


Class[MemoNote semi Project 4]

▼ Python Basic

Google Colaboratory

 <https://colab.research.google.com/drive/1q6erndsnZzuS3rDdKWRgRaRXrzelv6Q5?usp=sharing>



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)

```

```

↳ 1
   ['b', 'c']
   b
   [9, 2, 3]
   ['a', 'b', 'c']
   ['a', 'b']
   []
   [0]

```

```

# 리스트에서 많이 사용되는 함수 정리

# append
# 마지막에 요소 추가.

a = []

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

# insert
# 요소 추가, 특정 위치를 지정하여 마음대로 요소 추가.

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

# 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 발생.
print(a.index(3))

```

```

☞ [0]
   [0, 'a']
   [4, 1, 2, 3]
   [1, 2, 3, 4]
   [4, 3, 2, 1]
   1

```

```

# 삭제 remove
# 중복되는 수는 처음 등장하는 하나만 삭제.
print()

a = [1, 2, 3, 4, 5]
a.remove(3)
print(a)

# 삭제 pop
# 리스트의 마지막 요소를 삭제.

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)

# 갯수 세기 count
# 특정 요소의 갯수 세기

a = [1, 2, 3, 3]
print(a.count(3))

```



```

[1, 2, 4, 5]
[1, 2]
[2, 3]
2

```

Tuple

```

# TUPLE
# () 소괄호로 만든다.
# immutable. 수정이 불가능하다.
# indexing & slicing 이 가능하다.
# 한 개의 요소로 tuple을 만들 때는 (1,) 와 같이 쉼표를 넣는다.

a = ()
b = (1,) # 한 개의 요소로 tuple을 만들 때는 (1,) 와 같이 쉼표를 넣는다.
c = (1, 2, 3)
d = (1, 2, 'a', 'b')

# del d[0]           # immutable. 삭제이 불가능하다.
# del d[0] = 9       # immutable. 수정이 불가능하다.
# del 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

a = {'이름' : '홍길동', '나이' : 20, '보유코인' : ['비트코인', '에이다', '이더리움']}

# 딕셔너리에서 key 값들만 추출
print(a.keys())

# 딕셔너리에서 value 값들만 추출
print(a.values())

# 쌍으로 추출(튜플 쌍으로 추출)
print(a.items())

# key 값으로 value 값 추출
print(a.get('이름'))
print(a.get('몸무게')) # 찾는 key 값이 없을 때는 None을 출력한다.

# key 값으로 value 값 추출 (인덱싱이 불가능하기 때문에 꺼낼 때 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 = {} # Dictionary 와 Set은 동일하게 {}를 사용한다. 따라서 빈 값으로 생성할 때는 주의하자.
a = set()
b = {1, 2, 3}
c = {2, 'a', 1, '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}
{1, 2, 'a', 'b'}
{1, 'a', 'b'}

```

| 자료형 | 설명 | 함수 |
|-------|--|--|
| List | # [] 대괄호로 만든다. # indexing & slicing 이 가능하다. # ordered. 순서가 있다. # mutable. 수정 가능하다. | # 요소 추가 append [마지막에 요소 추가] a.append(0) # 요소 추가 insert [특정 위치를 지정하여 마음대로 요소 추가] d.insert(0,4) # 정렬 sort a.sort() # 뒤집기 reverse a.reverse() # 인덱스 index a.index(3) # 삭제 remove [중복되는 수는 처음 등장하는 하나만 삭제] a.remove(3) # 삭제 pop [리스트의 마지막 요소를 삭제] a.pop() # 괄호에 값이 없으면 마지막 요소를 하나 삭제. a.pop(0) # 괄호에 값이 있으면 그 요소를 삭제. # 갯수 세기 count [특정 요소의 갯수 세기] a = [1, 2, 3, 3] print(a.count(3)) |
| Tuple | # () 소괄호로 만든다. # immutable. 삽입, 수정, 삭제가 불가능하다. # indexing & slicing 이 가능하다. # ordered. 순서가 있다. # 한 개의 요소로 tuple을 만들 때는 (1,) 와 같이 쉼표를 넣는다. | # len() [튜플 요소의 개수를 반환하는 함수] d = (1, 2, 'a', 'b') print(len(d)) |
| Dict | # {} 중괄호로 만든다. # mutable. 삽입, 수정, 삭제가 가능하다. # indexing & slicing 이 불가능하다. # unordered. 순서가 없다. # key 값과 value 값으로 데이터가 저장된다. # 세상의 많은 데이터가 key 값 : value 값 형태로 표현된다. ex. 이름 : 홍길동, 나이 : 20 # 파이썬에서 이러한 대응 관계를 데이터로 저장하는 방식을 딕셔너리라고 한다. # 마치 JS 의 JSON 과 상당히 유사한 공통점을 가지고 있다. | # 딕셔너리에서 key 값들만 추출 print(a.keys()) # 딕셔너리에서 value 값들만 추출 print(a.values()) # 쌍으로 추출(튜플 쌍으로 추출) print(a.items()) # key 값으로 value 값 추출 print(a.get('이름')) print(a.get('몸무게')) # 찾는 key 값이 없을 때는 None을 출력한다. # key 값으로 value 값 추출 (인덱싱이 불가능하기 때문에 꺼낼 때 많이 사용된다.) print(a['이름']) # print(a['몸무게']) # 찾는 키값이 없을 때 예외 발생된다. # 키가 딕셔너리안에 있는지 확인 print('이름' in a) # 찾는 키값이 있을때 true 반환 print('몸무게' in a) # 찾는 키값이 없을때 false 반환 |
| Set | # {} 중괄호로 만든다. # mutable. 수정 가능하다. # indexing & slicing 이 불가능하다. # unordered. 순서가 없다. # 중복데이터를 허용하지 않는다. | |

Boolean

▶ # 불 Bool 자료형 boolean

반드시 대문자로 시작한다.

```
a = True
b = False
c = "true"

print(type(a))
print(type(c))

print(1 == 1)
print(1 != 1)
```

↳ <class 'bool'>
<class 'str'>
True
False

input [키보드로 입력 받기]

키보드로 입력 받기

```
a = input()
print(a)
```

↳ Hi
Hi

```
a = input("지금이 오후인가요?")
print(a)
```

↳ 지금이 오후인가요?네
네


```
# 파이썬에서 input 은 항상 먼저 문자 취급한다.

# a = input("지금이 몇시예요?")
# print(a + 1)

a = int(input("지금부터 한 시간 뒤는 몇시 인가요?"))
print(a + 1)
```

📄 지금부터 한 시간 뒤는 몇시 인가요?1
2

▼ Python Practice

069

Create a tuple containing the names of five countries and display the whole tuple. Ask the user to enter one of the countries that have been shown to them and then display the index number (i.e. position in the list) of that item in the tuple.

```
country = ('USA', 'Korea', 'China', 'Spain', 'Japan')
print(country)
print()
a = input("나라 입력 : ")
print(country.index(a))
```

069

```
country_tuple = ("France", "England", "Spain", "Germany", "Australia")
print(country_tuple)
print()
country = input("Please enter one of the countries from above: ")
print(country, "has index number", country_tuple.index(country))
```

071

Create a list of two sports. Ask the user what their favourite sport is and add this to the end of the list. Sort the list and display it.

```
sports = ['tennis', 'football']
sports.append(input('가장 좋아하는 스포츠가 무엇인가요?\n'))
sports.sort()
print(sports)
```

071

```
sports_list = ["tennis", "football"]
sports_list.append(input("What is your favourite sport? "))
sports_list.sort()
print(sports_list)
```

072

Create a list of six school subjects. Ask the user which of these subjects they don't like. Delete the subject they have chosen from the list before you display the list again.

```
subjects = ['국어', '수학', '영어', '미술', '음악', '체육']
print(subjects)
a = input('싫어하는 과목?')
subjects.remove(a)
print(subjects)
```

072

```
subject_list = ["maths", "english", "computing", "history", "science", "spanish"]
print(subject_list)
dislike = input("Which of these subjects do you dislike? ")
getrid = subject_list.index(dislike)
del subject_list[getrid]
print(subject_list)
```

073

Ask the user to enter four of their favourite foods and store them in a dictionary so that they are indexed with numbers starting from 1. Display the dictionary in full, showing the index number and the item. Ask them which they want to get rid of and remove it from the list. Sort the remaining data and display the dictionary.

```
food_dictionary = {}
food1 = input("첫번째로 좋아하는 음식은? ")
food_dictionary[1] = food1
food2 = input("두번째로 좋아하는 음식은? ")
food_dictionary[2] = food2
food3 = input("세번째로 좋아하는 음식은? ")
food_dictionary[3] = food3
food4 = input("네번째로 좋아하는 음식은? ")
food_dictionary[4] = food4
print(food_dictionary)
dislike = int(input("제거할 음식은? "))
del food_dictionary[dislike]
print(sorted(food_dictionary.values()))
```

073

```
food_dictionary = {}
food1 = input("Enter a food you like: ")
food_dictionary[1] = food1
food2 = input("Enter another food you like: ")
food_dictionary[2] = food2
food3 = input("Enter a third food you like: ")
food_dictionary[3] = food3
food4 = input("Enter one last food you like: ")
food_dictionary[4] = food4
print(food_dictionary)
dislike = int(input("Which of these do you want to get rid of? "))
del food_dictionary[dislike]
print(sorted(food_dictionary.values()))
```

리스트에서 슬라이싱하기 문제

074

Enter a list of ten colours.
Ask the user for a starting
number between 0 and 4
and an end number
between 5 and 9. Display
the list for those colours
between the start and end
numbers the user input.

