WEB!VULN

ganghuichan-i gaesaekkiya jiga hagi gwichanhdago namdeulhante jjamttaeliji mala.
i michinsaekkiya swibal igeol naega wae mandeulgo iss-eoyahanya ehyu michinsg

Team_SCA

제작 : 박민기

Contact

웹 해킹 심화 개념

이 단계에서 뭘 배우냐?

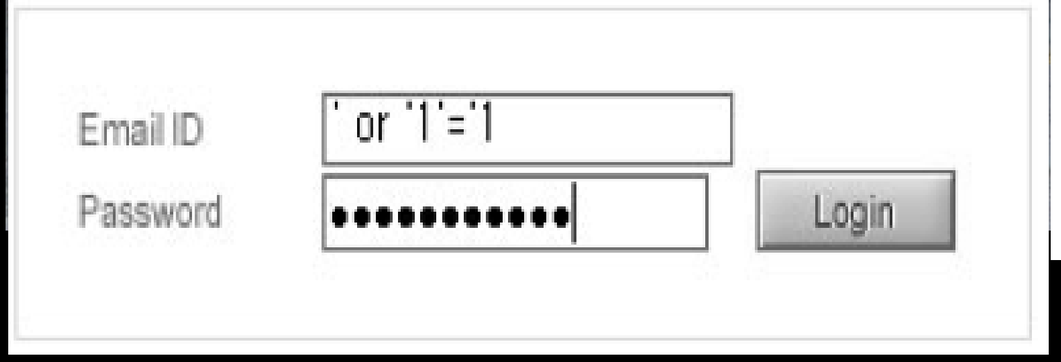
솔직히 만들기 싫은데 뭐 하긴해야하니깐 대충 만들어는 봄
여기서부터는 웹해킹 기초-심화 개념 단계라 확실히 알아두는게 좋음
알아서 독학해도 되지만 좀 그럼 이해가 안되고 귀찮고 하니깐 내가 알려줌

진짜 기초 개념이랑 원리 배울거임 그리고 약간의 실습 포함해서
진짜 기초니깐 걍 알고있어야됨
물론 해킹하면 다 알게되긴함

추가로 이거 디자인은 신경 안썼으니깐 그냥 보셈

Read More

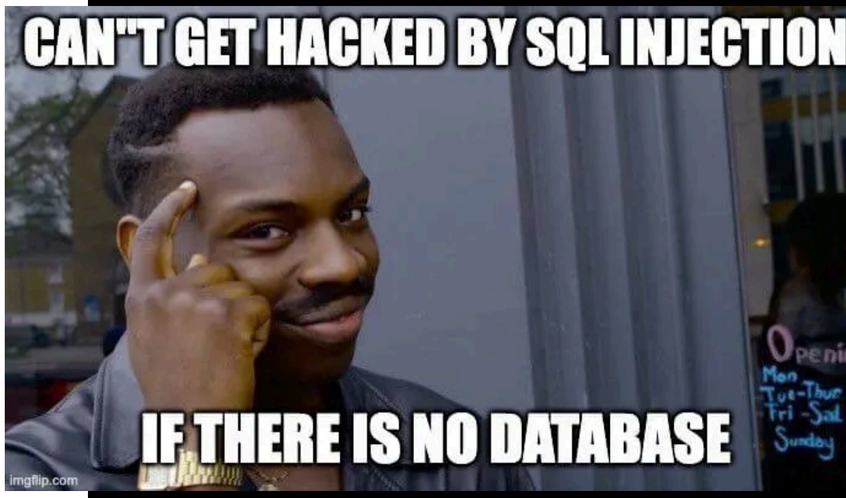
SQL STRUCTURED QUERY LANGUAGE INJECTION



SQL INJECTION

SQL Injection MEMExplained

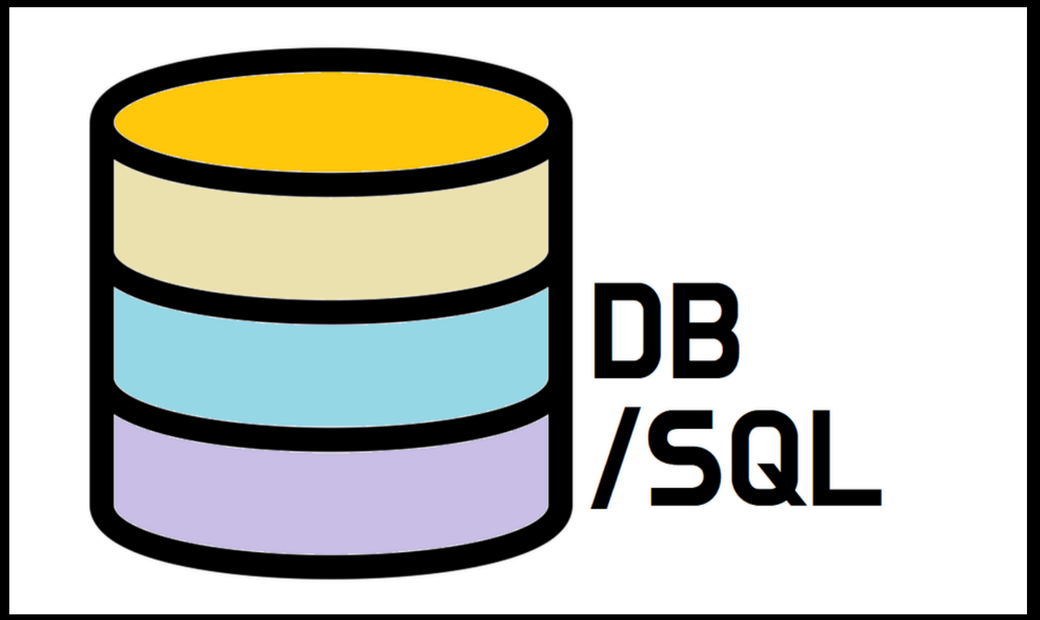
Learning about SQL Injection with memes...



Structured Query Language

구조화된 질의 언어

이걸 어디에 갔다 쓰냐?



