

MemoNote semi Project 3

✓ Python

✓ git 관련 명령어

✓ Font

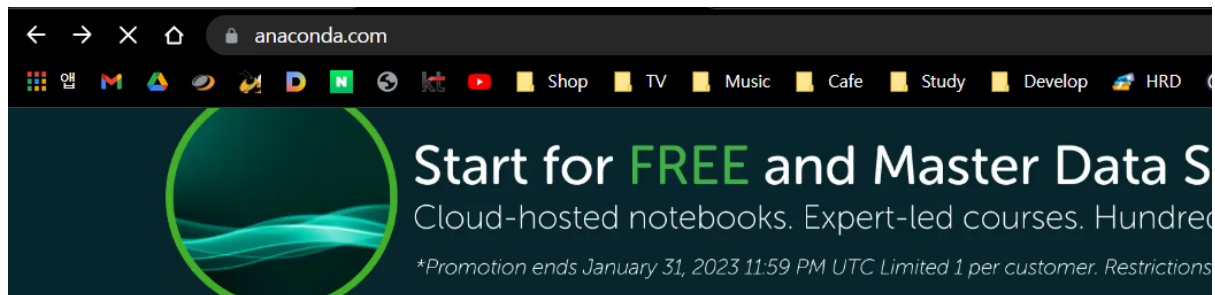
✓ Mail

✓ MyMEMO project 파일 정리

Python Basic

- about python - colab, anaconda, vs code

anaconda 다운로드



Data science techn a competitive e

Anaconda offers the easiest way to perform Python machine learning on a single machine. Start working with open-source packages and libraries to

Download 

For Windows

Python 3.9 • 64-Bit Graphical Installer • 621 MB

파이썬은 1990 귀도 반 로섬 이 개발한 인터프리터 언어(≠ 컴파일러 언어, 자바는 일반적으로는 컴파일러 언어로 구분하지만 엄밀히 따지면 인터프리터 언어의 특징도 가지고 있기 때문에 하이브리드 언어라는 표현도 사용 하기도 한다.) 이다.

파이썬 창시자
귀도 반 로섬
(1956년생)



귀도가 파이썬이라는 이름을 자신이 좋아하는 코미디 프로 '파이썬의 날라다니는 서커스'에서 따왔다고 한다. 파이썬의 사전적인 의미는 고대 신화에 나오는 파르나소스 산의 동굴에 살던 큰 뱀을 뜻한다.



파이썬은 컴퓨터 프로그래밍 교육을 위해 많이 사용되기도 하며 동시에 기업의 실무에서도 많이 사용되는 언어이다. 대표적으로 구글에서 만든 소프트웨어의 50% 이상이 파이썬으로 작성되었다고 한다. 예를 들면 인스타그램 드롭박스 등등이 파이썬으로 작성되었다.

파이썬의 특징

- 파이썬은 쉽다.
- 파이썬은 빠르다.
- 파이썬은 강력하다.
- 파이썬은 간결하다.
- 파이썬은 즐겁다.