```
colours = ['red', 'blue', 'green', 'orange', 'yellow', 'purple', 'black', 'white', 'navy', 'gray']
start = int(input('시작 숫자 입력 (0~4) : '))
end = int(input('끝 숫자 입력 (5~9) : '))
print(colours[start : end])
```

074

```
colours = ["red", "blue", "green", "black", "white", "pink", "grey", "purple", "yellow", "brown"]
start = int(input("Enter a starting number (0~4): "))
end = int(input("Enter an end number (5~9): "))
print(colours[start:end])
```

▼ Cafe24 호스팅 Setting

카페24 호스팅 | 온라인 비즈니스의 시작

Tomcat JSP, node.js 기반의 호스팅 서비스는 다양한 개발 니즈를 만족하는 서비스를 구축할 수 있습니다. 웹, 트래픽 용량은 웹/스트리밍/이미지CDN 합산 용량입니다. Tomcat JSP 호스팅의 장점 단독 Tomcat 서버 제공 전 사양에 단독 Tomcat을 제공하여 서버릿 재 https://hosting.cafe24.com/?controller=new_product_page&page=language

cafe24
호스팅

1. 서버 환경설정

☛ 서버 환경설정

- ☐ Tomcat 10.0.x / JSP 3.0 / Servlet 5.0 / JDK 17 / MariaDB 10.1.x UTF-8
- ☐ Tomcat 10.0.x / JSP 3.0 / Servlet 5.0 / JDK 11 / MariaDB 10.1.x UTF-8
- ☒ Tomcat 8.5.x / JSP 2.3 / Servlet 3.0 / JDK 8 / MariaDB 10.1.x UTF-8

2. MySQL 외부 IP 접근설정

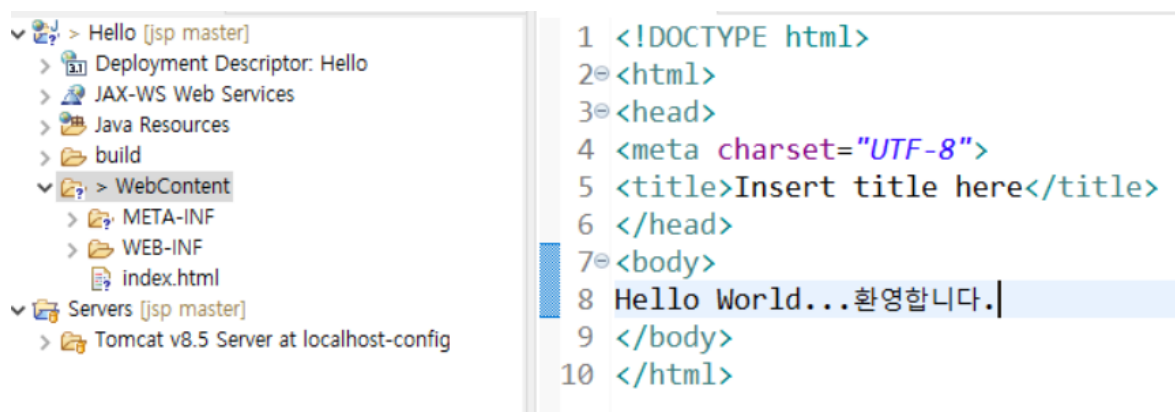
• MySQL 외부 IP 접근설정

| | |
|--------|-------------------------|
| 사용유무 | 사용중 ➤ 설정하기 |
| 등록된 IP | 111.118.52.74 |

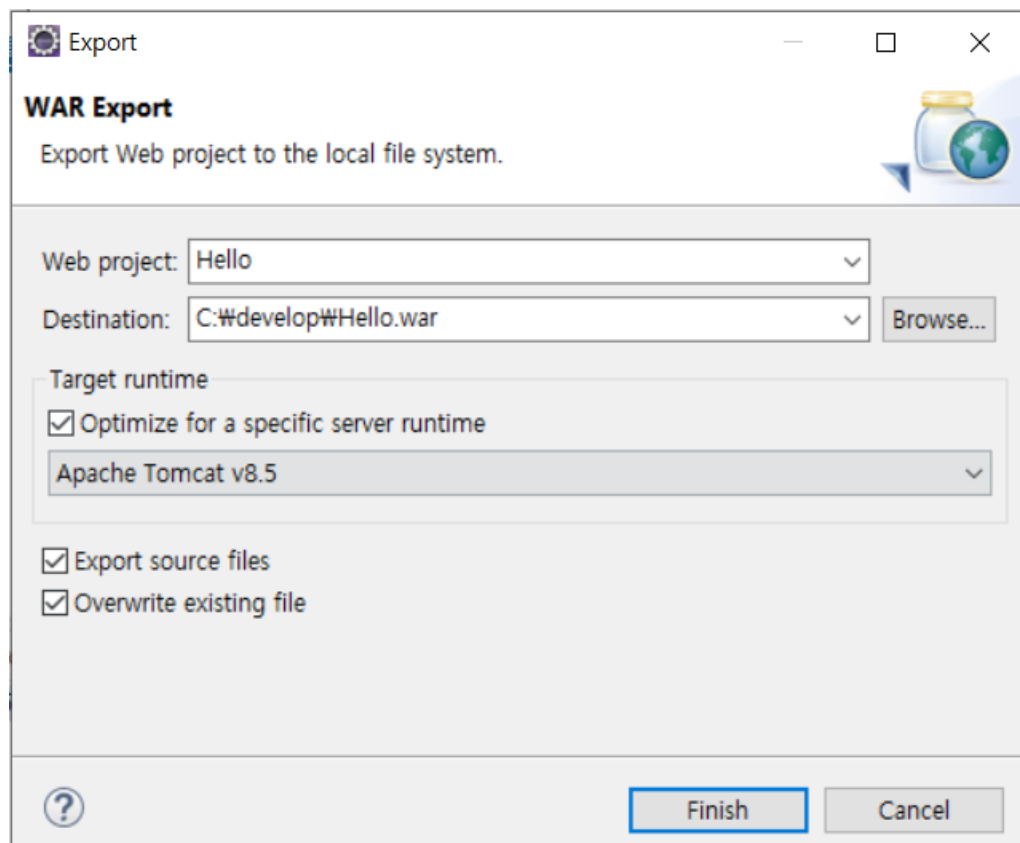
• FTP 접속 설정

| | |
|-------------------------|---------------------------------|
| 접속 설정 | 국내에서만 접속 허용 ➤ 설정하기 |
| 접속 차단 해제 ? | ➤ FTP 접속 차단 해제 |

FileZilla TEST



기본 프로젝트 작성



전부 체크 후 war로 내보내기

FileZilla

Advertisement: Welcome to the homepage of FileZilla®, the free FTP solution. The FileZilla Client not only supports FTP, but also FTP over TLS (FTPS) and SFTP. It is open source software distributed free of charge under the terms of the GNU General Public License.

<https://filezilla-project.org/>

Cafe24로 FTP 전송 해주는 프로그램 Download

[←](#)
[→](#)
[↻](#)
[🏠](#)
filezilla-project.org
[🔗](#)
[★](#)
[🌐](#)
[🔍](#)
[🔄](#)
[📶](#)
[📄](#)
[🐱](#)

[앱](#)
[M](#)
[A](#)
[D](#)
[N](#)
[S](#)
[K](#)
[Y](#)
[Shop](#)
[TV](#)
[Music](#)
[Cafe](#)
[Study](#)
[Develop](#)



FileZilla

The free FTP solution

[Home](#)

FileZilla
[Features](#)
[Screenshots](#)
[Download](#)
[Documentation](#)
[FileZilla Pro](#)

FileZilla Server
[Download](#)

Community
[Forum](#)
[Wiki](#)

General
[FAQ](#)
[Support](#)
[Contact](#)
[License](#)
[Privacy Policy](#)
[Trademark Policy](#)

Development
[Source code](#)
[Nightly builds](#)
[Translations](#)
[Version history](#)
[Changelog](#)