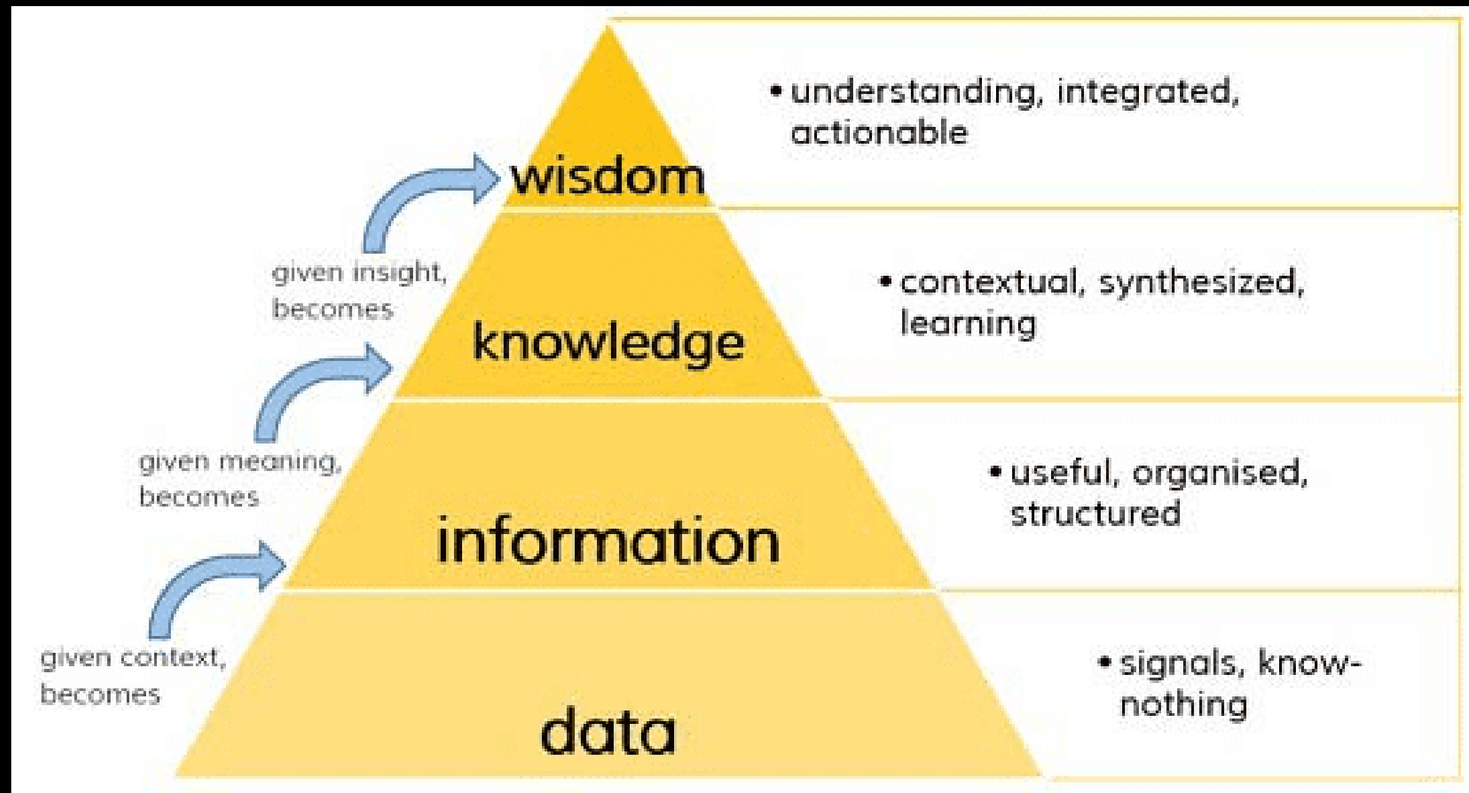
Structured Query Language

구조화된 질의 언어

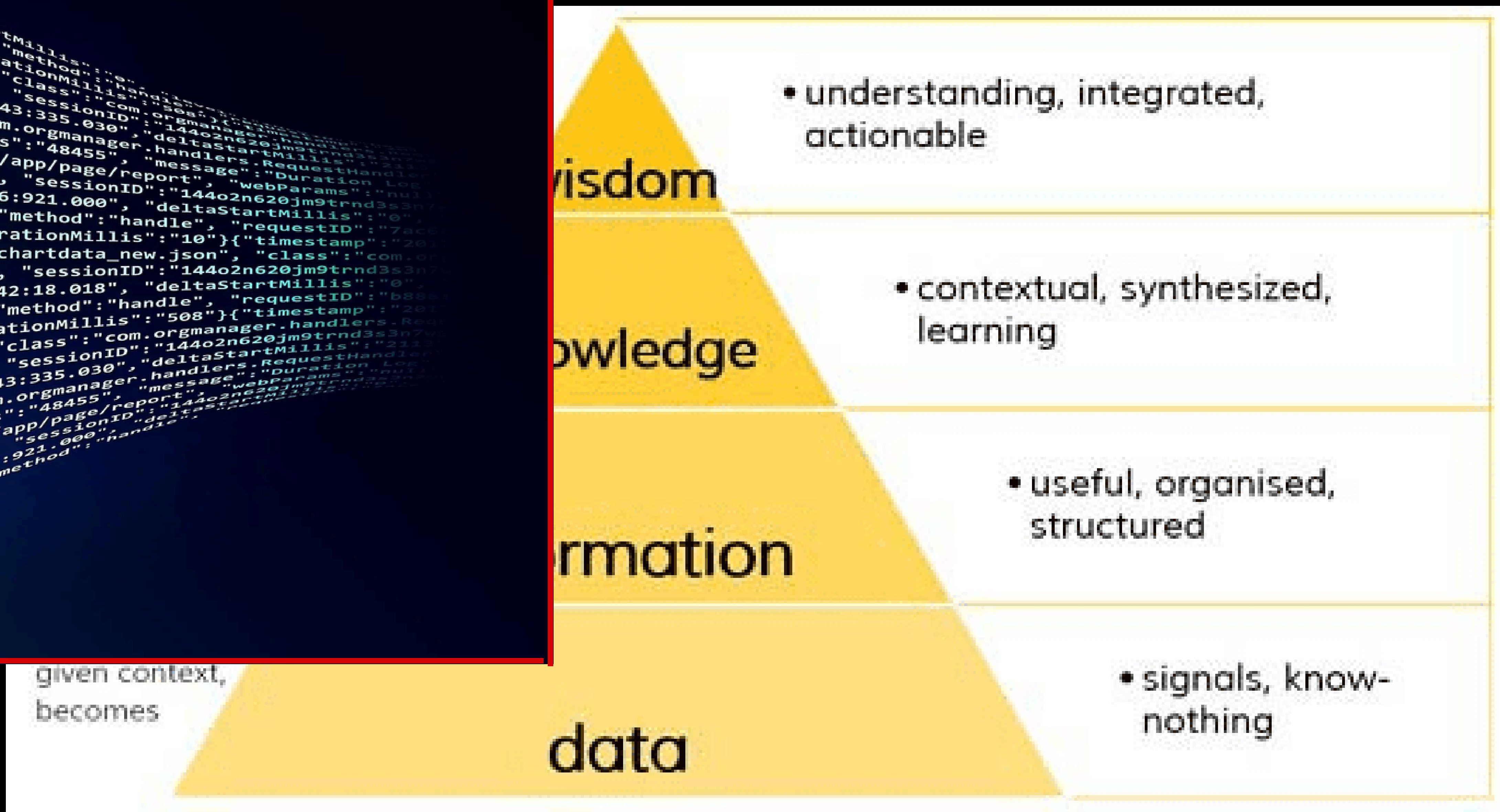
여기

일명 ‘데이터 베이스’

