

DB 연결

1. HTML

- 파일 저장 : 확장자 .html

```
<!-- Load c3.css -->
<link href="/path/to/c3.css" rel="stylesheet">
<script src="/path/to/d3.v5.min.js" charset="utf-8"></script>
<script src="/path/to/c3.min.js"></script>
```

```
<html> 시작
<title> 브라우저 상단 제목 </title>
<body>디자인
```

```
이건 내용이야
<font color = red > <br>
<b>ㅎㅎㅎㅎㅎ 떡볶이<b>
  </font>
```

```
<br>줄바꿈
```

```
<a href="http://www.google.com" target = "blank"> 구글로 이동
<img width=300 height=200 src =
"https://w.namu.1a/s/afce205b637145dc3d6d679f9dbcb0fd22c062aafe0f3f61ed18f7e3fce
d860ca721a65afcd9b0c3d60b56ded88c61a608ed157510e14afed525537feb273c46cd84bbfc186
f3ab829711bb280b9c3bde03226429e0979c8961458b8ade710ac89c8cb67db2c213464737de0405
3fd34">
```

```
</a>
```

```
<a href="http://www.google.com" target = "blank"> </a>
```

```
<table border='3' width='300'>
<tr>
<th bgcolor='#cccccc'>id </th><th>pw</th>
</tr>
<tr>
  <td>kim </td><td>111</td>
</tr>
</table><br>
```

```
<form action="/" method="post">
id : <input type="text" name = "id"><br>
pw : <input type="password" name = "pw"><br>
hidden : <input type="hidden" name = "sec"><br>
통신사 :<select name = "mob">
<option> 111</option>
```

```

<option>222</option>
</select><br>
취미 : <input type="checkbox" name="habit">등산
      <input type="checkbox" name="habit">낚시

<br>
성별 : <input type="radio" name="gen">남
      <input type="radio" name="gen">여
      <input type="radio" name="gen">기타
<br>

<input type="button" value="가입버튼">
<input type="reset" value="초기화">
<input type="submit" value="서버로전송">

</form>
<br>
<a href="/?bseq=3&bg=A"> c로이동</a>

</body><br>
</html> 끝<br>
=====
=====
<br>
<form></form> --><form>은 닫히기 전까지 겹쳐서 사용 불가능<br>
<form action="경로" method="메소드종류">
<br>    ->action : 사용자에게 입력 받은 것을 보낸다
<br>    -><form></form> --><form>은 닫히기 전까지 겹쳐서 사용 불가능<br>
<br>    ->get: 단순히 링크걸어서 전달할 때 사용 , 밖으로 값이 보여도 되는 것에 사용
<br>    ->post:전송 값 숨김, 버튼이 있어야 함(form 태그로 감싼 submit 형태, method가
post일때 전송 가능), 전송 크기에 제약이 없다->그외 모두 get
서버로 전송하기 위해 이름 필요

<br>줄바꿈

"사이트 연결은 꼭 알아줘야 한다.", <br>
새창으로 열기 blank, <br>
<a> 사이에 넣으면 그림도 연결<br>

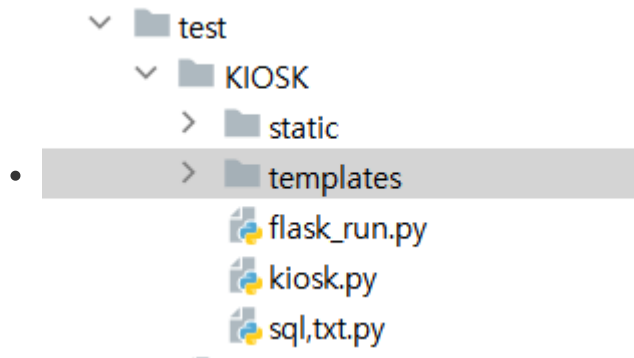
th는 제목으로 쓰기 : 두껍고 가운데<br>
td는 보통 값<br>
<th bgcolor='#cccccc'> 색 지정<br>
hidden:사용자 모르게 넘길 때<br>
<input type="radio" name="gen">이름 똑같이 해주면 하나로 묶임 ->한개만 선택가능

