

MemoNote semi Project 4

카페24

200만 고객이 사용하는 무료쇼핑몰. 마켓 판매, 글로벌 쇼핑몰, 마케팅, 엔터프라이즈 서비스 제공

 <https://www.cafe24.com/>

cafe24

Python Basic

List



```
a = []
b = [1,2,3]
c = ['a','b','c']
d = [1, 2, 'a', 'b']

print(b[0]) #인덱싱
print(c[1:]) #슬라이싱
print(d[-1]) #마지막 요소는 -1

# a[0] = 9 # a 리스트에는 아무 원소도 없어서 에러 발생
b[0] = 9 # b 리스트는 원소가 있으므로 수정 가능
print(b)

print(c)
del c[2] # 리스트는 삭제가 가능하다.
print(c)

print(a)
a.append(0) # 리스트는 삽입이 가능하다.
print(a)
```



리스트에서 많이 사용되는 함수 정리

요소 추가 append 마지막에 요소추가

```
a.append(0)
print(a)
a.append('a')
print(a)
```

요소 추가 insert 특정 위치를 지정하여 맘대로 요소 추가

```
d = [1,2,3]
d.insert(0,4)
```

정렬 sort

```
a = [1,3,2,4]
a.sort()
print(a)
```

뒤집기 reverse

```
a.reverse()
print(a)
```

인덱스 index

```
a.index(3)
# a.index(5)   없는 요소는 에러 발생한다.
```



```
# 삭제 remove 중복되는 수는 처음 등장하는 하나만 삭제한다.
print()
a = [1,2,3,3,5]
a.remove(3)
print(a)

# 삭제 pop 리스트의 마지막 요소를 삭제한다.
a = [1,2,3]
a.pop() # 괄호에 값이 없으면 마지막 요소를 하나 삭제
print(a)

a = [1,2,3]
a.pop(0) # 괄호에 값이 있으면 그 요소를 삭제
print(a)

# 개수 세기 count 특정 요소의 갯수 세기
a = [1,2,3,3]
print(a.count(3))
```



```
[1, 2, 3, 5]
[1, 2]
[2, 3]
2
```

Tuple



```
# TUPLE
# () 소괄호로 만든다.
# immutable 수정이 불가능하다.
# indexing & slicing 이 가능하다.
# 한 개의 요소로 tuple을 만들때는 (1,) 쉼표를 괜히 넣는다.

a = ()
b = (1,) #한 개의 요소로 tuple을 만들때는 (1,) 쉼표를 괜히 넣는다.
c = (1,2,3)
d = (1,2,'a','b')

# del d[0]    immutable 삭제가 불가능하다.
# d[0] = 9    immutable 수정이 불가능하다.
# d.append(5) immutable 삽입이 불가능하다.

print(d[0])   # 인덱싱이 가능하다.
print(d[0:2]) # 슬라이싱이 가능하다. 마지막은 미만으로 처리한다.

print(len(d)) # 튜플 요소의 개수를 반환하는 함수
```



```
1
(1, 2)
4
```

Dict

```
# 딕셔너리에서 많이 사용되는 함수 정리
# 딕셔너리에서 키값들만 추출

a = {'이름': '홍길동', '나이': 20, '보유코인': ['비트코인', '에이다', '이더리움']}
print(a.keys())

# 딕셔너리에서 밸류값들만 추출
print(a.values())

# 쌍으로 추출 (튜플 쌍으로 추출)
print(a.items())

# 키값으로 밸류값 추출
print(a.get('이름'))
print(a.get('몸무게')) # 찾는 키값이 없을때 None 출력한다.

# 키값으로 밸류값 추출 (인덱싱이 불가능하기 때문에 꺼낼때 많이 사용된다.)
print(a['이름'])
# print(a['몸무게']) # 찾는 키값이 없을때 예외 발생된다.

# 키가 딕셔너리안에 있는지 확인
print('이름' in a) # 찾는 키값이 있을때 true 반환
print('몸무게' in a) # 찾는 키값이 없을때 false 반환
```

dict_keys(['이름', '나이', '보유코인'])
dict_values(['홍길동', 20, ['비트코인', '에이다', '이더리움']])
dict_items([('이름', '홍길동'), ('나이', 20), ('보유코인', ['비트코인', '에이다', '이더리움'])])
홍길동
None
홍길동
True
False