데이터 : 진짜 그 데이터
데이터 베이스 = 데이터 졸라 많은거

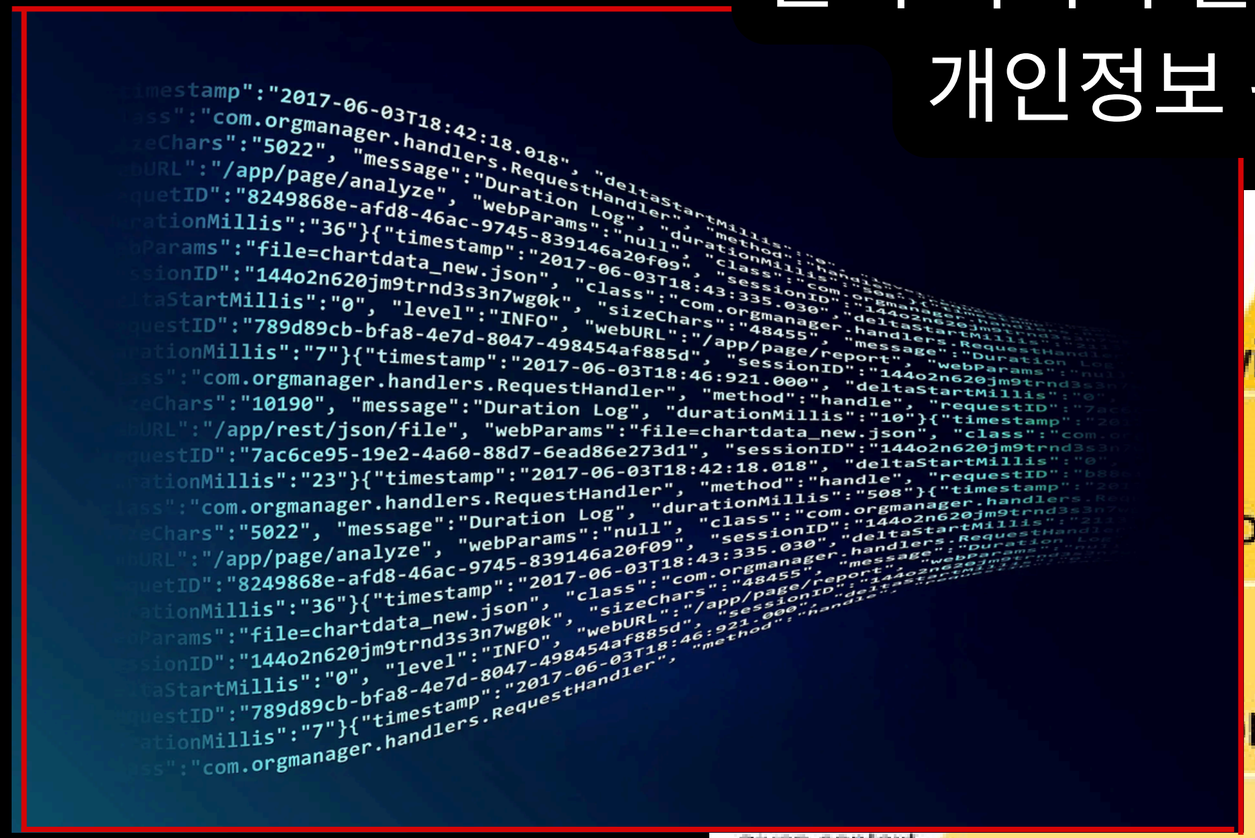


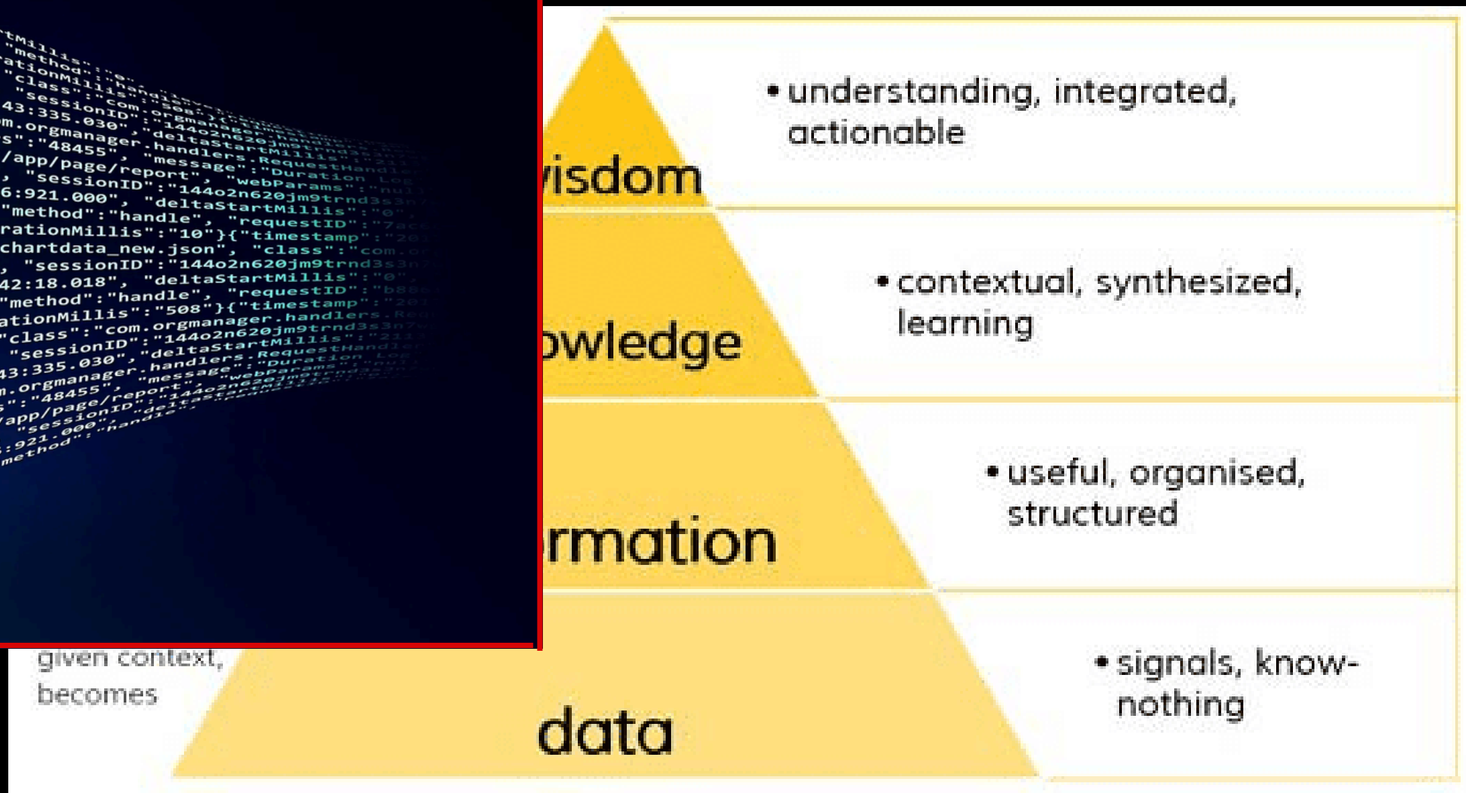
데이터 : 진짜 그 데이터
데이터 베이스 = 데이터 졸라 많은거



데이터 : 진짜 그 데이터
데이터 베이스 = 데이터 졸라 많은거

흔히 여기서 말하는 데이터는
개인정보 뭐 이런거

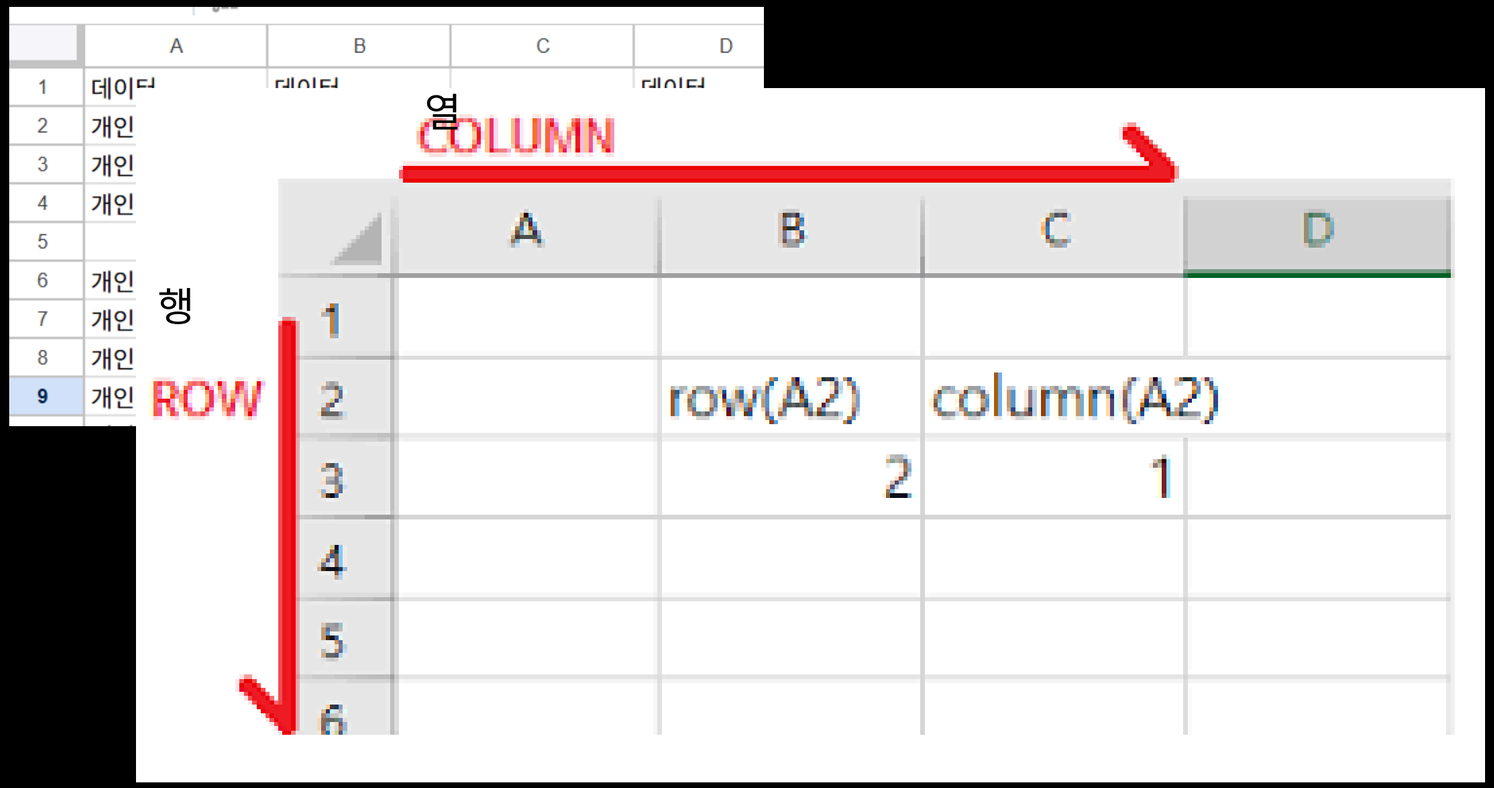




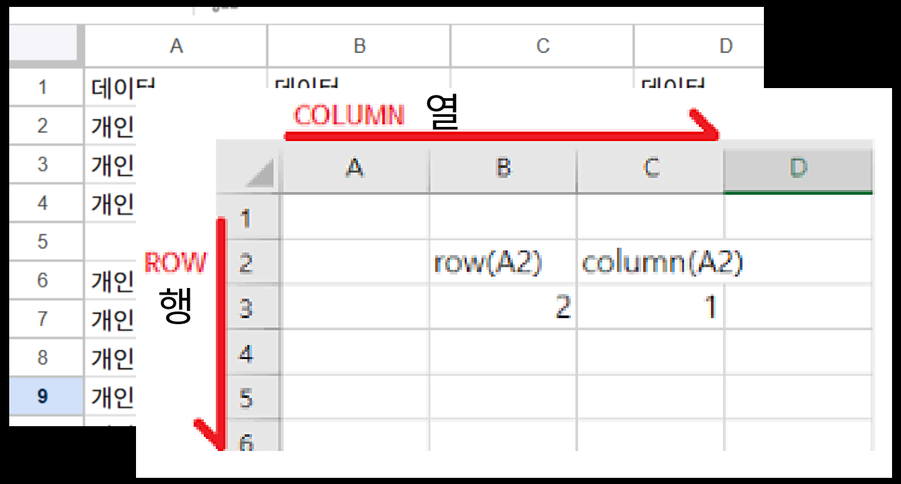
데이터 베이스 형태 :
엑셀 생각하면됨

	A	B	C	D
1	데이터	데이터		데이터
2	개인정보	개인정보		개인정보
3	개인정보	개인정보		개인정보
4	개인정보	개인정보		개인정보
5				개인정보
6	개인정보	개인정보	개인정보	개인정보
7	개인정보	개인정보	개인정보	개인정보
8	개인정보	개인정보	개인정보	개인정보
9	개인정보	개인정보	개인정보	개인정보