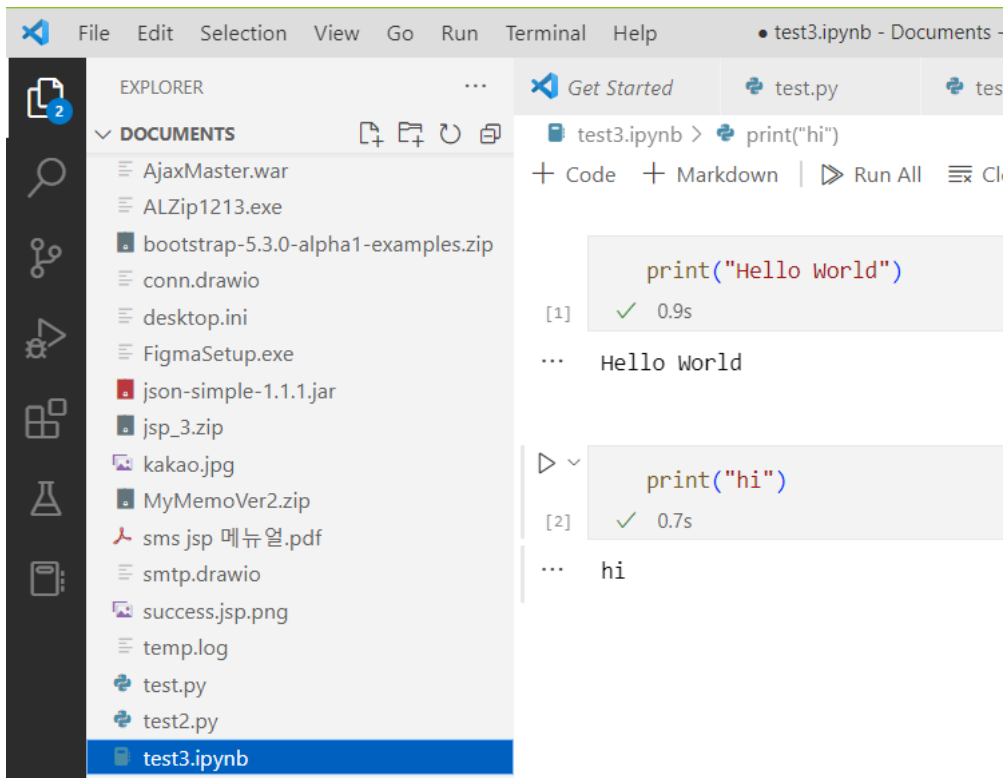
파이썬의 약점

- 파이썬은 모바일 앱을 만들수는 없다. (아이폰 앱 안드로이드 폰 앱 등은 만들수 없다.)
- 시스템과 아주 밀접한 프로그래밍 영역의 접근은 조금 힘들다. (C 라이브러리를 사용할수 있기 때문에 앞으로는 접근 가능한 영역이 될것이다.)

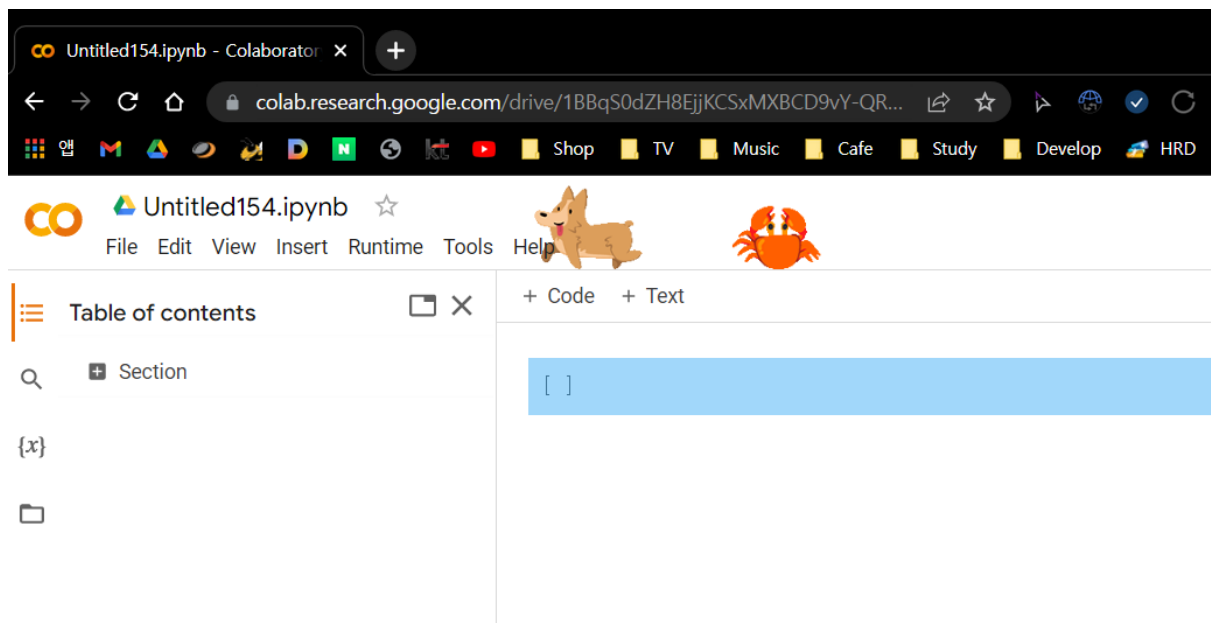
파이썬 설치

- anaconda 를 설치하면 파이썬을 설치해 주고 Jupyter Notebook 이라는 개발환경 그리고 많이 사용되는 라이브러리들과 가상환경 설정까지 한 번에 끝낼수 있다.
- 웹 개발환경인 Colab은 설치의 과정이 필요없이 바로 파이썬 개발이 가능하다. 다만 파이썬 으로 윈도우 앱을 개발하거나 웹 개발 등은 성격상 코랩에서는 진행 할수 없다.