Set

```

# SET
# {} 중괄호로 만든다.
# mutable 수정 가능하다.
# indexing & slicing 이 불가능하다.
# unordered 순서가 없다.
# 중복데이터를 허용하지 않는다.

# a = {} # 딕셔너리와 셋은 동일하게 {} 를 사용한다. 따라서 빈 값으로 생성할때는 주의하자.
a = set()
b = {1,2,3}
c = {1,2,'a','b'}
d = {1,2,2,2,3,4,4}
print(d)

a.add(0).....# mutable 삽입 가능하다.
print(a)
print(c)
c.remove(2) # mutable 삭제 가능하다.
print(c)

```

```

{1, 2, 3, 4}
{0}
{'b', 1, 2, 'a'}
{'b', 1, 'a'}

```

4가지 자료형 표로 정리

| 자료형 | 설명 | 함수 |
|-------|---|---|
| List | # [] 대괄호로 만든다. # indexing & slicing 이 가능하다. # ordered 순서가 있다. # mutable 수정 가능하다. | # 요소 추가 append 마지막에 요소추가 a.append(0) # 요소 추가 insert 특정 위치를 지정하여 앞대로 요소 추가 d.insert(0,4) # 정렬 sort a.sort() # 뒤집기 reverse a.reverse() # 인덱스 index a.index(3) remove pop count |
| Tuple | # () 소괄호로 만든다. # immutable 삽입, 수정, 삭제가 불가능하다. # indexing & slicing 이 가능하다. # ordered 순서가 있다. # 한 개의 요소로 tuple 을 만들때는 (1,) 침표를 괜히 넣는다.# 튜플 요소의 개수를 반환하는 함수 len() | |

| 자료형 | 설명 | 함수 |
|------|--|--|
| Dict | <p># {} 중괄호로 만든다. # mutable 삽입, 수정, 삭제가 가능하다. # indexing & slicing 이 불가능하다. # unordered 순서가 없다. # 키값과 밸류 값으로 데이터가 저장된다. # 세상의 많은 데이터가 키값:밸류값 형태로 표현된다. 이름:홍길동, 나이:20 # 이러한 대응 관계를 파이썬에서 데이터를 저장하는 방식을 딕셔너리라고 한다. # 마치 JS 의 JSON 과 상당히 유사한 공통점을 가지고 있다.</p> | <p># 딕셔너리에서 키값들만 추출 a = {'이름':'홍길동', '나이':20, '보유코인':['비트코인','에이다','이더리움']} print(a.keys()) # 딕셔너리에서 밸류값들만 추출 print(a.values()) # 쌍으로 추출 (튜플 쌍으로 추출) print(a.items()) # 키값으로 밸류값 추출 print(a.get('이름')) print(a.get('몸무게')) # 찾는 키값이 없을때 None 출력한다. # 키값으로 밸류값 추출 (인덱싱이 불가능하기 때문에 꺼낼때 많이 사용된다.) print(a['이름']) # print(a['몸무게']) # 찾는 키값이 없을때 예외 발생된다. # 키가 딕셔너리안에 있는지 확인 print('이름' in a) # 찾는 키값이 있을때 true 반환 print('몸무게' in a) # 찾는 키값이 없을때 false 반환</p> |
| Set | <p># {} 중괄호로 만든다. # mutable 수정 가능하다. # indexing & slicing 이 불가능하다. # unordered 순서가 없다. # 중복데이터를 허용하지 않는다.</p> | |

a.index(3)