[Issue tracker](#)

Other projects

Promotion:



FileZilla® Pro

The Best FTP Solution



[GET IT](#)

Overview

Welcome to the homepage of FileZilla®, the free FTP solution. The *FileZilla Client* not only supports FTP but also FTP over TLS (FTPS) and SFTP. It is open source software distributed free of charge under the terms of the GNU General Public License.

We are also offering *FileZilla Pro*, with additional protocol support for WebDAV, Amazon S3, B2, Dropbox, Microsoft OneDrive, Google Drive, Microsoft Azure Blob and File Storage Service.

Last but not least, *FileZilla Server* is a free open source FTP and FTPS Server.

Support is available through our [forums](#), the [wiki](#) and the [bug and feature request tracker](#).

In addition, you will find documentation on how to compile FileZilla and nightly builds for various platforms in the development section.

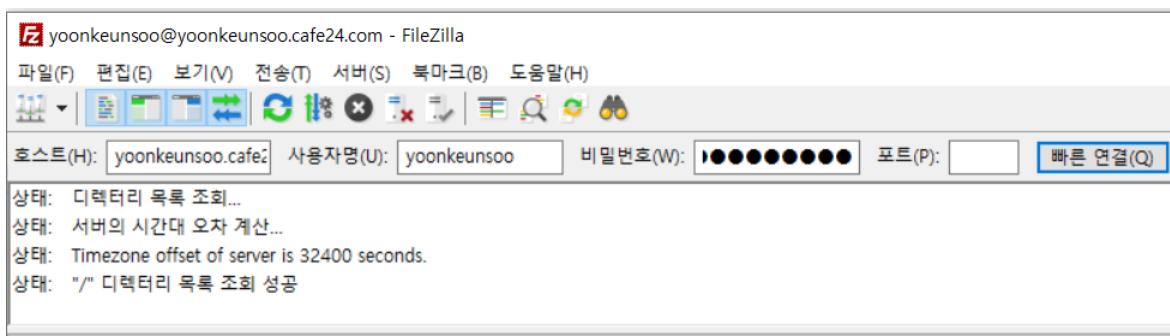
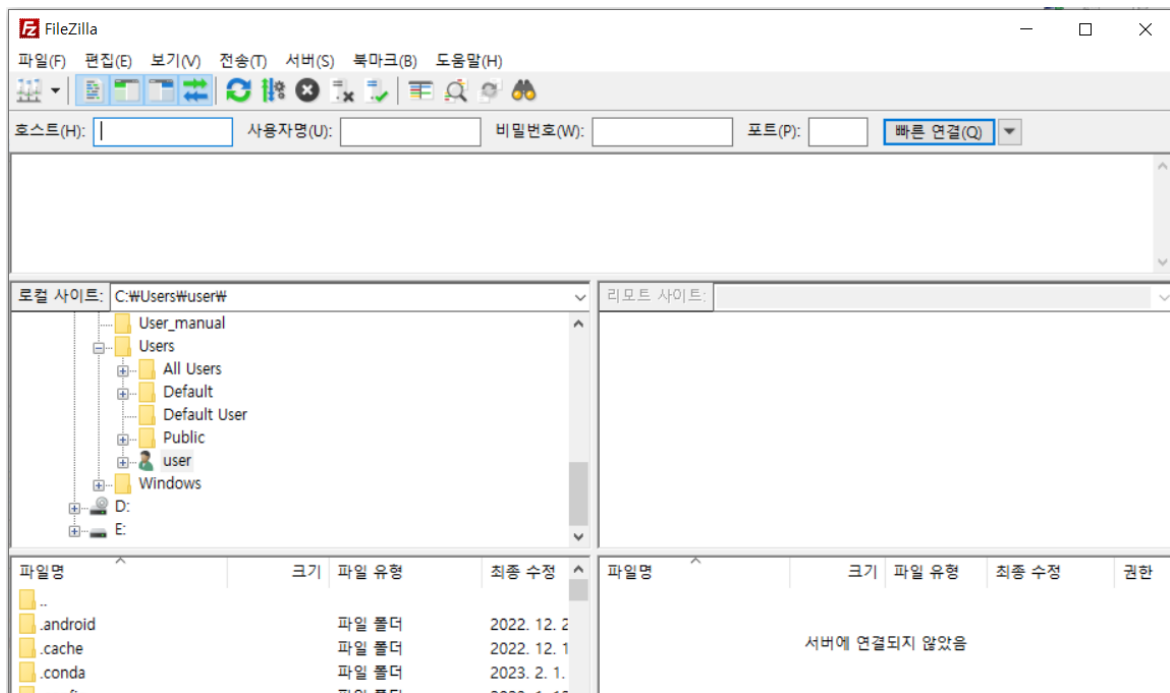
Quick download links

[Download FileZilla Client](#)

All platforms

[Download FileZilla Server](#)

All platforms



호스트 칸 : 아이디.cafe24.com

사용자명 칸 : 아이디

비밀번호 : 비번 <— FTP 비번

포트 : 빈칸

FileZilla interface showing local and remote site connections.

Local site: C:\develop\#

Remote site: /

Files in local site:

| 파일명 | 크기 | 파일 유형 | 최종 수정 |
|-----------|----|-------|-----------------|
| .. | | | |
| .git | | 파일 폴더 | 2023-01-18 오... |
| .metadata | | 파일 폴더 | 2023-01-17 오... |
| cmdr | | 파일 폴더 | 2023-02-03 오... |
| eclipse | | 파일 폴더 | 2023-02-03 오... |
| JSP | | 파일 폴더 | 2023-02-03 오... |
| Notebook | | 파일 폴더 | 2023-01-31 오... |
| Server | | 파일 폴더 | 2023-01-14 오... |
| Servlet | | 파일 폴더 | 2023-01-18 오... |
| tomcat8 | | 파일 폴더 | 2023-01-14 오... |
| Workspace | | 파일 폴더 | 2023-01-18 오... |

10 디렉터리

Files in remote site:

| 파일명 | 크기 | 파일 유형 | 최종 수정 | 권한 | 소유자/... |
|---------------|--------|------------|---------------|------------|--------------|
| .. | | | | | |
| tomcat | | 파일 폴더 | 2022-11-11... | drwxr-s--- | difbfl475... |
| .bash_logout | 18 | Bash Lo... | 2016-08-03 | -rwxr-x--- | difbfl475... |
| .bash_profile | 688 | Bash Pr... | 2023-02-02... | -rw----- | difbfl475... |
| .bashrc | 231 | Bash RC... | 2016-08-03 | -rwxr-x--- | difbfl475... |
| .ftppass | 38,106 | FTPACC... | 2019-08-01 | -rw-r--r-- | difbfl475... |
| .ftppass2 | 53,166 | FTPACC... | 2019-08-01 | -rw-r--r-- | difbfl475... |