파일 만들때 확장자를 파일명.ipynb 로 적으면 vs 코드에서도 라인바이라인 실행 가능하다.



파이썬 웹 개발환경 Colab



Data Type

자바

int a = 20;

String b = "Hello";

a= "hi"; (X)

파이썬

a=20

b="Hello"

a="hi" (O)

파이썬과 자바스크립트 등은 타입을 강제하지는 않지만 이점이 반드시 장점으로만 작용하지는 않는다.

어떤 프로그래밍 언어든지 그 언어의 데이터 타입을 잘 알고 이해한다면 이미 그언어의 절반을 터득한 것이나 마찬가지다.

특히 파이썬에서는 List, Tuple, Dict, Set 네가지 자료형이 아주 중요한 의미를 갖는다.

숫자형 자료형

- 정수형 Integer 양의 정수 음의 정수 0

```
# 정수형 자료형
a = 100
b = -123
c = 0
```

- 실수형 float 소수점이 포함된 숫자

```
# 실수형 자료형
a = 3.14
b = -5.16
```

- 숫자형 자료형에서의 연산

```
# 숫자형 자료형에서의 연산

a = 3
b = 4

# 코랩에서는 print 를 생략해도 알아서 print() 넣어준다. 마지막에만 넣어준다..
a + b
a - b
```

➡ -1

```
# 숫자형 자료형에서의 연산

a + b
a - b

print(a+b)
print(a-b)

# print() 구문을 생략하면 마지막 구문만 출력해주기 때문에 전부 출력하고 싶으면
# 각각 print() 구문을 다 적어야 한다. 이것은 파이썬의 특징이 아닌 코랩의 특징이다.
```

7
-1

```
# 나머지 연산자 - 홀수 짝수 배수 확인
```

```
5 / 2 # 일반적인 나머지 구하기
```

```
2.5
```

```
[8] 5 % 2
```

```
# 나눈 결과의 몫은 버리고 나머지만 취한다.
```

```
# 따라서 어떤수를 2로 나눈 나머지가 0이 나오면 짝수 1이 나오면 홀수
```

```
1
```

```
[9] 8 % 2
```

```
# 어떤수 8를 어떤수 2로 나눈 나머지가 0이면 배수이다.
```

```
0
```

문자열 자료형

- 문자열 String 자료형은 문자 또는 단어로 구성된 글자들의 집합이다.

```
[10] # 문자열 자료형 - 파이썬에서는 "" 와 ''가 완전히 동일하게 사용된다.
```

```
# 1 큰 따옴표
```

```
"Hello World"
```

```
# 2 작은 따옴표
```

```
'Hello World'
```

```
# 3 3중 따옴표 Doc String
```

```
"""Hello World"""
```

```
'''Hello World'''
```

```
'Hello World'
```



```
# 파이썬에서는 큰따옴표와 작은 따옴표에 전혀 차이가 없지만
```

```
# 아래의 용도로 분리하여 사용된다.
```

```
"Python's" # 작은 따옴표 하나를 출력하기 위한 용도
```

```
'say "Hello"' # 작은 따옴표 안에서 큰 따옴표를 출력하기 위한 용도
```

```
'say "Hello"'
```

```

▶ # \n 줄바꿈
a = "I love you\nYou love me?"
print(a)

b = """I love you
You love me?"""
print(b)

# 여러 발생
# c = "I love you
# You love me "

```

```

↳ I love you
   You love me?
   I love you
   You love me?

```

문자열 연산

```

▶ # 문자열 연산

# 파이썬에서는 문자열도 연산이 가능하다. 특히 다른 언어에서는 발견하기 어려운 파이썬만의
# 독특한 특징이다. 이런 점들이 바로 파이썬은 우리의 생각을 그대로 반영할수 있다는 파이썬 만의
# 장점들이다.

a = "I "
b = "love "
c = "you"

print(a+b+c)

d = "love "
print(d*3)

e = "="
print(e *30)

```

```

I love you
love love love
=====

```



문자열 연산

파이썬에서는 문자열도 연산이 가능하다. 특히 다른 언어에서는 발견하기 어려운 파이썬만의
독특한 특징이다. 이런 점들이 바로 파이썬은 우리의 생각을 그대로 반영할수 있다는 파이썬 만의
장점들이다.

```
a = "I "
b = "love "
c = "you"

print(a+b+c)

d = "love "
print(d*3)

e = "="
print(e *30)
```

```
I love you
love love love
=====
```



인덱싱과 슬라이싱 indexing & slicing

참고로 인덱싱과 슬라이싱은 리스트와 튜플에서도 완전히 동일하게 사용된다.