데이터 베이스 형태 :
엑셀 생각하면됨

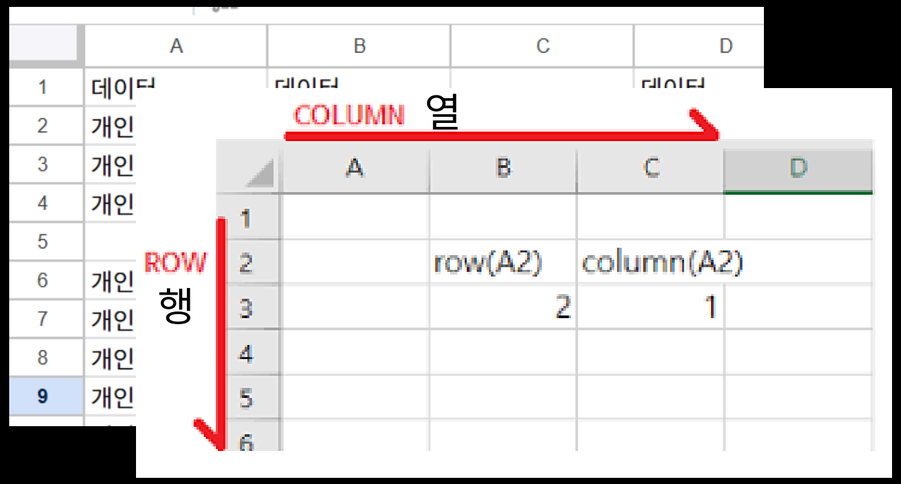


데이터 베이스 형태 :
엑셀 생각하면됨



이런 엑셀(데이터베이스)을
‘명령어’로 조작한다고 보면됨
= SQL

데이터 베이스 형태 :
엑셀 생각하면됨



이런 엑셀(데이터베이스)을
‘명령어’로 조작한다고 보면됨
= SQL

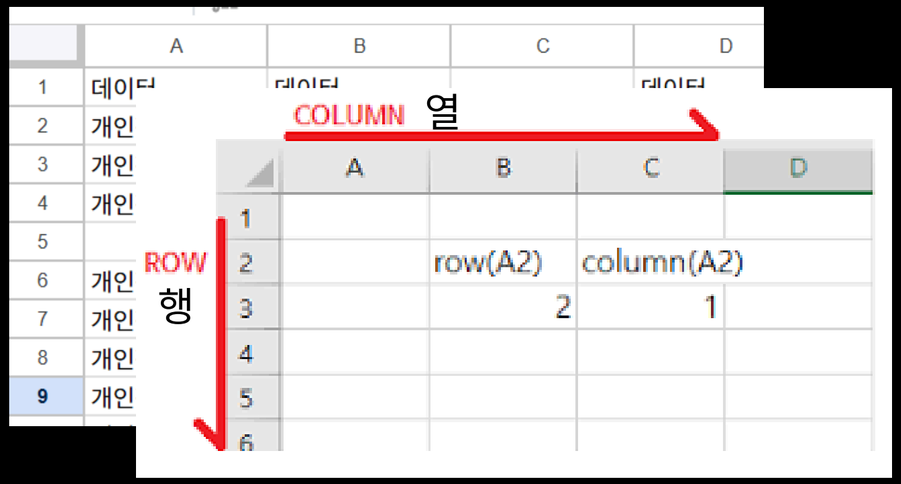
뭐~ 어디 몇행 숫자 1로 바꿔주세요~

뭐 뭐~ 이거 어디에 있는지 알려주세요

뭐~ 이거 뭐 어딨어요?~ 뭐 이거 이걸로 바꿔주세요

이거 여기서부터 여기까지 뭔지 알려주세요 ~

데이터 베이스 형태 :
엑셀 생각하면됨



이런 엑셀(데이터베이스)을
‘명령어’로 조작한다고 보면됨
= SQL

뭐~ 어디 몇행 숫자 1로 바꿔주세요~

뭐 뭐~ 이거 어디에 있는지 알려주세요

뭐~ 이거 뭐 어딨어요?~ 뭐 이거 이걸로 바꿔주세요

이거 여기서부터 여기까지 뭔지 알려주세요 ~

```
UPDATE table_name
SET column_name = 1
WHERE id = 특정행_번호;

SELECT *
FROM table_name
WHERE column_name = '찾는값';

UPDATE table_name
SET column_name = '바꿀값'
WHERE column_name = '기존값';

SELECT *
FROM table_name
WHERE column_name BETWEEN '시작값' AND '종료값';
```

그래서 SQL INJECTION? 뭔데 씹X덕아

그래서 SQL INJECTION? 뭔데 씹X덕아

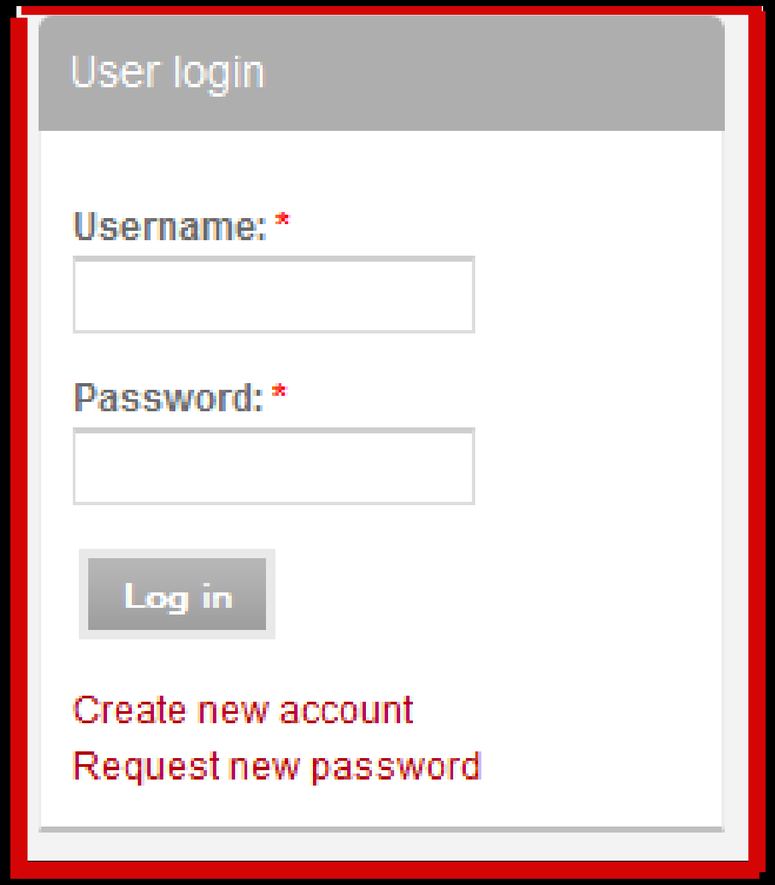
보통 서버가 관리하는 DB는 아주아주 중요함
(CLIENT)이가 SQL문을 간접적으로 사용하는 경우가 있음