remove

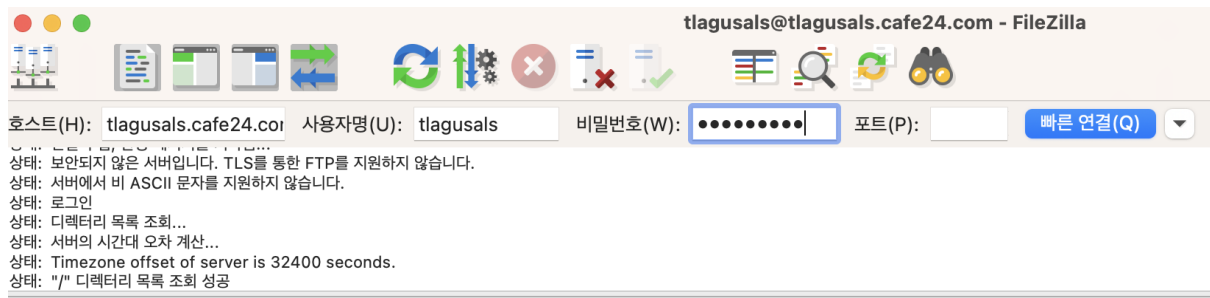
pop

count

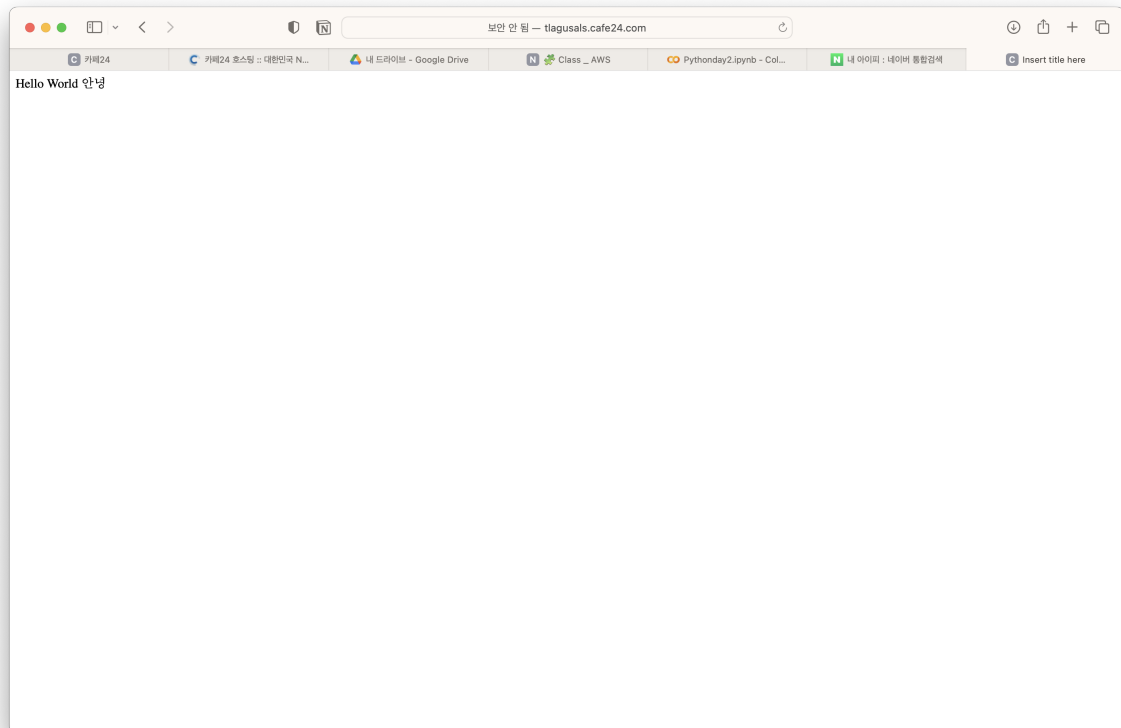
Cafe24

Basic

- Cefe24 Setting
- FTP. filezilla