인덱싱 - 한 개의 글자 추출

```
a = 'I love You'
print(a[0])
print(a[9])
print(a[-1]) # 마지막 글자 추출은 -1 을 사용한다.
print(a[-3])
```

슬라이싱 - 여러개의 글자 추출

```
a = 'I love you'
print(a[2] + a[3] + a[4] + a[5])
print(a[2:6]) # 끝자리를 나타내는 수는 미만을 의미한다.
print(a[7:]) # 끝자리를 생략하면 끝까지 출력한다.
print(a[:2]) # 시작 자리를 생략하면 처음 즉 0 부터 시작한다.
print(a[:]) # 모든 자리 생략하면 처음 부터 끝까지 출력한다.
print(a[7:-1])# 슬라이싱에서도 - 표기를 사용한다. 역시 미만으로 사용된다.
```



```
I
u
u
Y
love
love
you
I
I love you
yo
```

문자열 출력 방법 - 총 4가지 방법 중 2가지를 가장 많이 사용한다.



파이썬에서 출력 방법 총 4가지

1 문자열 포매팅 (중요도 하)

```
print("I %s you" % "love")
print("I ate %d apples" % 5)
print("I ate %s apples" % 5)

print("I %s you, I ate %s apples" % ("love", 5))
```

2 format() 포맷 함수 (중요도 하)

```
print("I {} you".format("love"))
print("I ate {} apples".format(5))
print("I ate {} apples".format(5))

print("I {} you, I ate {} apples".format("love", 5))
```



```
I love you
I ate 5 apples
I ate 5 apples
I love you, I ate 5 apples
I love you
I ate 5 apples
I ate 5 apples
I love you, I ate 5 apples
```



3 f스트링 (중요도 상)

```
a = "love"
b = 5

print(f"I {a} you")
print(f"I ate {b} apples")
print(f"I ate {b} apples")

print(f"I {a} you, I ate {b} apples")
```



```
I love you
I ate 5 apples
I ate 5 apples
I love you, I ate 5 apples
```




4 연산자 사용


```
a = "love"
b = 5

print("I " + a + " you")
print("I ate" , b , "apples")
print("I ate" , b , "apples")
print("I " + a + " you, I ate" , b , "apples")
print("I" , a , " you")
print("I ate" , b , "apples")
print("I ate" , b , "apples")
print("I" , a , " you, I ate" , b , "apples")
```



```
I love you
I ate 5 apples
I ate 5 apples
I love you, I ate 5 apples
I love you
I ate 5 apples
I ate 5 apples
I love you, I ate 5 apples
```

Google Colaboratory

 <https://colab.research.google.com/drive/1iePnbdEXt63GZ9xUQQR-F8ls0t01FJIZ?usp=sharing>



git 관련 명령어

mac

iTerm2

- git --version : 깃 버전 확인
- git config --global init.defaultBranch main : 깃 설치 후 최초 1회만 실행(대충 오류 방지)
- git config --global user.name "이름" : 유저네임 세팅
- git config --global user.email "이메일" : 유저이메일 세팅
- git config --list : 제대로 되었는지 확인
- ls : 폴더 조회 cd 폴더명 : 폴더로 들어가기
- cd .. : 상위폴더로 나가기
- cd \ : 최상위 폴더로 나가기
- mkdir 디렉토리명 : 디렉토리 만들기
- rmdir 디렉토리명 : 디렉토리 삭제
- touch 파일명.확장자 : 파일 만들기
- cat 파일명.확장자 : 파일 읽기
- rm 파일명.확장자 : 파일 삭제