```

2.키오스크

2.1 파이썬

- 파이썬 디렉토리 새로 만들기 : static, templates
- static에 KIOSK의 폴더 5개 이동(css,fonts,images,img,js)
- templates에 그외 파일 모두 이동



#<flask_run.py>

#flask : 파이썬으로 구동 가능한 웹 프레임워크

#필요한 동적인 것 static

#디자인 templates

#프레임워크(기본 뼈대,구조)-->유지,관리 편함

#input,output만 알면 구동 가능

#-----

pip install flask, requests(통신 API)

#-----

```
from flask import Flask, render_template,make_response, jsonify, request
```

```
app = Flask(__name__)
```

```
@app.route('/')
```

```
def index():
```

```
    return render_template('index.html')
```

```
if __name__ == '__main__': #내가 직접돌리는 메인 구동이다
```

```
    app.debug = True #개발자 모드 , 코드
```

```
    app.run(host='localhost', port=80)
```

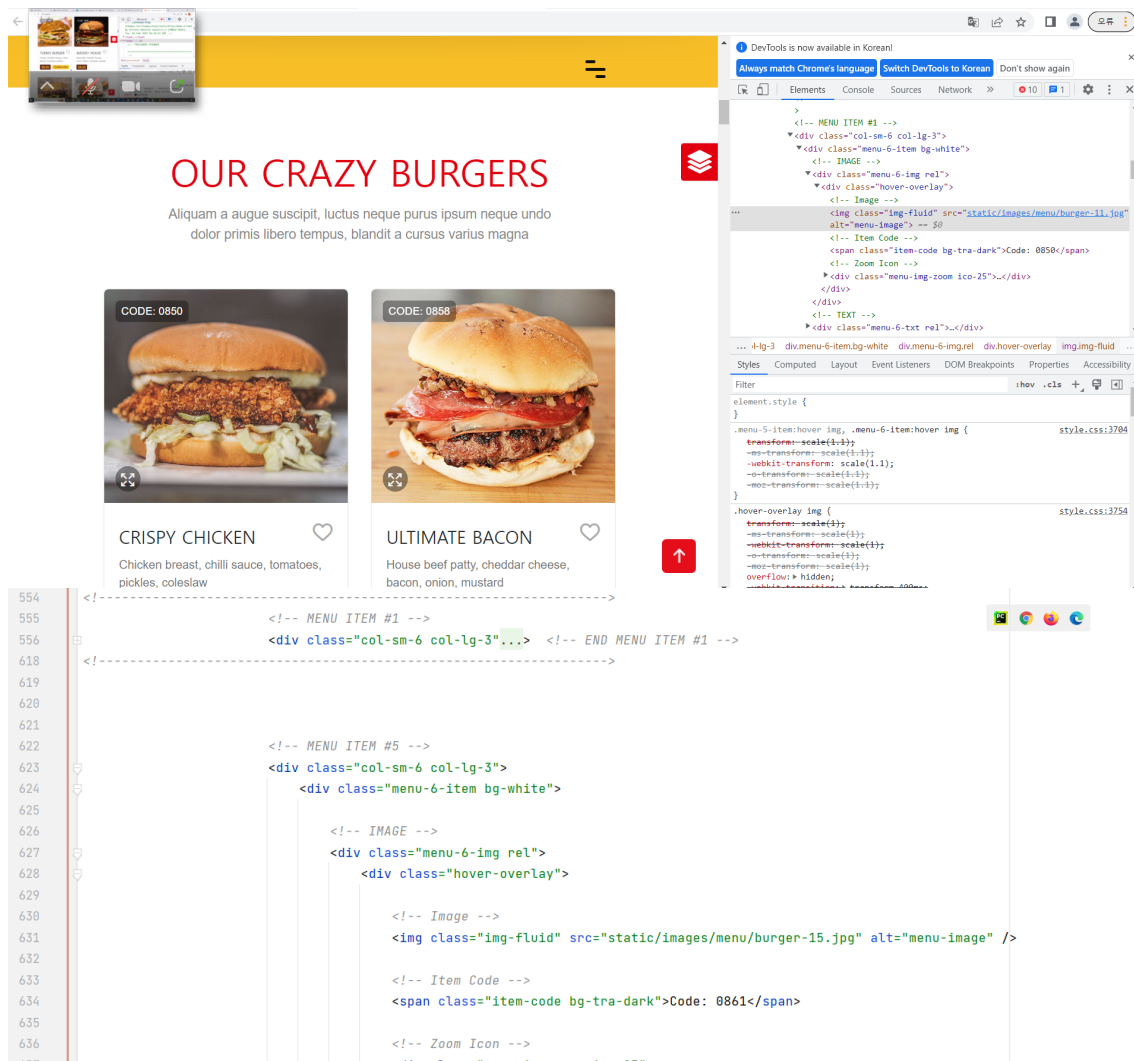
```
    #flask가 구동해라, py파일이어도 flask가 코드를 구동하여 보여줌
```

```
    #웹서비스 기본포트 80
```

```
    #첫 진입점(화면) index
```