5 파일 및 1 디렉터리. 총 크기: 92,209 바이트

서버/로컬 파일 | 방향 | 리모트 파일 | 크기 | 우선 ... | 상태



한번 접속하면 두번째부터는 빠른 연결로 입력없이 간편하게 접속 가능.

생성한 war 파일의 이름을 ROOT.war 로 변경하고 업로드

FileZilla interface showing the upload of ROOT.war to the remote site.

Local site: C:\develop\#

Remote site: /tomcat/webapps

Files in local site:

| 파일명 | 크기 | 파일 유형 | 최종 수정 |
|----------|-----|----------------|-------------|
| notebook | | 파일 폴더 | 2023. 1. 12 |
| servlet | | 파일 폴더 | 2023. 1. 12 |
| tomcat8 | | 파일 폴더 | 2023. 1. 12 |
| ROOT.war | 895 | ALZip WAR File | 2023. 2. 2 |

Files in remote site:

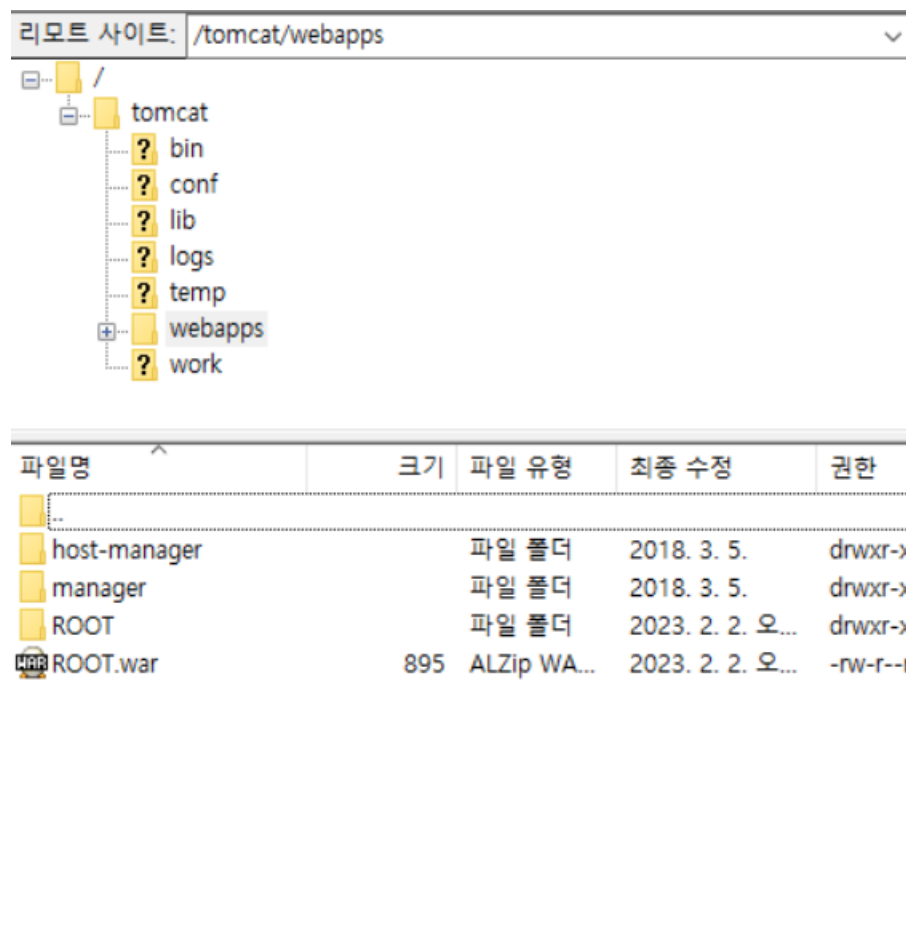
| 파일명 | 크기 | 파일 유형 | 최종 수정 | 권한 |
|--------------|-----|-------------|------------------|--------|
| .. | | | | |
| host-manager | | 파일 폴더 | 2018. 3. 5. | drwxr- |
| manager | | 파일 폴더 | 2018. 3. 5. | drwxr- |
| ROOT.war | 895 | ALZip WA... | 2023. 2. 2. 오... | -rw-r- |

위에서 생성한 war파일의 이름을 ROOT.war 로 변경하여

오른쪽. 즉, Cafe24 로 보낸다. (더블클릭 또는 Drag & Drop)

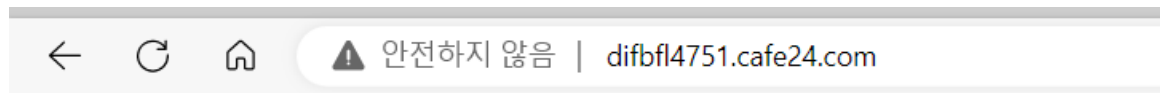
그리고 파일질라를 완전히 종료 후 다시 들어가면 ROOT 폴더가 생성되어 있다.

설정 완료



주소에 https에서 s 를 삭제한다.

<http://difbf14751.cafe24.com/>

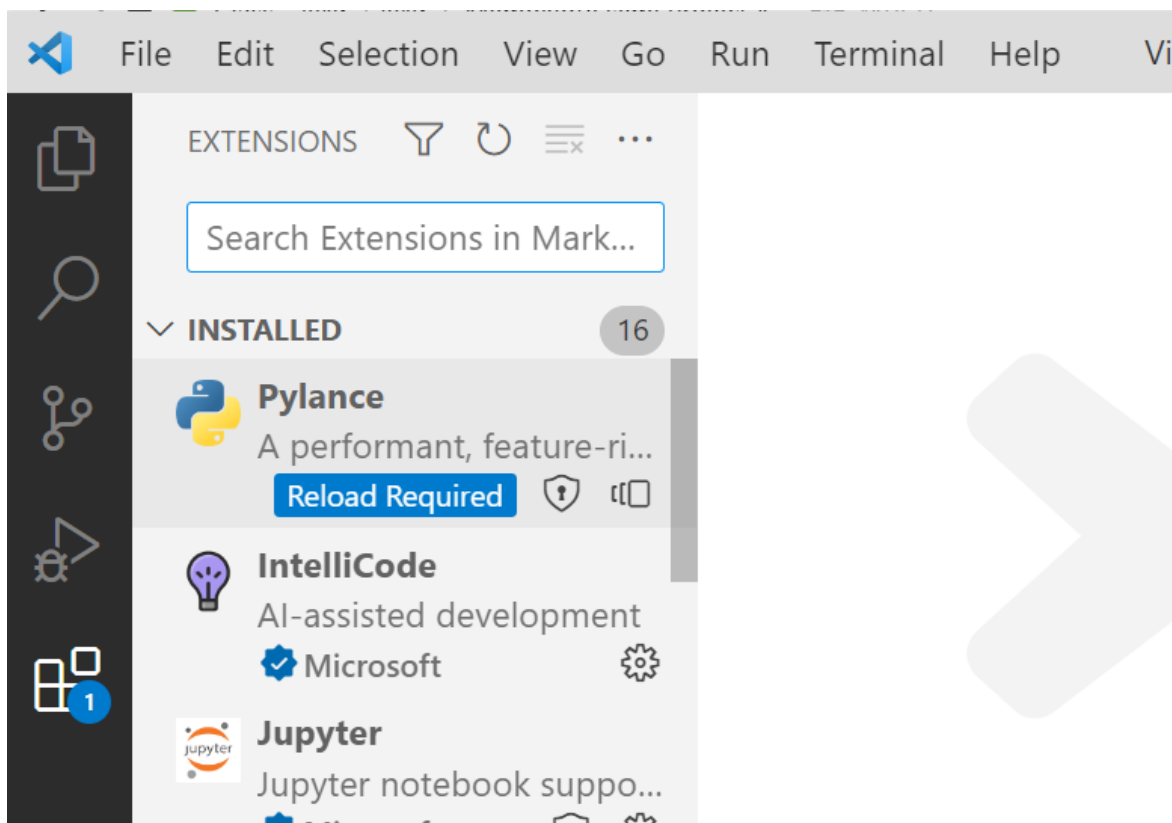


Hello~ world! 환영합니다!

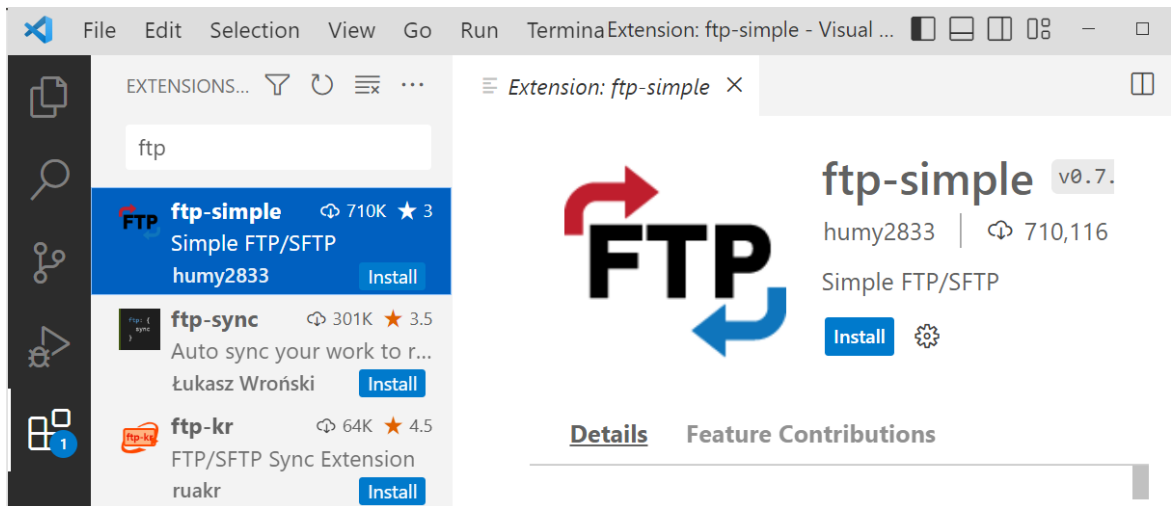
사이트의 일부 내용등을 조금 수정할때도 새로운 war 를 만들어서 다시 파일질라로 보내기는 부담스럽다.

이렇게 조금 변경을 할때는 VS Code 를 사용하면 아주 편하다.

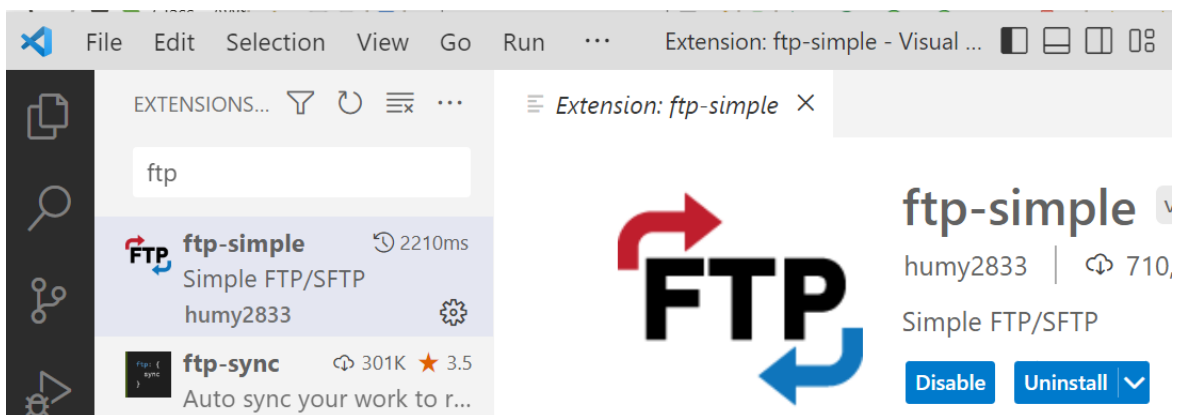
왼쪽하단 Extension



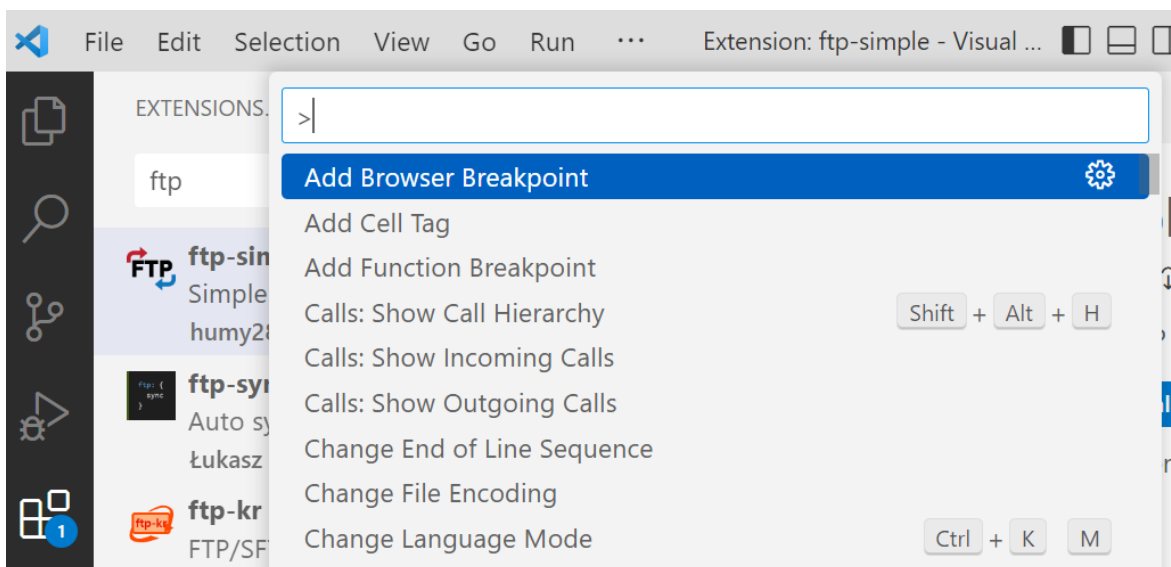
ftp 로 검색하여 ftp-simple 설치



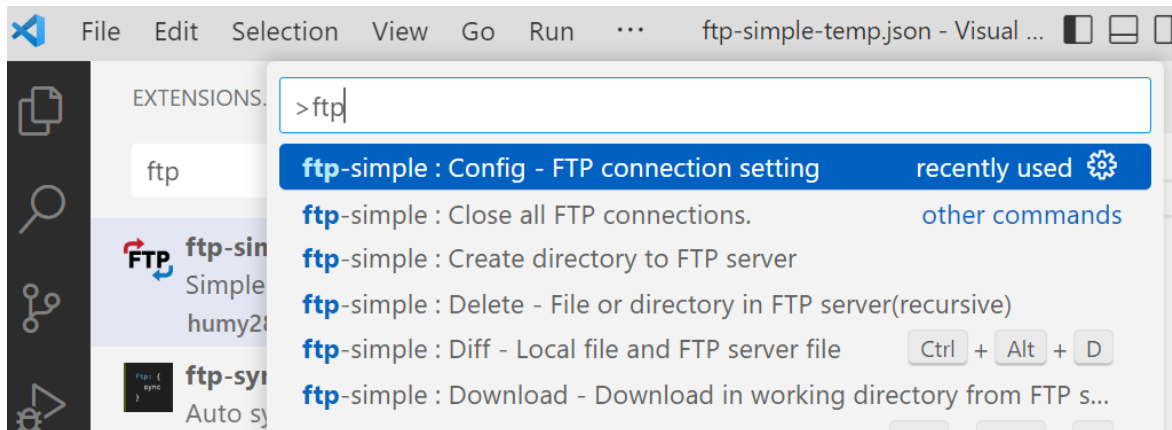
설치완료



설정 들어가서 F1 - 검색 : ftp



ftp-simple : Config - FTP connection setting

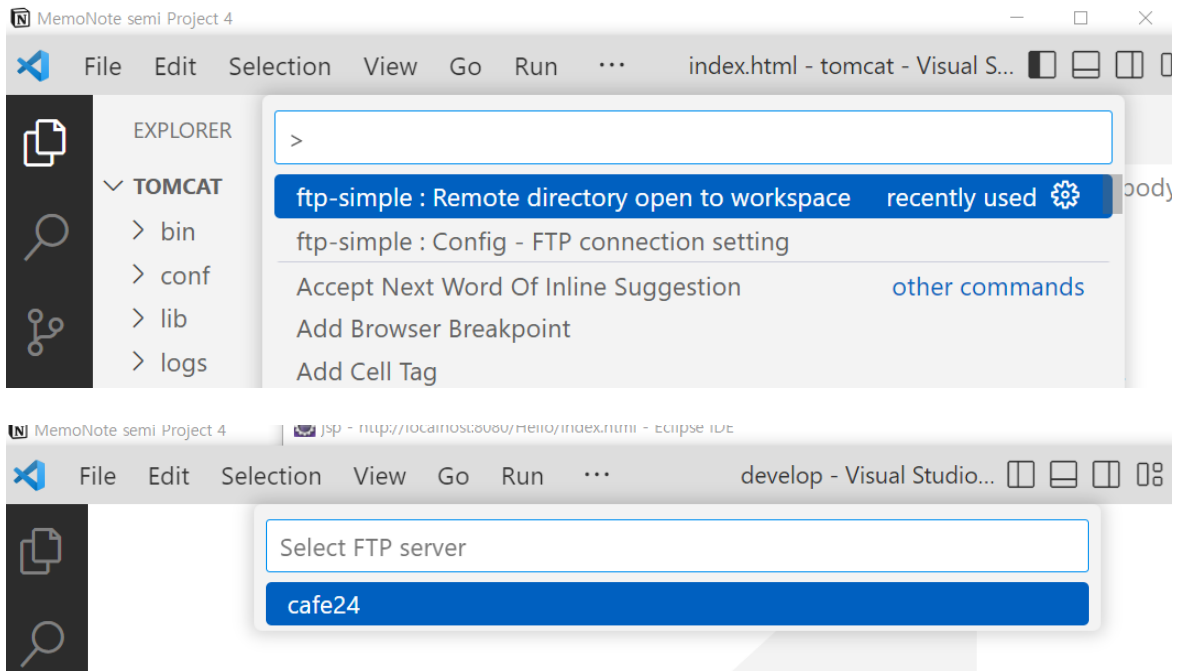


설정은 한번만 하면 앞으로는 계속 사용만 한다.

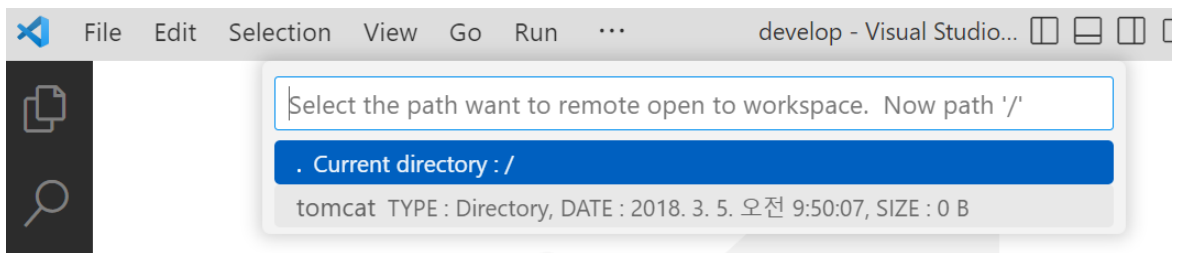
```
{ } ftp-simple-temp.json
C: > Users > difbf > AppData > Roaming > Code > User > globalStorage >
1  [
2      {
3          "name": "cafe24",
4          "host": "difbfl4751.cafe24.com",
5          "port": 21,
6          "type": "ftp",
7          "username": "difbfl4751",
8          "password": "비밀번호",
9          "path": "/",
10         "autosave": true,
11         "confirm": false
12     }
13 ]
```

F1 키를 누르고 ftp 를 검색 후 목록에서 Remote directory open to workspace 를 선택한다.

다음 부터는 F1키만 누르면 항상 상단에서 고를 수 있다.



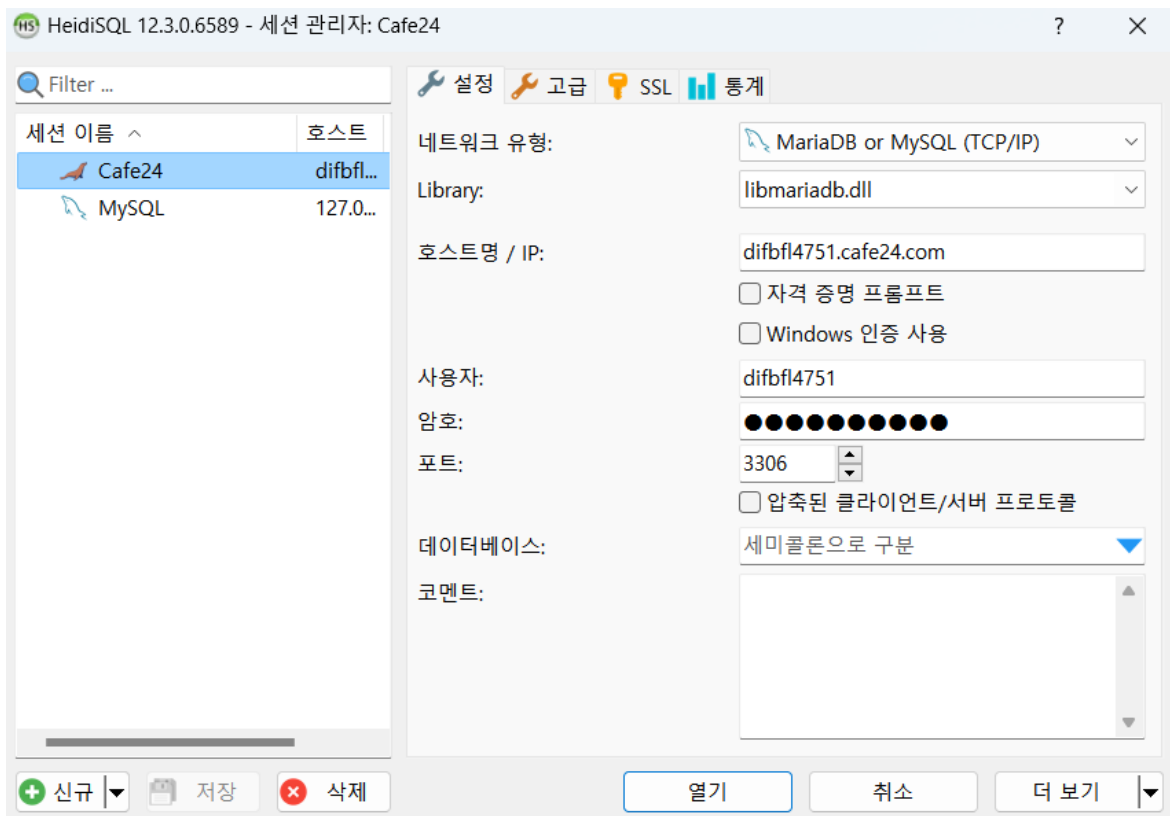
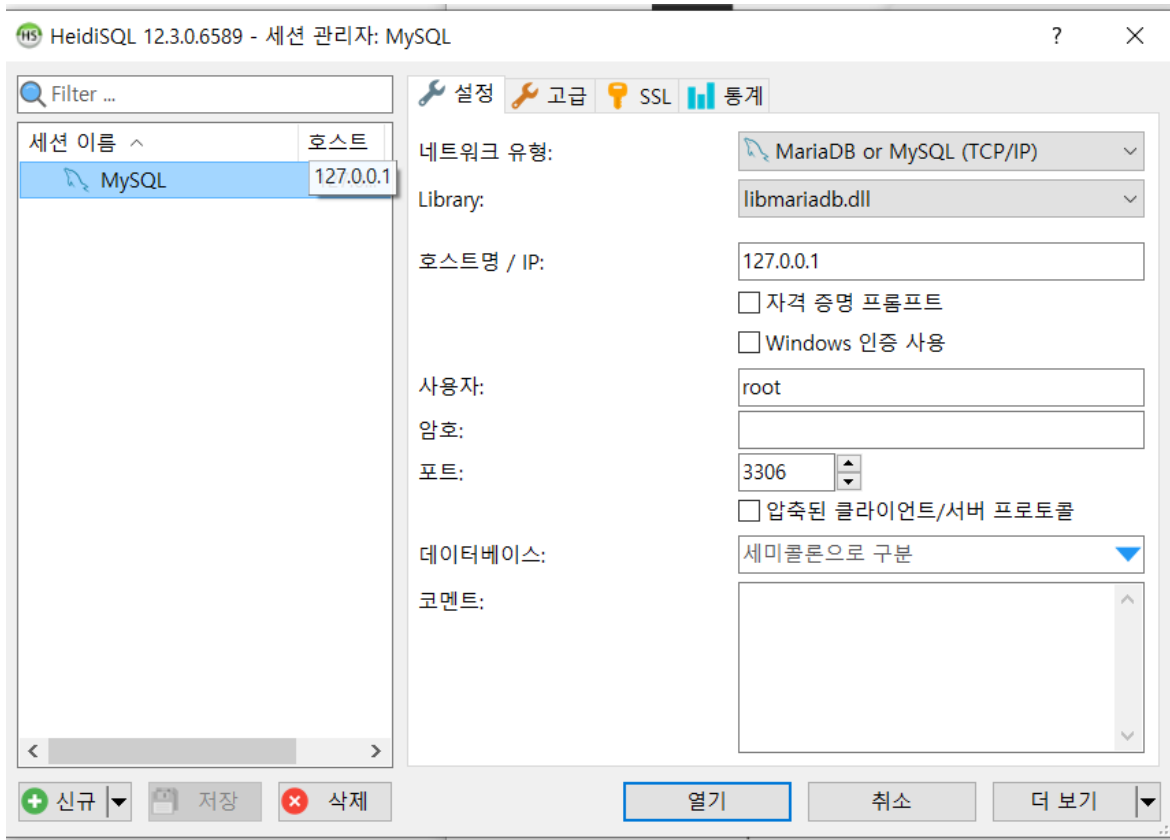
아래에서 tomcat 클릭



Heidi 설정



Heidi 설정시 중요한 것은 Cafe24 사이트 내에서 DATABASE IP설정이 제대로 되어 있어야 한다는 것이다.