- `rm -rf .git` : 기존 git history 삭제
- `git add .` : 모든 파일을 새롭게 git에 업로드 하기 위한 명령어
- `git init` : .git.폴더 생성 (깃 관리의 시작) / 새로운 git 설정
- `git commit -m "메세지"` : 커밋 "커밋내용작성"
- `git remote add origin {리포지토리 주소}` : 연결!!!!!!
- `git status` : 현재 깃 상태를 보여줌 `git add (.)` : 파일 추적 (.은 모두를 의미)
- `git log` : 커밋로그 확인
- `git reset --hard` : 수정전 커밋으로 롤백
- `git reset --hard` 커밋해쉬코드(번호) : 해당하는 번호 커밋으로 롤백
- `git push origin main` : 리포지토리에 커밋들을 넣는다
- `git clone` 리포지토리 주소 : 리포지토리에서 모든 커밋들을 내 로컬로 가져옴.
- `git pull origin main` : 서버 변동사항을 로컬로 동기화
- `.gitignore` : 원하지 않는 파일을 안 올릴 수 있는 방법 (확장자를 없앤 파일에 .txt 등 쓰고 저장하면 무시됨)
- `git branch` : 생성된 브랜치의 목록
- `git branch` 이름: 브랜치 만들기
- `git checkout` 이름 : 이름이라는 브랜치로 이동

MAC) git 문제 Permission denied (publickey).

안녕하세요 :)오늘은 git을 쓰면서 발생할 수 있는 이슈에 대해서 설명드릴게요!제가 지금 다른 노트북을 쓰게됐는데, 여기서 GitHub에 push나 clone하려니까 이런 오류가 나게 됩니다. git 입장에서선 당연한 오류예요. 등록되지 않은 기기에서 clone이나 push가 일어나면 인증문제가 발생하게 된답니다.결과적으로 유저의 ssh가 등록되지 않아 접근권

<https://zeddios.tistory.com/120>

```
Permission denied (publickey).
You are not read from remote repository.
Please make sure you have the correct access rights
and the repository exists.
```