예?를 들어 서~~~~~~~~

그래서 SQL INJECTION? 뭔데 씹X덕아

보통 서버가 관리하는 DB는 아주아주 중요함
(CLIENT)이가 SQL문을 간접적으로 사용하는 경우가 있음

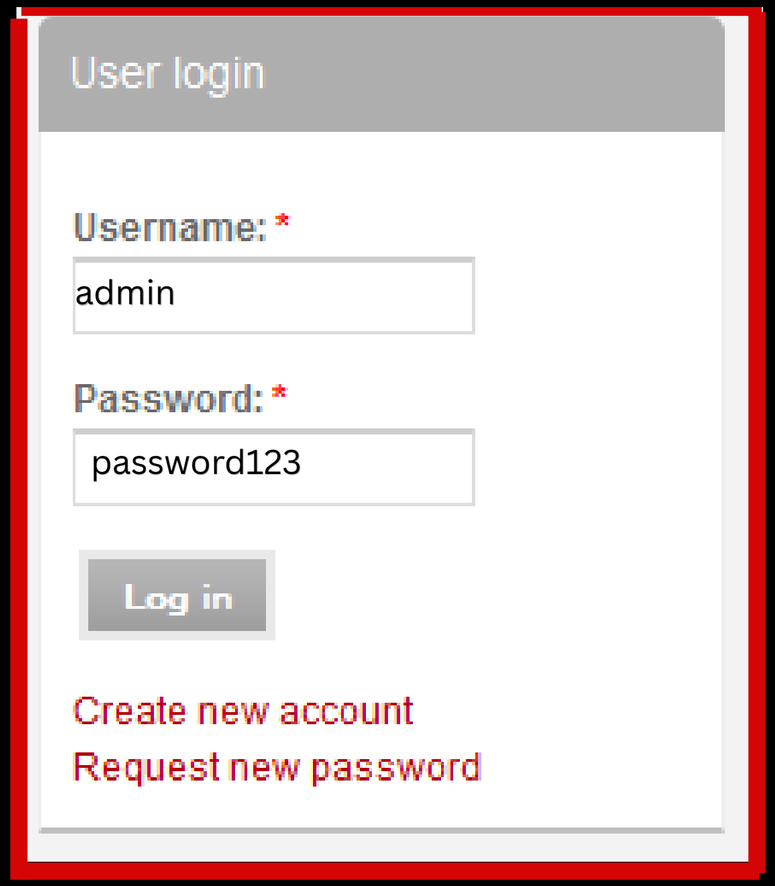
예?를 들어 서~~~~~~~~



그래서 SQL INJECTION? 뭔데 씹X덕아

보통 서버가 관리하는 DB는 아주아주 중요함
(CLIENT)이가 SQL문을 간접적으로 사용하는 경우가 있음

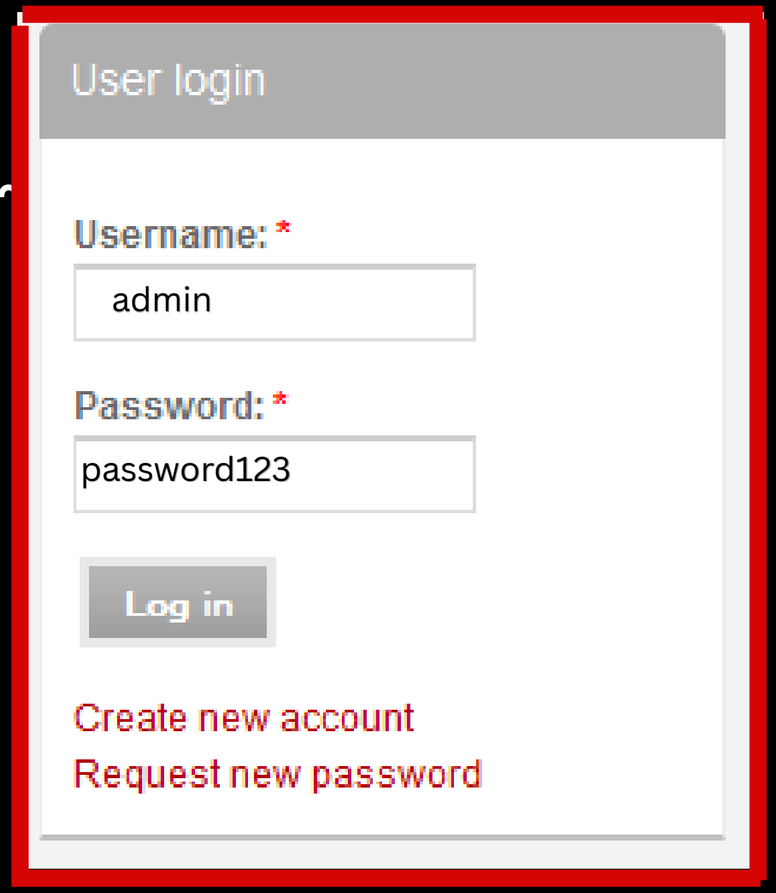
예?를 들어 서~~~~~~~~



그래서 SQL INJECTION? 뭔데 씹X덕아

보통 서버가 관리하는 DB는 아주아주 중요함
(CLIENT)이가 SQL문을 간접적으로 사용하는 경우가 있음

예?를 들어 서~~~~~

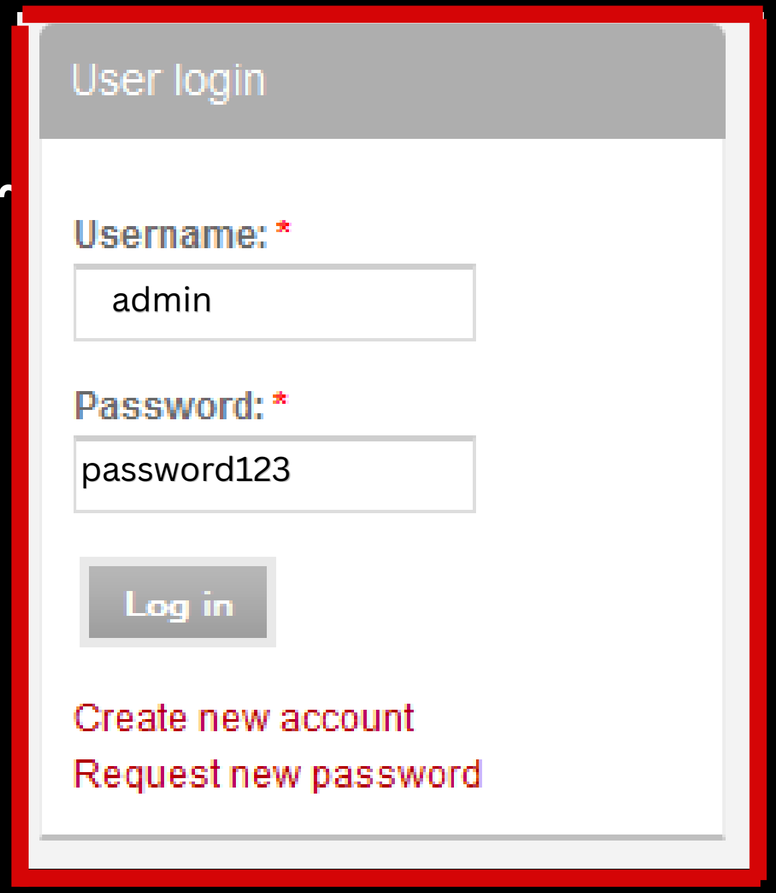



```
SELECT *
FROM users
WHERE user_id = 'admin' AND password = 'password123';
```

그래서 SQL INJECTION? 뭔데 씹X덕아

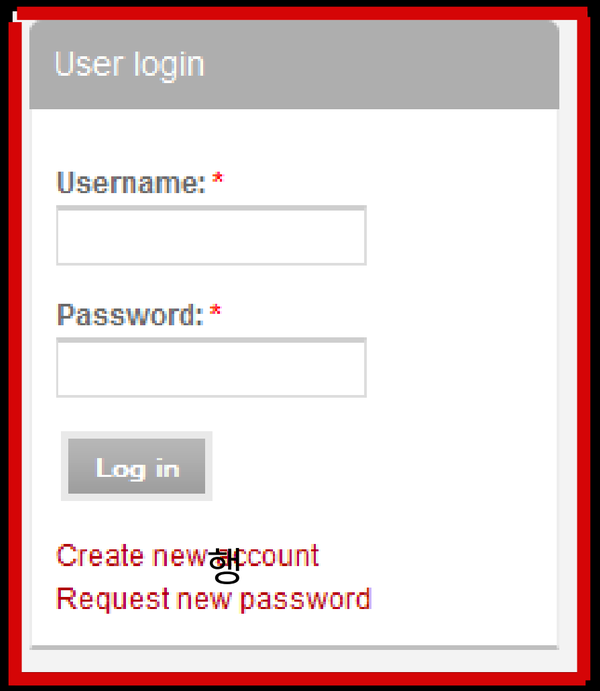
보통 서버가 관리하는 DB는 아주아주 중요함
(CLIENT)이가 SQL문을 간접적으로 사용하는 경우가 있음

예?를 들어 서~~~~~




```
SELECT *
FROM users
WHERE user_id = 'admin' AND password = 'password123';
```

이런식으로 사용자가 SQL값을 ‘넣을 수‘있다.




```
SELECT *
FROM users
WHERE user_id = 'admin' AND password = 'password123';
```

이런식으로 사용자가 SQL값을 ‘넣을 수‘있다.

여기서 알아 두어야 할 해킹세계 기본상식!

- 사용자의 input을 받는다.
- 값 조작이 가능하다.
- 서버의 구조가 보인다.

User login

Username: *

Password: *

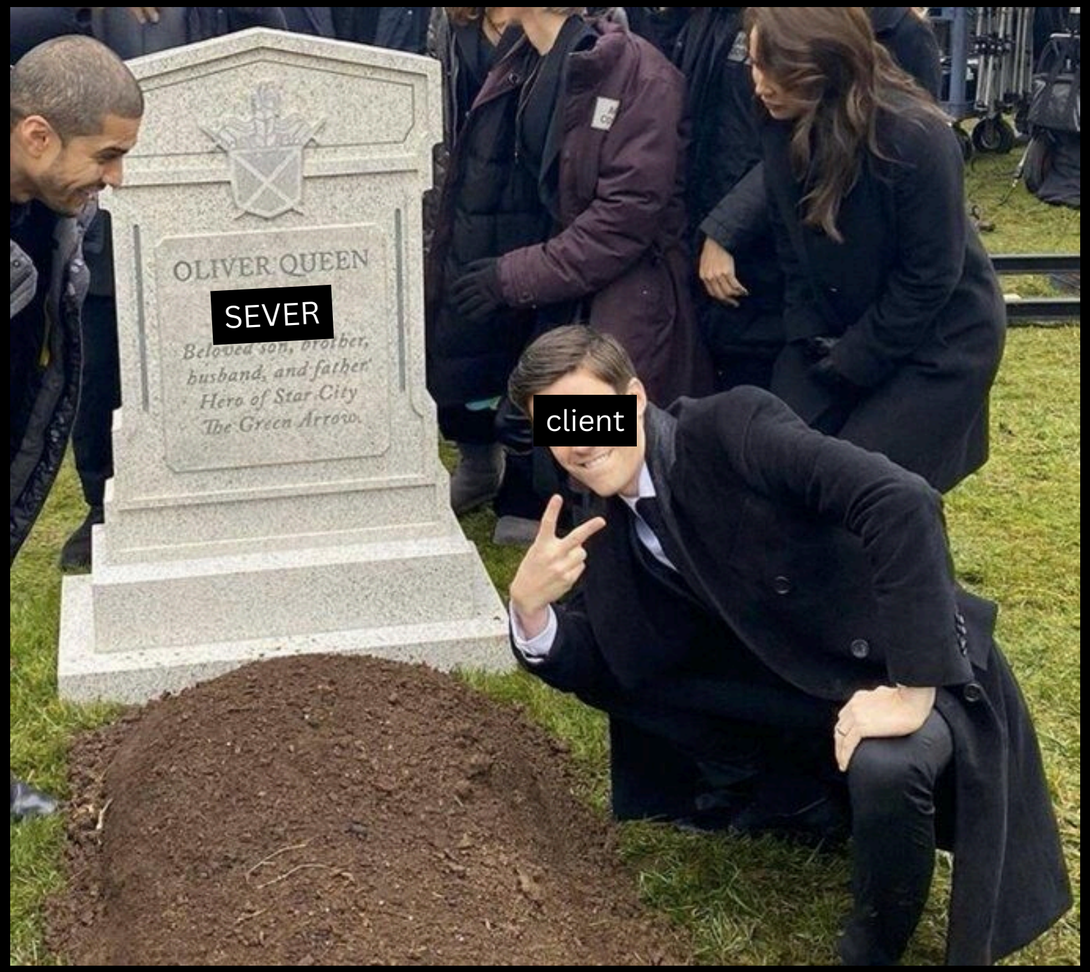
Log in

Create new account

Request new password

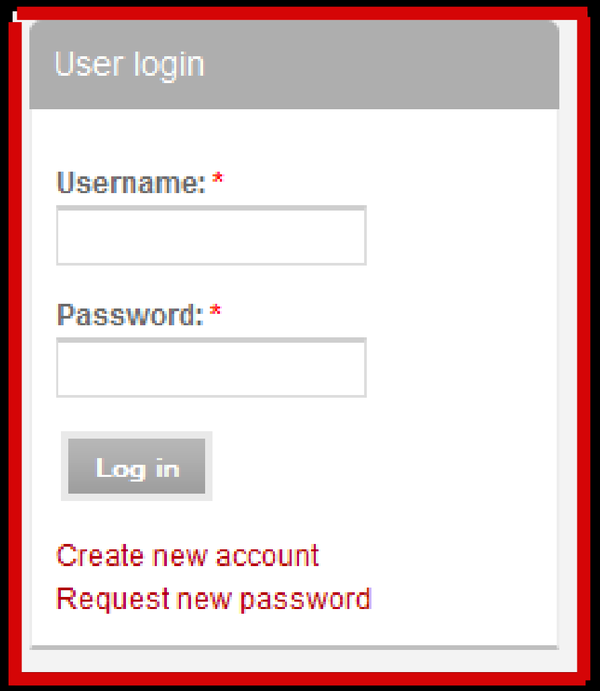
```
SELECT *
FROM users
WHERE user_id = 'admin' AND password = 'password123';
```

이런식으로 사용자가 SQL값을 ‘넣을 수‘있다.