```
CREATE TABLE `user` (
  `id` VARCHAR(128) NOT NULL COLLATE 'utf8mb4_0900_ai_ci',
  `password` VARCHAR(32) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',
  `name` VARCHAR(32) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',
  `ts` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`) USING BTREE
)
COLLATE='utf8mb4_0900_ai_ci'
ENGINE=InnoDB
;
```

```
CREATE TABLE `feed` (
  `no` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  `id` VARCHAR(128) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',
  `content` LONGTEXT NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',
  `ts` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`no`) USING BTREE
)
COLLATE='utf8mb4_0900_ai_ci'
ENGINE=InnoDB
;
```

The screenshot shows the Cafe24 database management interface. On the left, a tree view shows the database structure for 'Cafe24', including a database 'difbf14751' and tables 'feed', 'student', 'user', and 'information_schema'. The 'feed' table is highlighted with a size of 16.0 KiB. In the center, a table list shows the 'difbf14751' database with 3 tables and a size of 48.0 KiB. On the right, a table with columns like '이름', '조합', '서버 기본값', '함수', '프로...', '트리거', '이벤트', and '기본 조합' is visible. Overlaid on the interface are two dialog boxes. The top one is '데이터베이스 생성...' (Create Database...) with fields for '이름' (Name) set to 'test', '조합' (Collation) set to 'armscii8_bin', and '서버 기본값' (Server default) set to 'utf8_general_ci'. It has '확인' (OK) and '취소' (Cancel) buttons. The bottom one is 'Cafe24: 오류' (Cafe24: Error) with a red 'X' icon. It contains the text: '"test" 데이터베이스 생성에 실패하였습니다.' (Failed to create database 'test'). Below this, it says: 'SQL 오류 (1044): Access denied for user 'difbf14751'@'49.169.10.234' to database 'test''. There is an '확인' (OK) button and a link at the bottom: '해당 오류에 대한 몇 가지 도움말 찾기 (=> ecosia.org)' (Find some help for this error => ecosia.org).

테이블은 마음대로 만들었지만 Cafe24 정책으로 인해 새로운 DATABASE를 생성할 수 없다.

또한 자신의 ID 명으로만 DB를 사용할 수 있다.

따라서 지금까지 사용한 DB명은 사용할 수 없고 Cafe24 ID 명으로만 DB를 쓸 수 있다.

또한 지금까지는 root 사용자로 접근하였는데 역시 Cafe24 정책으로 인해 반드시 Cafe24 아이디와 비밀번호로만 접근할 수 있다.

DB 연결 TEST

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>

<!-- Step 1 import SQL Packages -->
<%@ page import="java.sql.*" %>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<%
//Step 2 load JDBC Driver
    try {
        Class.forName("com.mysql.jdbc.Driver");
    }catch(ClassNotFoundException err) {
        out.print("JDBC Driver loading error<br>" + err.getMessage());
    }

// Step 3 create Connection Object

    Connection conn = null;

    try {
        conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/yoonkeunsoo", "yoonkeunsoo", "guest0505");
    }catch(SQLException err) {
        out.print("Connection Object error<br>" + err.getMessage());
    }

// Step 4 create Statement Object

    String sql = "CREATE TABLE student("
        + "hakbun varchar(10),"
        + "name  varchar(10),"
        + "dept  varchar(20),"
        + "addr  varchar(30),"
        + "primary key(hakbun))";

    PreparedStatement pstmt = conn.prepareStatement(sql);

// Step 5 excute SQL Query

    pstmt.executeUpdate();

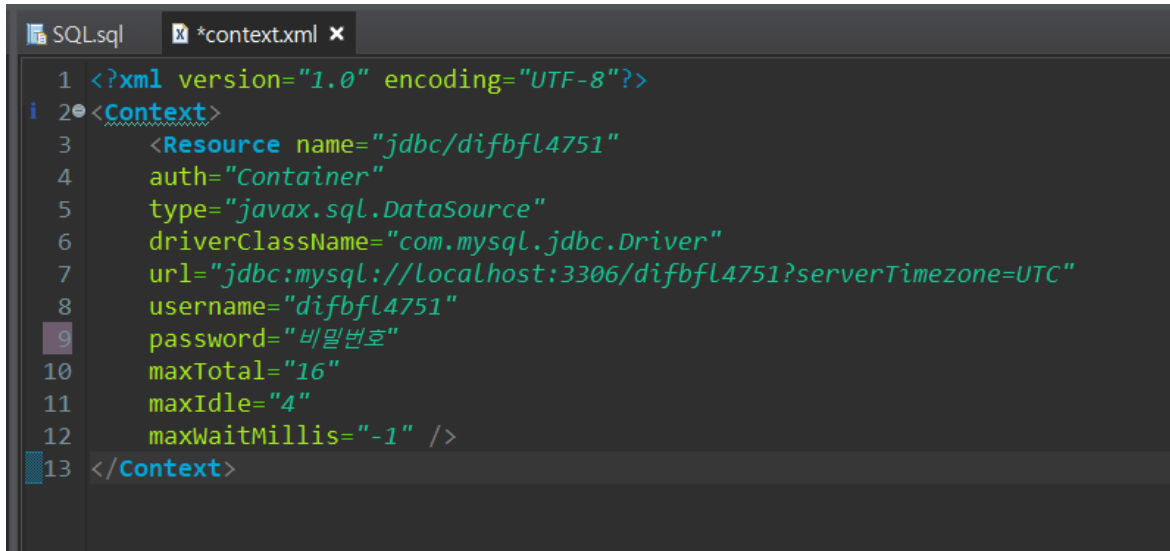
// Step 6 close Connection

    pstmt.close();
    conn.close();

%>
</body>
</html>
```