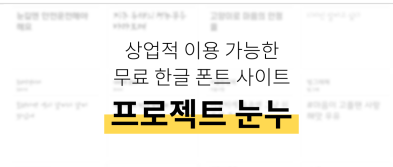
MyMEMO

프로젝트 눈누

눈누

상업용 무료한글폰트 사이트

<https://noonnu.cc/>



작성 메모 리스트

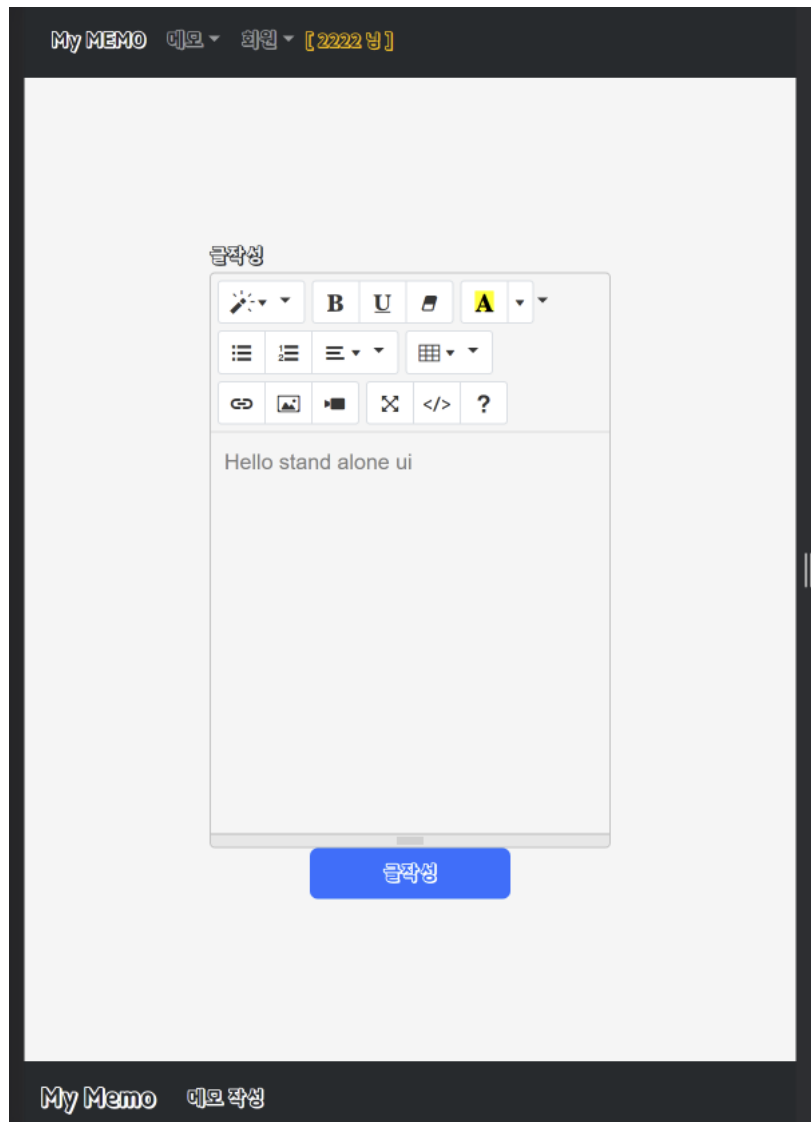
2222 2023-01-30 23:41:03
test button

2222 2023-01-30 23:32:09
fdsafdsafdsa
fdsa

2222 2023-01-30 22:30:25
summernote
1. test
hello

2222 2023-01-30 22:19:57
jhfgdhgfd

2222 2023-01-30 22:19:03
test



Gmail Setting

create memo, sign up

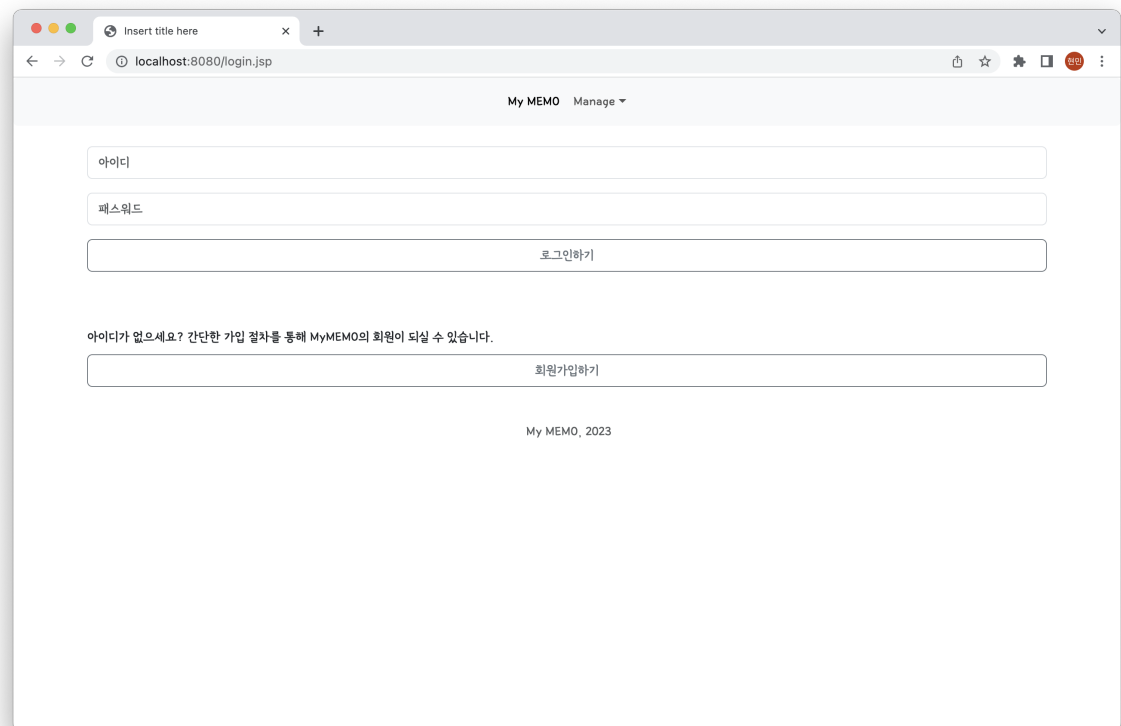
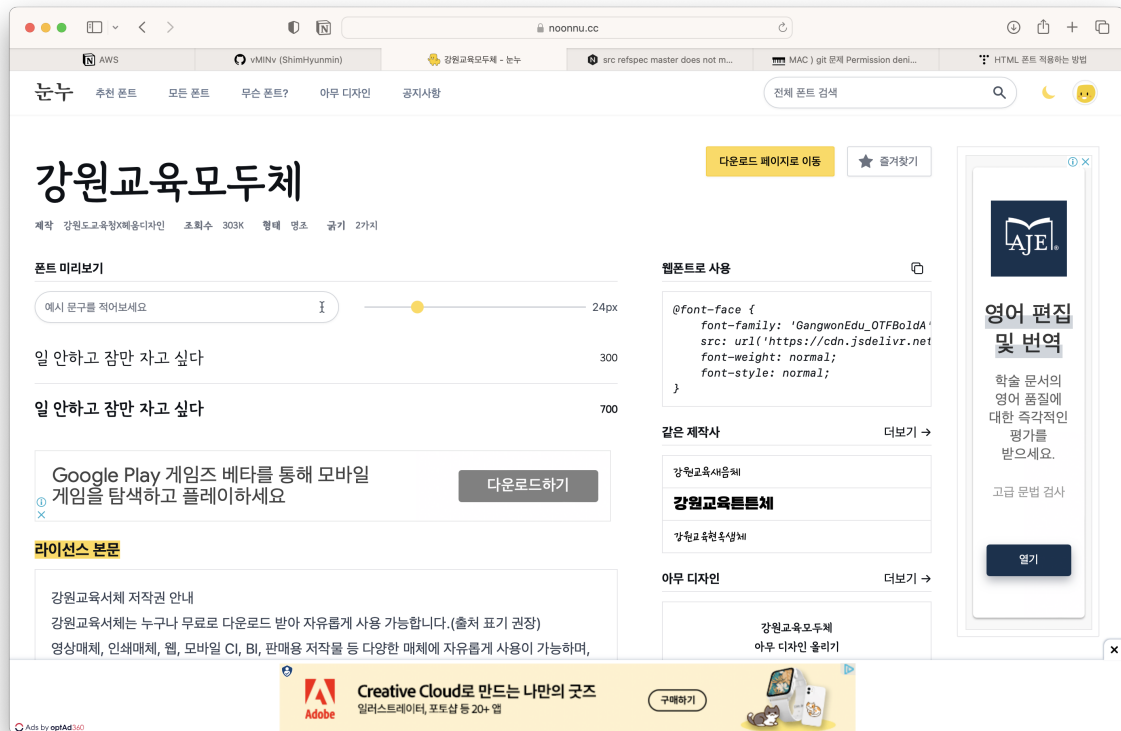
Font

