
substr, rbind,
python 데이트 타
입,
ggplot, cbind,
테이블 생성,
파이썬 datatype,
파이썬,
파이썬 date

전체 방문자

173

Today : 6

Yesterday : 5

```

address='seoul',
birth='730910',
gender=True)

print('dic -', type(dic01), dic01)

# dic - <class 'dict'> {'name': 'seop', 'age': 48, 'address': 'seoul', 'brith':
# '730910', 'gender': True}

```

dict의 내장함수 보기

```

print(dir(dic01))
# 해당 객체가 어떤 변수와 메소드(method)를
# 가지고 있는지 나열
- dict가 가지고 있는 변수와 메소드 나열

#[ '__class__', '__contains__', '__delattr__',
# '__delitem__', '__dir__', '__doc__', '__eq__',
# '__format__',....]

```

해당 키값의 유무를 판단

```

print('name' in dic01) # True

```

요소를 추가하는 방법

```

dic01['marriage'] = False
print(dic01) # {'name': 'seop', 'age': 48,

```

```
'address': 'seoul', 'brith': '730910', 'gender': True, 'marriage': False}
```

삭제 pop() , del

```
print("삭제")
del dic03['gender']
print(dic03) # {'name': 'seop', 'age': 48, 'address': 'seoul', 'brith': '730910'}

print('pop -', dic03.pop('birth')) # pop - 730910
print('dic03 - ', dic03) # dic03 - {'name': 'seop', 'age': 48, 'address': 'seoul'}
```

요소를 수정하는 방법

```
dic01['marriage'] = True
print(dic01) # {'name': 'seop', 'age': 48, 'address': 'seoul', 'brith': '730910', 'gender': True, 'marriage': True}
```

데이터 확인

```
print(dic01['birth']) # 730910
```

데이터 출력

```
print('dic03 - ', dic03['name']) # dic03 - seop
# print('dic03 - ', dic03['Name']) # dic03 - seop / 이름으로 가져올경우 잘못 가져오면 에러가 난다.
print('dic03 - ', dic03.get('name')) # dic03 - seop
print('dic03 - ', dic03.get('Name')) # dic03 - seop / get을 사용할때는 에러가 나지 않는다.
```

길이 값

```
print('len - ', len(dic03)) # len - 5
```

dict_keys - key 값을 반환

```
print('dict_keys - ', dic03.keys()) # dict_keys - dict_keys(['name', 'age', 'address', 'brith', 'gender']) - dict 타입
print('dict_keys - ', list(dic03.keys())) # dict_keys - ['name', 'age', 'address', 'brith', 'gender'] - list 타입으로 변환
```

dict_values - values 값을 반환

```
print('dict_keys - ', dic03.values()) # dict_keys - dict_values(['seop', 48, 'seoul', '730910', True]) - dict 타입
print('dict_keys - ', list(dic03.values())) # dict_keys - dict_values(['seop', 48, 'seoul', '730910', True]) - dict 타입
```

dict_items - key와 value를 반환

```

print('dict_keys -',
dic03.items()) # dict_keys - dict_items
([('name', 'seop'), ('age', 48), ('address', 'seoul'), ('brith', '730910'), ('gender', True)])
print('dict_keys -', list(
dic03.items())) # dict_keys - [('name', 'seop'), ('age', 48), ('address', 'seoul'), ('brith', '730910'), ('gender', True)]#

```

for문으로 값 반환하기

```

for key in dic03.keys():
    print("{0},{1}".format(key, dic03[key]))
# result
# name,seop
# age,48
# address,seoul
# brith,730910
# gender,True

for value in dic03.values():
    print(value)
# seop
# 48
# seoul
# 730910
# True

for (key, value) in dic03.items():
    print("key : {0}, value : {1}".format(key, value))
# key : name, value : seop
# key : age, value : 48
# key : address, value : seoul
# key : brith, value : 730910
# key : gender, value : True

```



'Python' 카테고리의 다른 글

[Python] 파이썬 기초 6 - Date type 에 대한 정의와 기본적인 함수 사용 법 (0)	2020.08.11
[Python] 파이썬 기초 5 - 집합(set)에 대한 정의와 기본적인 함수 사용법 (0)	2020.08.11
<u>[Python] 파이썬 기초 4 - 딕셔너리(D ictionary)에 대한 정의와 기본적인 함 수 사용법 (0)</u>	2020.08.11
[Python] 파이썬 기초 3 - 튜플(tuple) 에 대한 정의와 기본적인 함수 사용 법 (0)	2020.08.10
[Python] 파이썬 기초 2 - 리스트(list) 에 대한 정의와 기본적인 함수 사용 법 (0)	2020.08.10
[Python] 파이썬 기초 1 - 변수(variab le)에 대한 타입과 기본적인 함수 사 용법 (0)	2020.08.10

태그

[Dict](#)[python dict](#)[python dictionary](#)[파이썬 dictionary](#)[파이썬 딕셔너리](#)

관련글



[Python] ...

[Python] ...

[Python] ...

[Python] ...

댓글 0

< 1 2 3 4 5 6 7 8
9 ... 67 >

TEL. 02.1234.5678 / 경기 성남시 분당구 판교역로

© Kakao Corp.