2.2 crom

- crom에서 F12 -->해당 이미지에서 우클릭-->검색 -->해당 코드 찾기
- index 파일에서 이미지2,3,4 지우기



2.3 오라클

```
insert into KIO_GOODS(GOOD_SEQ,GOOD_NAME,GOOD_IMG,GOOD_PRICE,GOOD_DESC,REG_DATE)
values (KIO_GOODS_SEQ.nextval,'CRISPY CHICKEN','static/images/menu/burger-
11.jpg', 11, 'Chicken breast, chilli sauce, tomatoes, pickles,
coleslaw',sysdate);
```

```
insert into
KIO_GOODS(GOOD_SEQ,GOOD_NAME,GOOD_IMG,GOOD_PRICE,GOOD_DESC,REG_DATE)
values (KIO_GOODS_SEQ.nextval, 'ULTIMATE BACON','static/images/menu/burger-
12.jpg',8,'House beef patty, cheddar cheese, bacon, onion, mustard',sysdate);
```

```
insert into
KIO_GOODS(GOOD_SEQ,GOOD_NAME,GOOD_IMG,GOOD_PRICE,GOOD_DESC,REG_DATE)
values (KIO_GOODS_SEQ.nextval, 'BLACK SHEEP','static/images/menu/burger-
13.jpg',9,'American cheese, tomato relish, avocado, lettuce, red
onion',sysdate);
```

```
insert into
KIO_GOODS(GOOD_SEQ,GOOD_NAME,GOOD_IMG,GOOD_PRICE,GOOD_DESC,REG_DATE)
values (KIO_GOODS_SEQ.nextval, 'DOUBLE BURGER','static/images/menu/burger-
14.jpg',10,'2 beef patties, cheddar cheese, mustard, pickles,
tomatoes',sysdate);
```

2.4 파이썬

- flask for문 기본 형태

```
# flask for문
MY_LIST=[]
{% for val in MY_LIST%}
    {{val}},{{val}}
{% endfor %}
```

- test: test.html 새로 만들기

```
#flask_run.py
#test
@app.route('/list')

def test_list():
    test_list = [1,2,3]
    test_dict = {"uid":"kim", "upw":"111"}
    test_list_dict= [ {"uid": "kim1", "upw": "555"},
                      {"uid": "kim2", "upw": "666"} ]

    return render_template('test.html'
        , KEY_TEST_LIST = test_list
        , aaa = test_dict
        , KEY_TEST_LIST_DICT = test_list_dict
```

```
#test.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>

<body>
dfdf
test_list = [1,2,3]
{% for val in KEY_TEST_LIST %} {{ val }}
{% endfor %}
test_dict = {"uid":"kim", "upw":"111"}
aaa = test_dict
{{aaa.uid}} , {{aaa.upw}}
test_list_dict= [ {"uid": "kim1", "upw": "555"}, {"uid": "kim2", "upw": "666"} ]
KEY_TEST_LIST_DICT = test_list_dict

{% for dic in KEY_TEST_LIST_DICT %} {{ dic.uid }} , {{ dic.upw }}
{% endfor %}
{{ KEY_TEST_LIST_DICT[1]['upw'] }}
```

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

```
a href = " , src = " -->static/  
product_single -->/add_cart
```

```
detail_view, add_cart, order,>flas> db >web page
```