sign-in.css

header 에 링크 포함 `<link href="sign-in.css" rel="stylesheet">`

```
@font-face {
  font-family: 'GangwonEdu_OTFBoldA';
  src: url('https://cdn.jsdelivr.net/gh/projectnoonnu/noonfonts_2201-2@1.0/GangwonEdu_OTFBoldA.woff') format('woff');
  font-weight: normal;
  font-style: normal;
}

html,
body {
  font-family: 'GangwonEdu_OTFBoldA';
}
```



Mail

가입 환영 메일 발송

smtp

Mail.java

```
package smtp;

import java.util.Map;
import java.util.Properties;

import javax.mail.Authenticator;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

public class Mail {
    private final Properties serverInfo;//서버 정보
    private final Authenticator auth;//인증 정보

    public Mail() {
        //생성자 2가지 특징
        //1. 클래스명과 동일 2. 리턴 타입이 없다
        serverInfo = new Properties();
        serverInfo.put("mail.smtp.host", "smtp.gmail.com");//메일 서버 지정
        serverInfo.put("mail.smtp.port", "465");//포트 번호 거의 대부분 동일
        serverInfo.put("mail.smtp.starttls.enable", "true");
        serverInfo.put("mail.smtp.auth", "true");
        serverInfo.put("mail.smtp.debug", "true");
        serverInfo.put("mail.smtp.socketFactory.port", "465");
        serverInfo.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
        serverInfo.put("mail.smtp.socketFactory.fallback", "false");

        serverInfo.put("mail.smtp.ssl.protocols", "TLSv1.2");//맥에서 반드시

        //사용자 인증 정보
        auth = new Authenticator() {
            @Override
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication("hyunmin04050@gmail.com", "앱 비밀번호");
            }
        };
    }

    public void emailSending(Map<String, String> mailInfo) throws MessagingException {
        //1 세션 생성 ( 네이버/다음/구글 메일 서버에 사용자 인증)

        Session session = Session.getInstance(serverInfo, auth);
        session.setDebug(true);

        //2 메세지 작성 (emailCheck.jsp 에서 만든 객체)

        MimeMessage msg = new MimeMessage(session);
        msg.setFrom(new InternetAddress(mailInfo.get("from")));//보내는 사람
        msg.addRecipient(Message.RecipientType.TO, new InternetAddress(mailInfo.get("to")));//받는사람
        msg.setSubject(mailInfo.get("subject")); //메일 제목
        msg.setContent(mailInfo.get("content"), mailInfo.get("format"));

        //3 전송

        Transport.send(msg);
    }
}
```

signupCheck.jsp

가입 성공 시 메일 발송

```
<%@page import="jdbc.*, java.util.*, smtp.*"%>
<%@page import="java.io.FileReader"%>
<%@page import="java.io.FileOutputStream"%>
<%@page import="java.io.BufferedReader"%>
```