여기서 알아 두어야 할 해킹세계 기본상식!

- 사용자의 input을 받는다.
- 값 조작이 가능하다.
- 서버의 구조가 보인다.

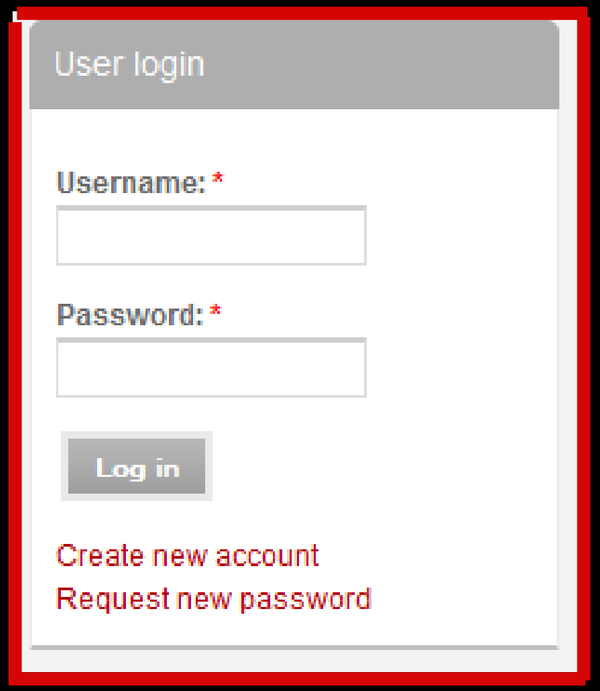



```
SELECT *
FROM users
WHERE user_id = 'admin' AND password = 'password123';
```

이런식으로 사용자가 SQL값을 ‘넣을 수‘있다.

SQL INJECT이 발생하는 매우 큰 이유 : {시험범위다 이거 밑줄쳐라}

- 쿼터로 닫는다는 거임



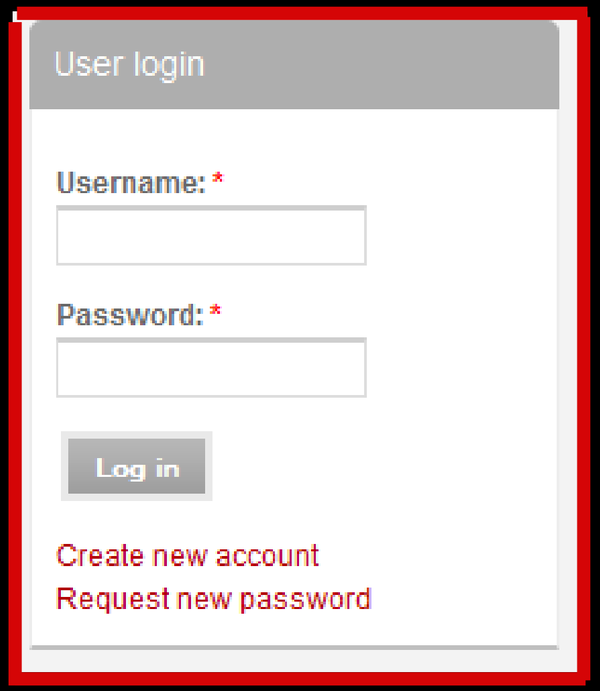

```
SELECT *
FROM users
WHERE user_id = 'admin' AND password = 'password123';
```

이런식으로 사용자가 SQL값을 ‘넣을 수‘있다.

SQL INJECT이 발생하는 매우 큰 이유 : {시험범위다 이거 밑줄쳐라}

- 쿼터로 닫는다는 거임

쿼터? ‘ ← 이거
따옴표임 쿼터. 두개있으면 뭐야




```
SELECT *
FROM users
WHERE user_id = 'admin' AND password = 'password123';
```

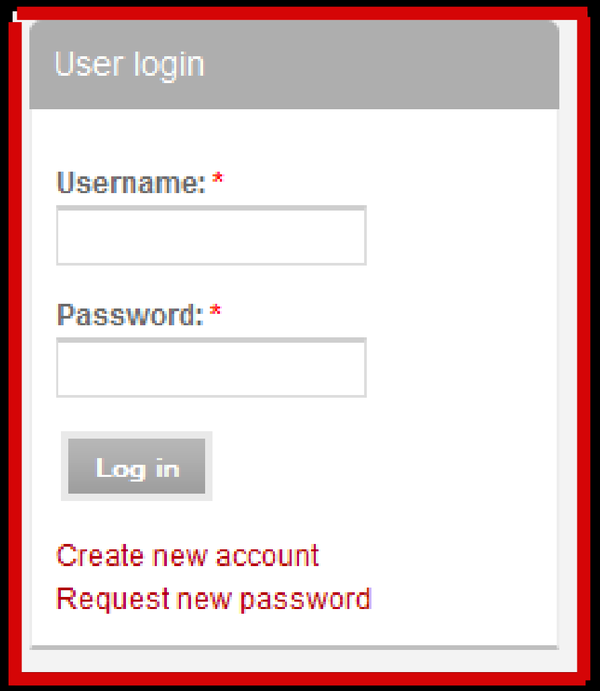
이런식으로 사용자가 SQL값을 ‘넣을 수‘있다.

SQL INJECT이 발생하는 매우 큰 이유 : {시험범위다 이거 밑줄쳐라}

- 쿼터로 닫는다는 거임

쿼터? ‘ ← 이거
따옴표임 쿼터. 두개있으면 뭐야

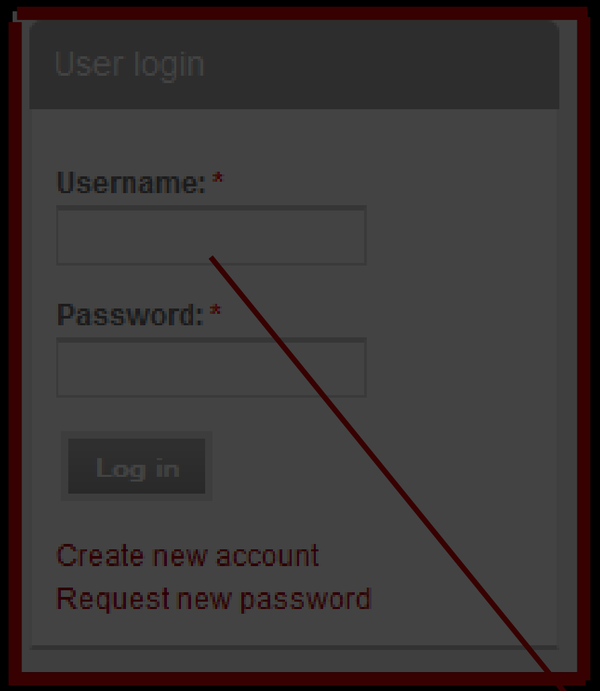
더블 쿼터 ㅇㅇ “




```
SELECT *
FROM users
WHERE user_id = 'admin' AND password = 'password123';
```

이런식으로 사용자가 SQL값을 ‘넣을 수‘있다.

```
SELECT *
FROM users
WHERE user_id = 'admin' AND password = 'password123';
```

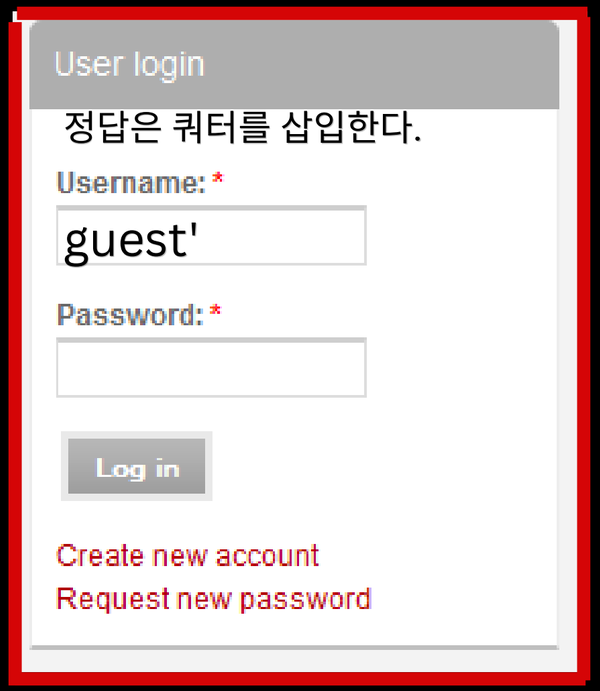



```
SELECT *
FROM users
WHERE user_id = 'admin' AND password = 'password123';
```

이런식으로 사용자가 SQL값을 ‘넣을 수‘있다.