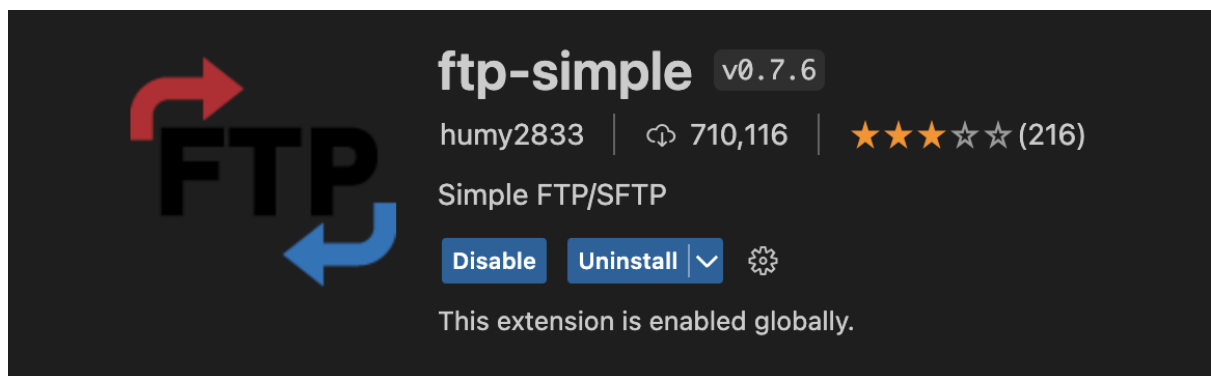


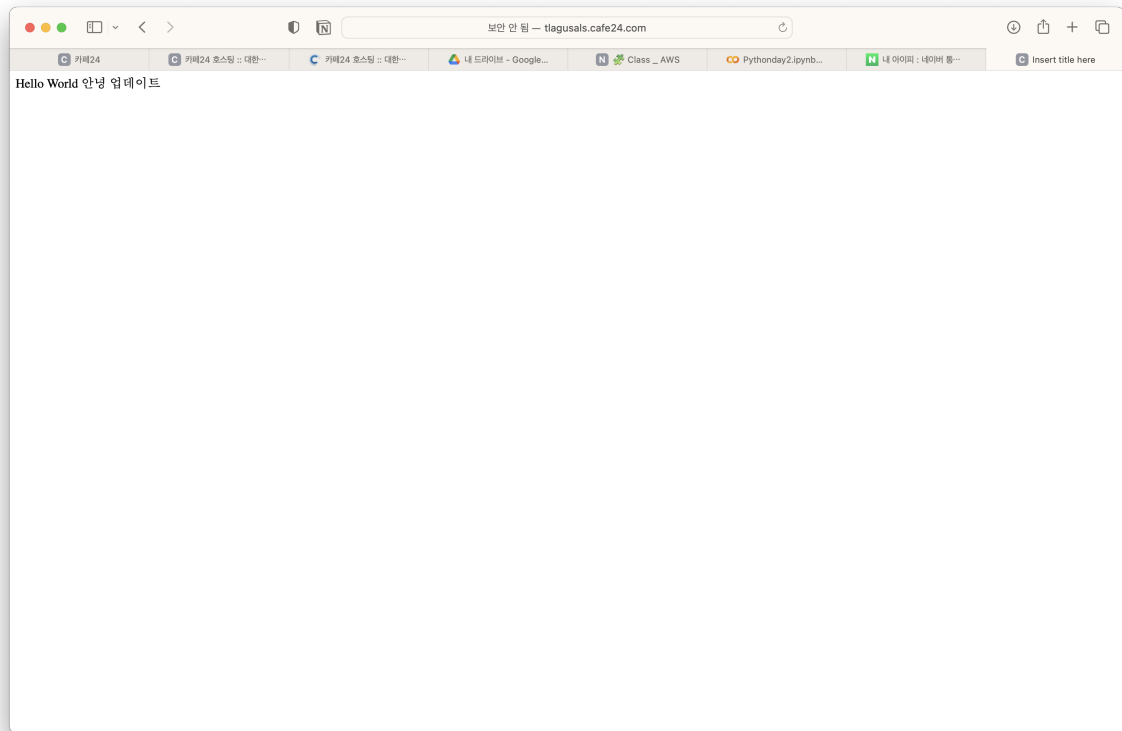
jdk11 호환 성공



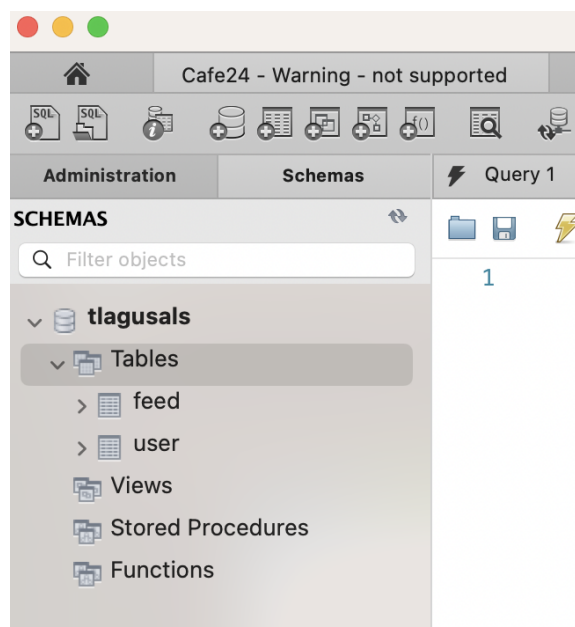
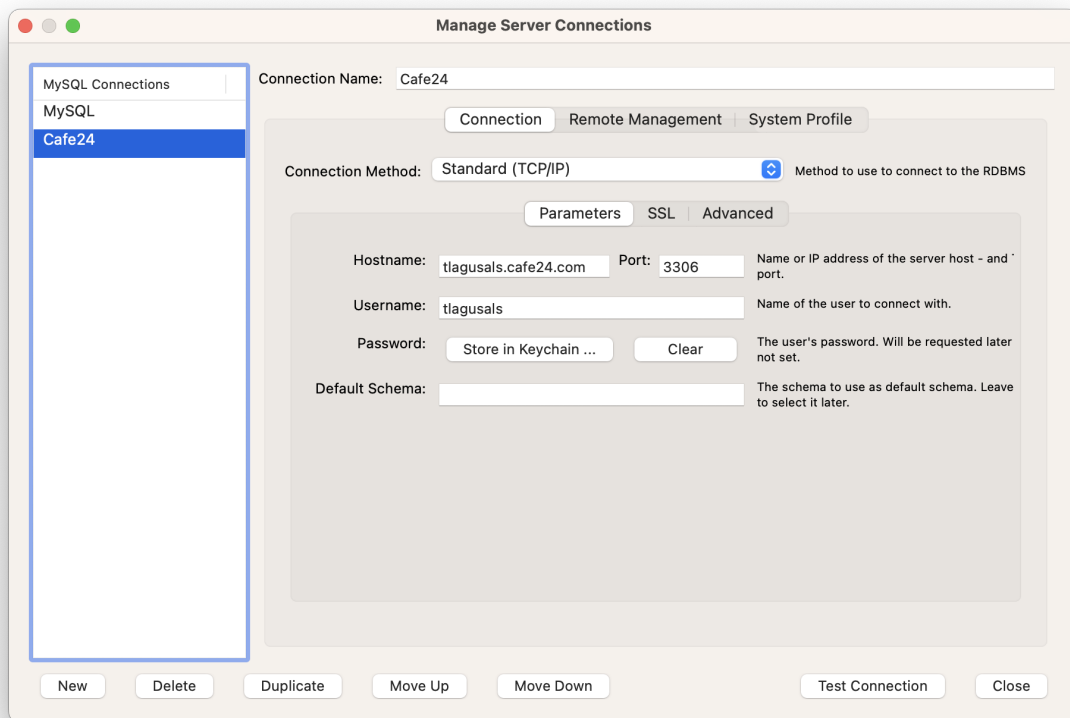
- VSCode