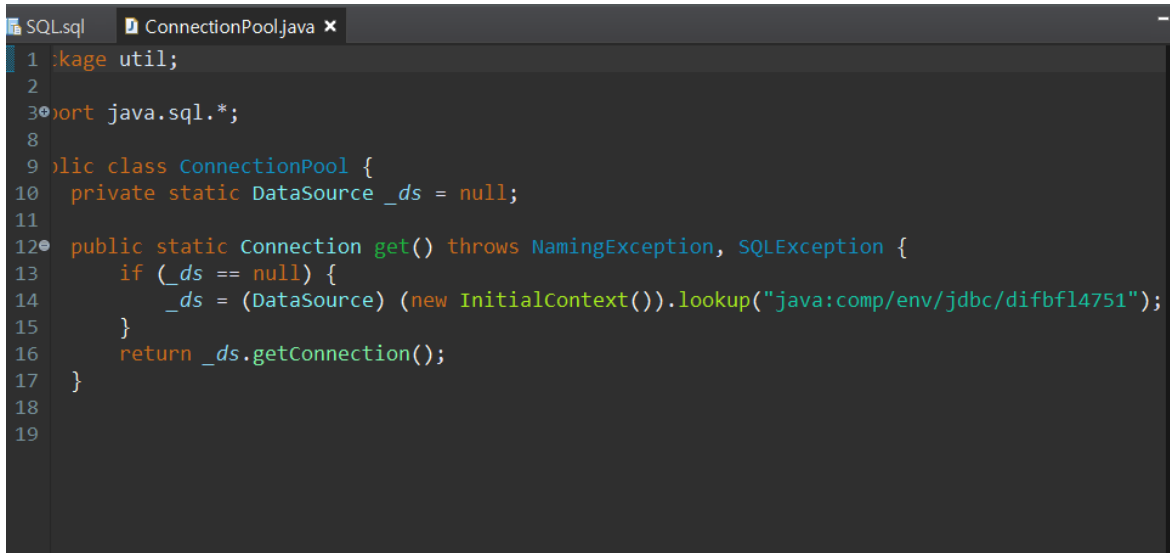
실제 프로젝트를 올리기 위해 사전 점검이 필요하다.