어떻게 하면?
창의적으로 나쁜짓을 할 수 있을까요? (어려움)

```
SELECT *
FROM users
WHERE user_id = 'admin' AND password = 'password123';
```

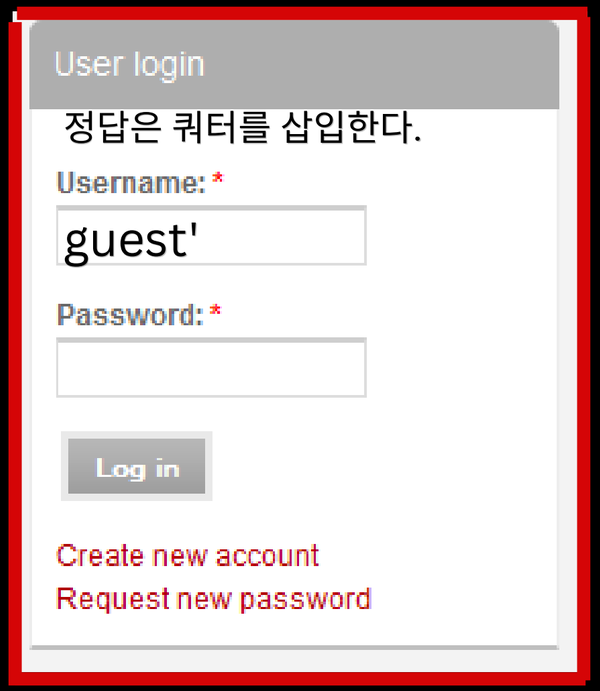



```
SELECT *
FROM users
WHERE user_id = 'admin' AND password = 'password123';
```

이런식으로 사용자가 SQL값을 '넣을 수'있다.

어떻게 하면?
창의적으로 나쁜짓을 할 수 있을까요? (어려움)

```
SELECT *
FROM users
WHERE user_id = 'admin' AND password = 'password123';
```

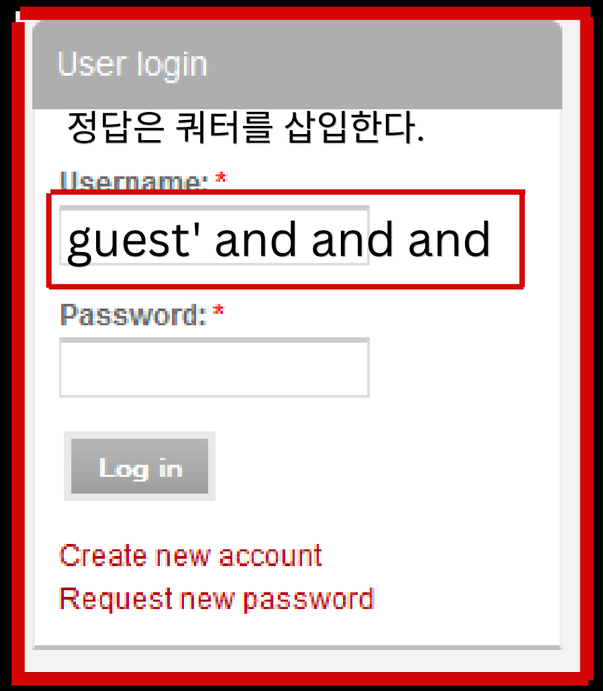



```
SELECT *
FROM users
WHERE user_id = 'admin' AND password = 'password123';
```

이런식으로 사용자가 SQL값을 ‘넣을 수‘있다.

```
SELECT *
FROM users
WHERE user_id = 'guest' ' AND password = 'password123';
```

자 틈이 생기죠
맞아요 이게 됩니다

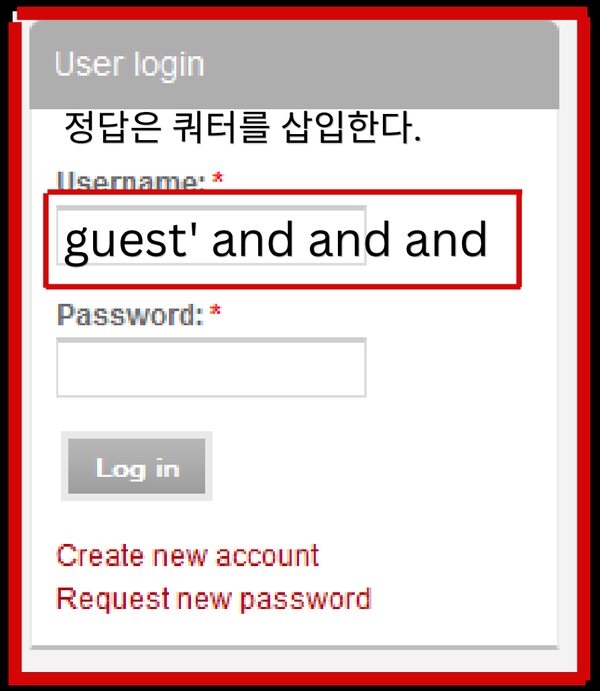



```
SELECT *
FROM users
WHERE user_id = 'admin' AND password = 'password123';
```

이런식으로 사용자가 SQL값을 ‘넣을 수‘있다.

```
SELECT *
FROM users
WHERE user_id = 'guest' ' AND password = 'password123';
```

자 틈이 생기죠
맞아요 이게 됩니다




```
SELECT *
FROM users
WHERE user_id = 'admin' AND password = 'password123';
```

이런식으로 사용자가 SQL값을 ‘넣을 수‘있다.

```
SELECT *
FROM users
WHERE user_id = 'guest' and and and ' AND password =
'password123';
```

놀랍게도 에러가 납니다

자 틈이 생기죠
맞아요 이게 됩니다

User login

정답은 쿼터를 삽입한다.

Username: *

guest' and and and

Password: *

Log in

Create new account

Request new password

```
SELECT *
FROM users
WHERE user_id = 'admin' AND passwo
```

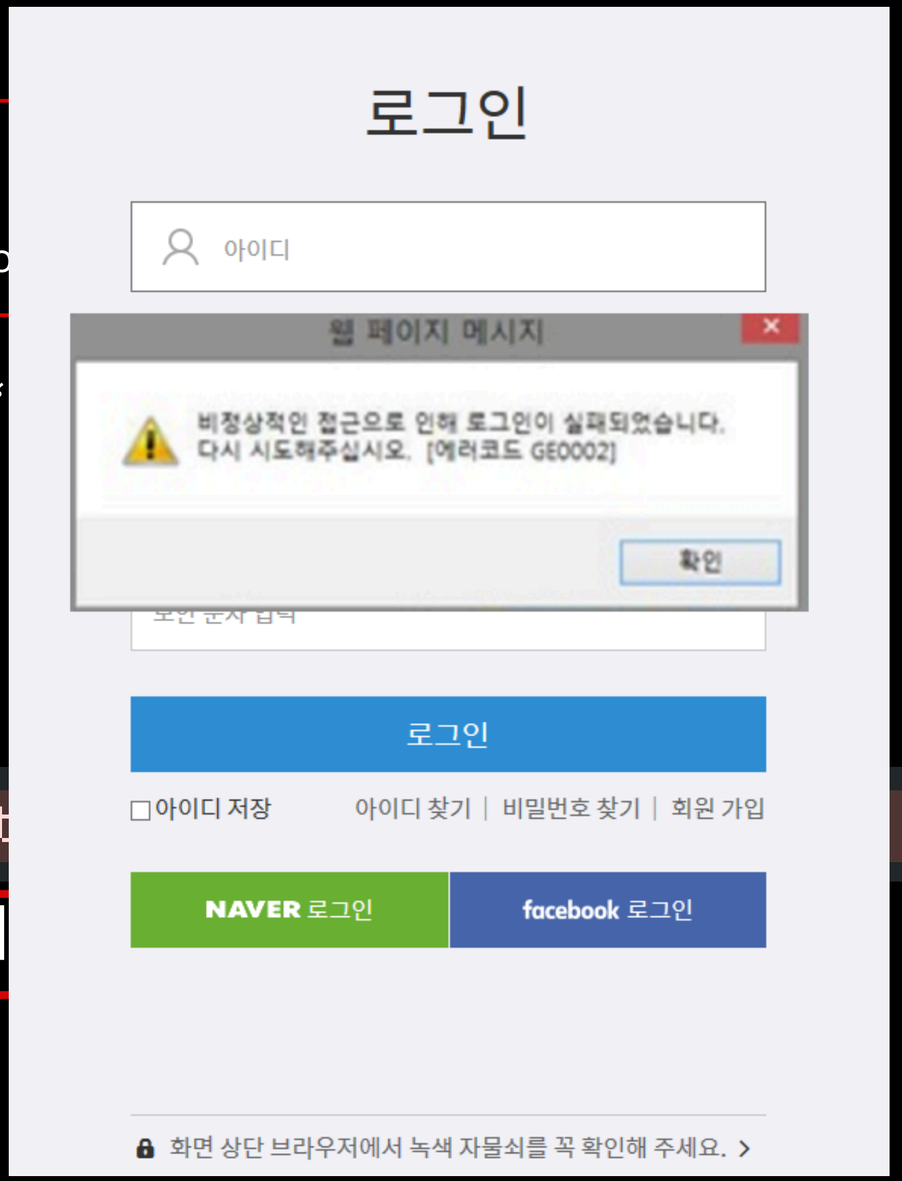
이런식으로 사용자가 SQL값을 '

```
SELECT *
FROM users
WHERE user_id = 'guest' and and
'password123';
```