간단하게 수정 시





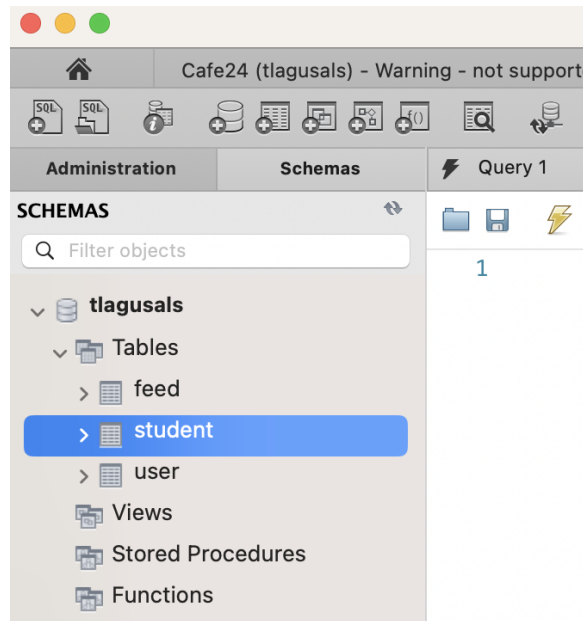
- workbench



- DB.jdbc. connectionpool

MySQL - maria 호환 성공

db연결해서 테이블 생성 성공



Adv

DB Setting. new db user

ROOT war