context.xml에서 DB 정보 수정.

A screenshot of an IDE window showing the content of a file named *context.xml. The code is XML and defines a JNDI resource for a MySQL database. The resource name is 'jdbc/difbfl4751', and it is of type 'javax.sql.DataSource'. The driver class is 'com.mysql.jdbc.Driver'. The URL is 'jdbc:mysql://localhost:3306/difbfl4751?serverTimezone=UTC'. The username is 'difbfl4751' and the password is masked as '비밀번호'. The pool size is set to 16, with a maximum idle of 4 and a maximum wait time of -1 milliseconds.

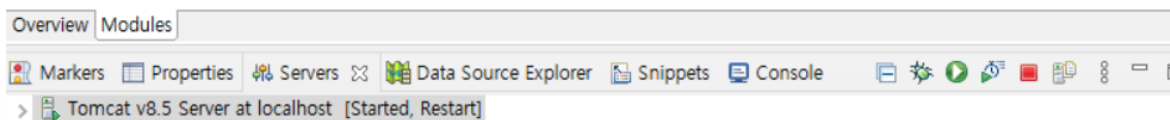
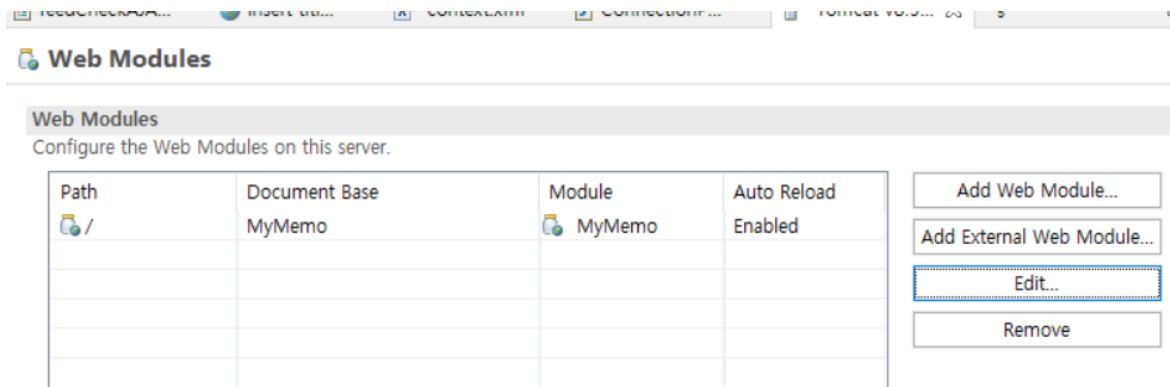
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Context>
3   <Resource name="jdbc/difbfl4751"
4     auth="Container"
5     type="javax.sql.DataSource"
6     driverClassName="com.mysql.jdbc.Driver"
7     url="jdbc:mysql://localhost:3306/difbfl4751?serverTimezone=UTC"
8     username="difbfl4751"
9     password="비밀번호"
10    maxTotal="16"
11    maxIdle="4"
12    maxWaitMillis="-1" />
13 </Context>
```

ConnectionPool에서 DB 정보 수정.

A screenshot of an IDE window showing the content of a file named ConnectionPool.java. The code is a Java class that provides a static method 'get()' to retrieve a DataSource. It uses JNDI to look up the resource 'java:comp/env/jdbc/difbfl4751' and returns a connection from the DataSource.