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-6Lh1TQ8
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256-pvPw+upLPUjgMXy0G+800xUf+/Im1MZjXxxg0cBQBxU=" crossorigin=
<link href="etc/sign-in.css" rel="stylesheet">
<%
    request.setCharacterEncoding("UTF-8");

    String id = request.getParameter("id");
    String password = request.getParameter("password");
    String name = request.getParameter("name");
    String to = request.getParameter("to");

    //중복 아이디 체크
    if(UserDAO.exsist(id)){//존재, 아이디 사용불가능%
    <!-- Modal -->
    <div class="modal fade" id="exampleModal1" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header">
                    <h1 class="modal-title fs-5" id="exampleModalLabel">회원가입 실패</h1>
                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
                </div>
                <div class="modal-body">
                    이미 존재하는 아이디입니다. 다시 시도해주세요.
                </div>
                <div class="modal-footer">
                    <button onclick="location.href='signup.jsp'" class="btn btn-primary">회원가입</button>
                </div>
            </div>
        </div>
    </div>

    <script>
        $(function() {
            $("#exampleModal1").modal("show");
        });
    </script>
<%
    }else{
        //디비에 넣기
        int result = UserDAO.insert(id, password, name);

        if(result==1){//가입 성공

            Map<String, String> emailInfo = new HashMap<String, String>();
            emailInfo.put("from", "hyunmin04050@gmail.com");
            emailInfo.put("to", request.getParameter("to"));
            emailInfo.put("subject", "[구글] 회원가입을 환영합니다.");

            String htmlContent = "";

            String templatePath = application.getRealPath("Welcome.html");
            BufferedReader br = new BufferedReader(new FileReader(templatePath));

            String oneLine;
            while((oneLine = br.readLine()) != null ){
                htmlContent += oneLine + "\n";
            }
            br.close();

            htmlContent = htmlContent.replace("__NAME__", name);
            htmlContent = htmlContent.replace("__TO__", to);
            htmlContent = htmlContent.replace("__PASS__", password);

            emailInfo.put("content", htmlContent);//내용
            emailInfo.put("format", "text/html;charset=UTF-8");//메일의 형식

            Mail smtpServer = new Mail();//메일 전송 객체 생성
            smtpServer.emailSending(emailInfo);
        }%>
    <!-- Modal -->
    <div class="modal fade" id="exampleModal2" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header">
                    <h1 class="modal-title fs-5" id="exampleModalLabel">회원 가입 성공 </h1>
                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
                </div>
                <div class="modal-body">
                    환영합니다. 로그인 페이지로 이동합니다.
                </div>
                <div class="modal-footer">
                    <button onclick="location.href='login.jsp'" class="btn btn-primary">로그인 페이지</button>
                </div>
            </div>
        </div>
    </div>

```

```

</div>

<script>
  $(function() {
    $("#exampleModal2").modal("show");
  });
</script>
<% }
}%>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js" integrity="sha384-w76AqPfdkMBDXo30jS

```

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/nonuser/signup.jsp'. The page features a light gray header with a hamburger menu icon. The main content area contains a signup form with the following elements:

- A hamburger menu icon in the top left corner.
- Five input fields stacked vertically, labeled: '아이디' (ID), '패스워드' (Password), '패스워드 (확인)' (Password Confirm), '이름' (Name), and '가입을 확인 받을 이메일을 입력해주세요' (Please enter the email you want to receive the confirmation).
- A '회원가입하기' (Sign Up) button.
- A text line: '회원 가입 시 MyMEMO의 약관에 동의하신 것으로 간주합니다.' (By registering, you are deemed to agree to the MyMEMO Terms of Service).
- A '약관보기' (View Terms) button.
- At the bottom, the text 'My MEMO, 2023' is displayed.

MyMEMO project 파일 정리

경로 설정 주의