```
1 package util;
2
3 import java.sql.*;
4
5 public class ConnectionPool {
6   private static DataSource _ds = null;
7
8   public static Connection get() throws NamingException, SQLException {
9     if (_ds == null) {
10       _ds = (DataSource) (new InitialContext()).lookup("java:comp/env/jdbc/difbfl4751");
11     }
12     return _ds.getConnection();
13   }
14 }
15
16
17
18
19
```

루트 디렉토리 설정.



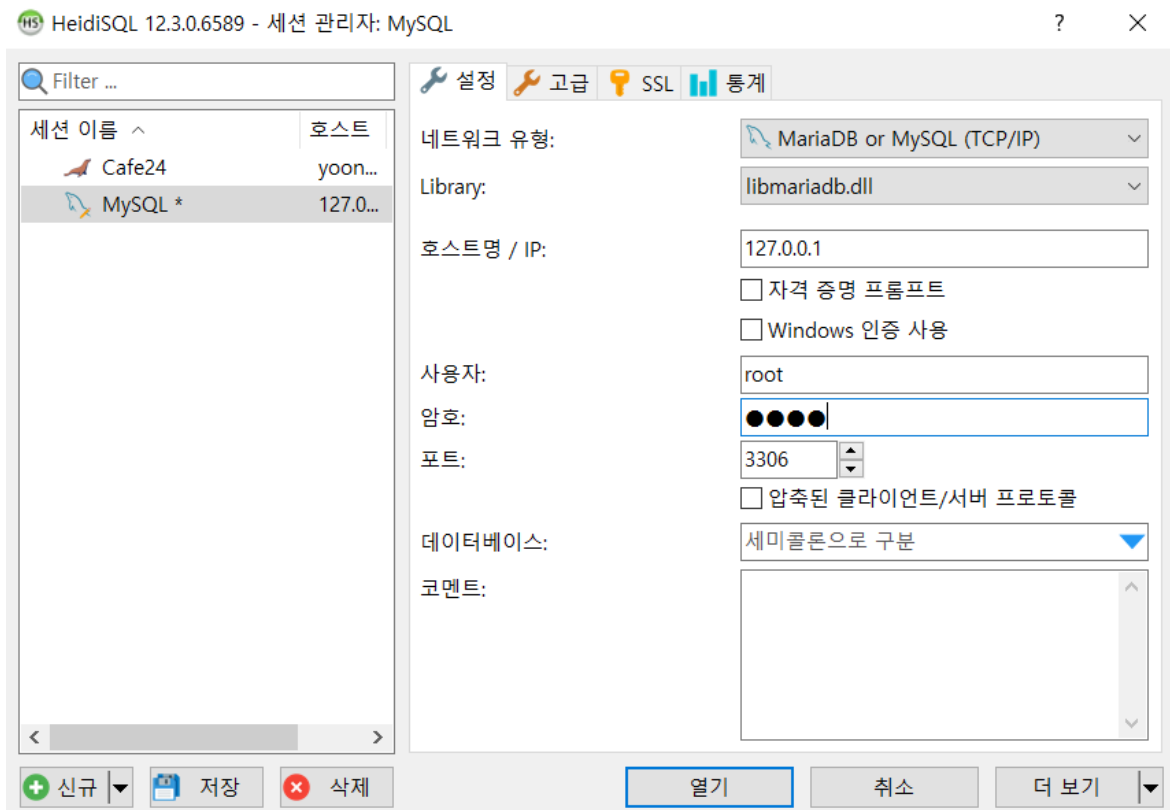
서버에 설치된 DB 이름은 직접 만들 수 없고 변경할 수 없으므로
Cafe24 에서 제공하는 DB 명(ID명과 동일)에 맞춘다.

마리아 DB드라이버는 자바 1.8 버전과 호환이 되는 **mariadb-java-client-1.8.0.jar**를 사용하여야 오류가 나지 않는다.(2.X 버전 이용시 자바 1.8과 호환 안됨)

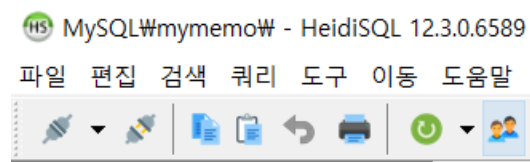
로컬 DB환경과 Cafe24 DB환경이 달라서 로컬에서 테스트가 힘든 상태이다.

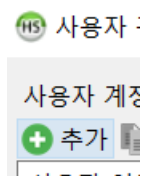
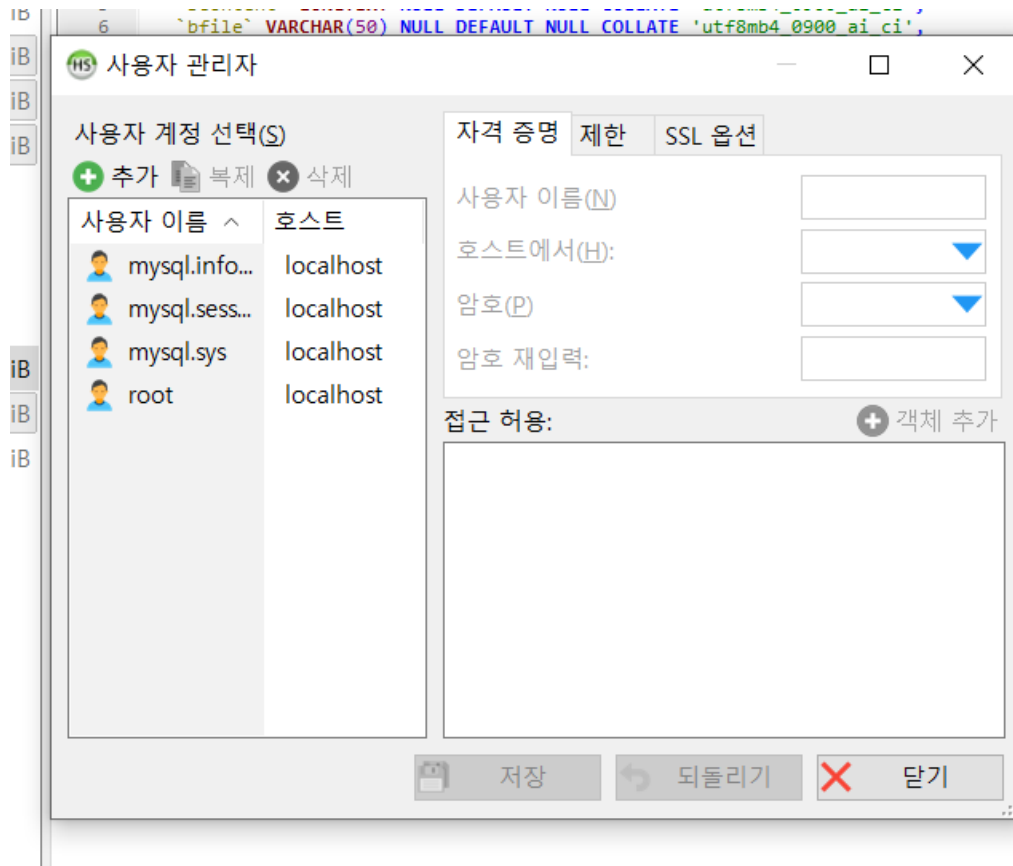
따라서 로컬에 Cafe24 와 동일하게 사용자 1명을 만들어 모든 권한을 주면 동일하게 로컬 작업을 하고 그대로 업로드하여 돌릴 수 있다.

먼저 로컬에 기존 아이디인 root로 들어간다.

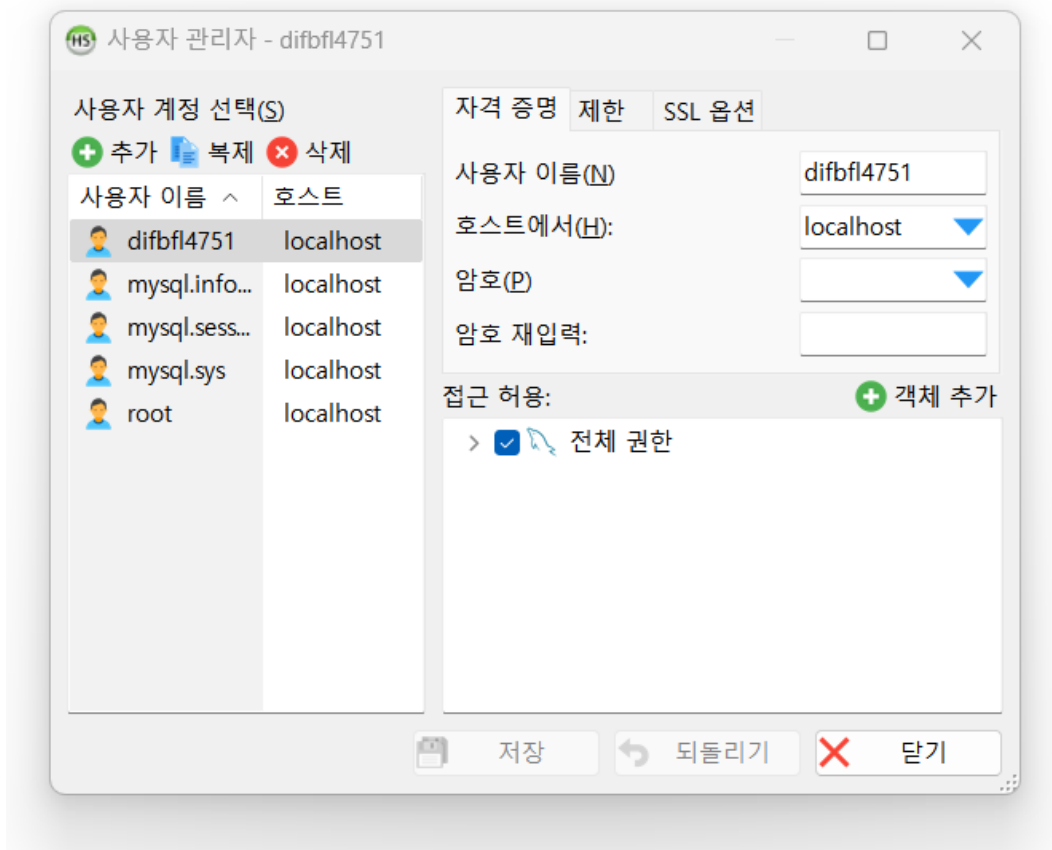


화명 상단의 사람 그림(사용자 인증 및 권한 관리)

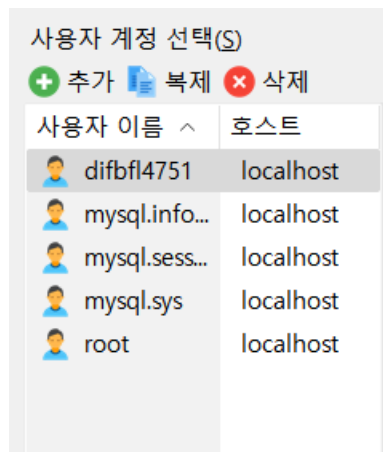




카페24와 동일한 아이디 동일한 패스워드를 넣고 전체 권한을 부여한다.



생성완료.



마지막으로 Cafe24의 DB명과 동일한 환경을 갖기 위해 이름을 변경한다.