Uncaught Synt

자 틈이 생기죠

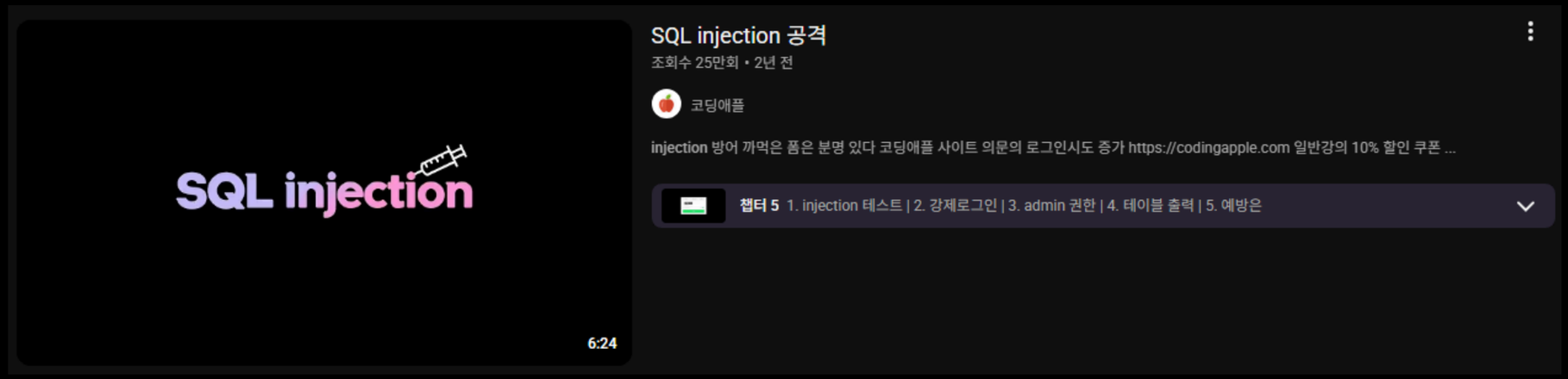
맞아요 이게 됩니다



해킹적은 관점에서 이건 굉장한 취약점임

만약 한 웹사이트에서 SQL인젝션이 터진다?
→ 레전드 사건임

일단 여기까지만 알려주고
영상보고 오겠습니다.



실습 :

CROSS SITE SCRIPT

XSS

SQL injection과 함께 웹 상에서 가장 기초적인 취약점 공격 방법의 일종으로, 악의적인 사용자가 공격하려는 사이트에 스크립트를 넣는 기법을 말한다. 공격에 성공하면 사이트에 접속한 사용자는 삽입된 코드를 실행하게 되며,[2] 보통 의도치 않은 행동을 수행시키거나 쿠키나 세션 토큰 등의 민감한 정보를 탈취한다.

크로스 사이트 스크립팅이란 이름 답게, 자바스크립트를 사용하여 공격하는 경우가 많다. 공격 방법이 단순하고 가장 기초적이지만, 많은 웹사이트들이 XSS에 대한 방어 조치를 해두지 않아 공격을 받는 경우가 많다. 여러 사용자가 접근 가능한 게시판 등에 코드를 삽입하는 경우도 많으며, 경우에 따라서는 메일과 같은 매체를 통해서도 전파된다. 심지어는 닉네임에 코드를 심기도 한다.

SQL injection과 함께 웹 상에서 가장 기초적인 취약점 공격 방법의 일종으로, 악의적인 사용자가 공격하려는 사이트에 스크립트를 넣는 기법을 말한다. 공격에 성공하면 사이트에 접속한 사용자는 삽입된 코드를 실행하게 되며,[2] 보통 의도치 않은 행동을 수행시키거나 쿠키나 세션 토큰 등의 민감한 정보를 탈취한다.

크로스 사이트 스크립팅이란 이름 답게, 자바스크립트를 사용하여 공격하는 경우가 많다. 공격 방법이 단순하고 가장 기초적이지만, 많은 웹사이트들이 XSS에 대한 방어 조치를 해두지 않아 공격을 받는 경우가 많다. 여러 사용자가 접근 가능한 게시판 등에 코드를 삽입하는 경우도 많으며, 경우에 따라서는 메일과 같은 매체를 통해서도 전파된다. 심지어는 닉네임에 코드를 심기도 한다.

자 이것만 읽어도 대충 뭔지 감이 오죠

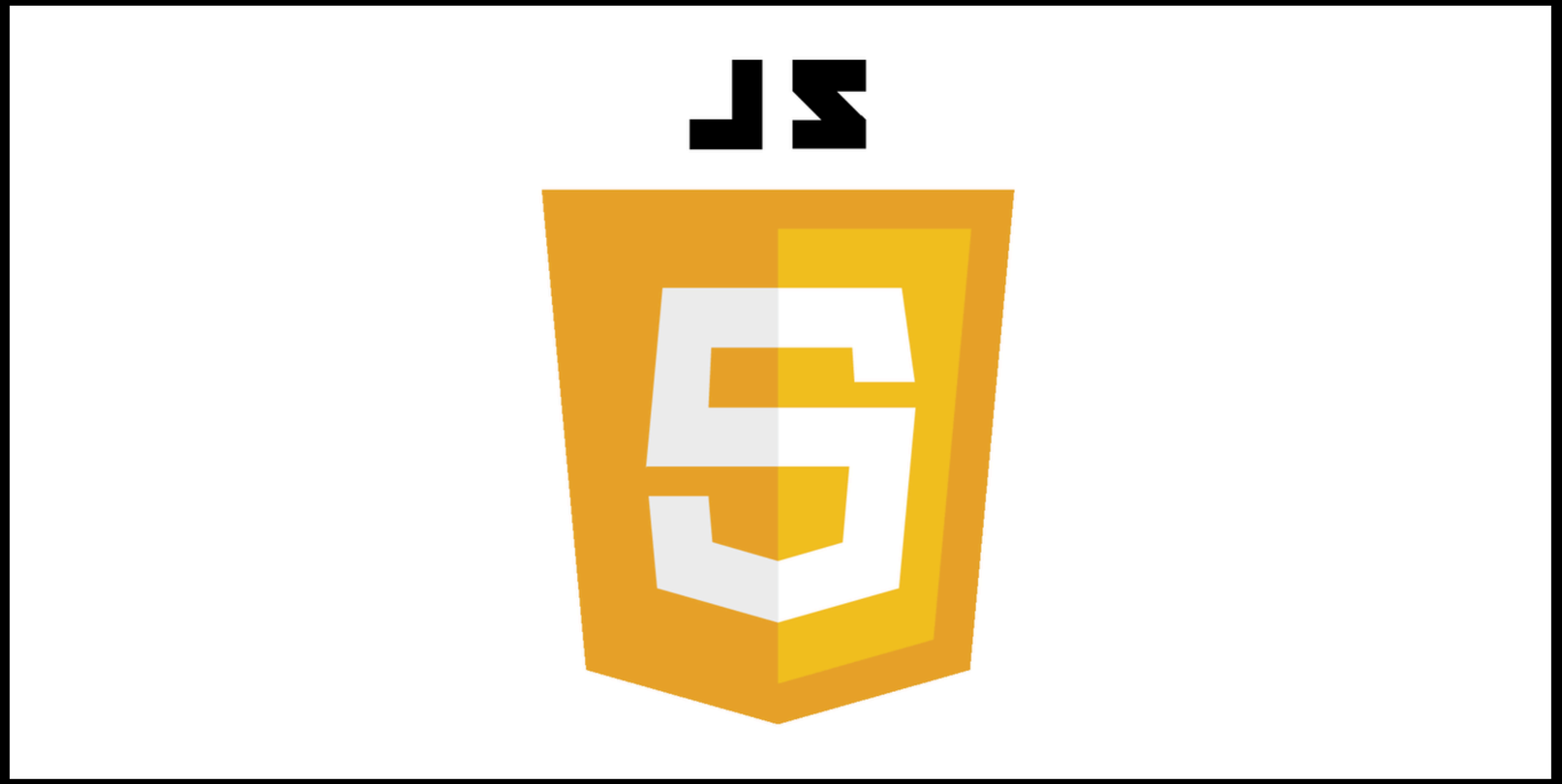
SQL injection과 함께 웹 상에서 가장 기초적인 취약점 공격 방법의 일종으로, 악의적인 사용자가 공격하려는 사이트에 스크립트를 넣는 기법을 말한다. 공격에 성공하면 사이트에 접속한 사용자는 삽입된 코드를 실행하게 되며,[2] 보통 의도치 않은 행동을 수행시키거나 쿠키나 세션 토큰 등의 민감한 정보를 탈취한다.

크로스 사이트 스크립팅이란 이름 답게, 자바스크립트를 사용하여 공격하는 경우가 많다. 공격 방법이 단순하고 가장 기초적이지만, 많은 웹사이트들이 XSS에 대한 방어 조치를 해두지 않아 공격을 받는 경우가 많다. 여러 사용자가 접근 가능한 게시판 등에 코드를 삽입하는 경우도 많으며, 경우에 따라서는 메일과 같은 매체를 통해서도 전파된다. 심지어는 닉네임에 코드를 심기도 한다.

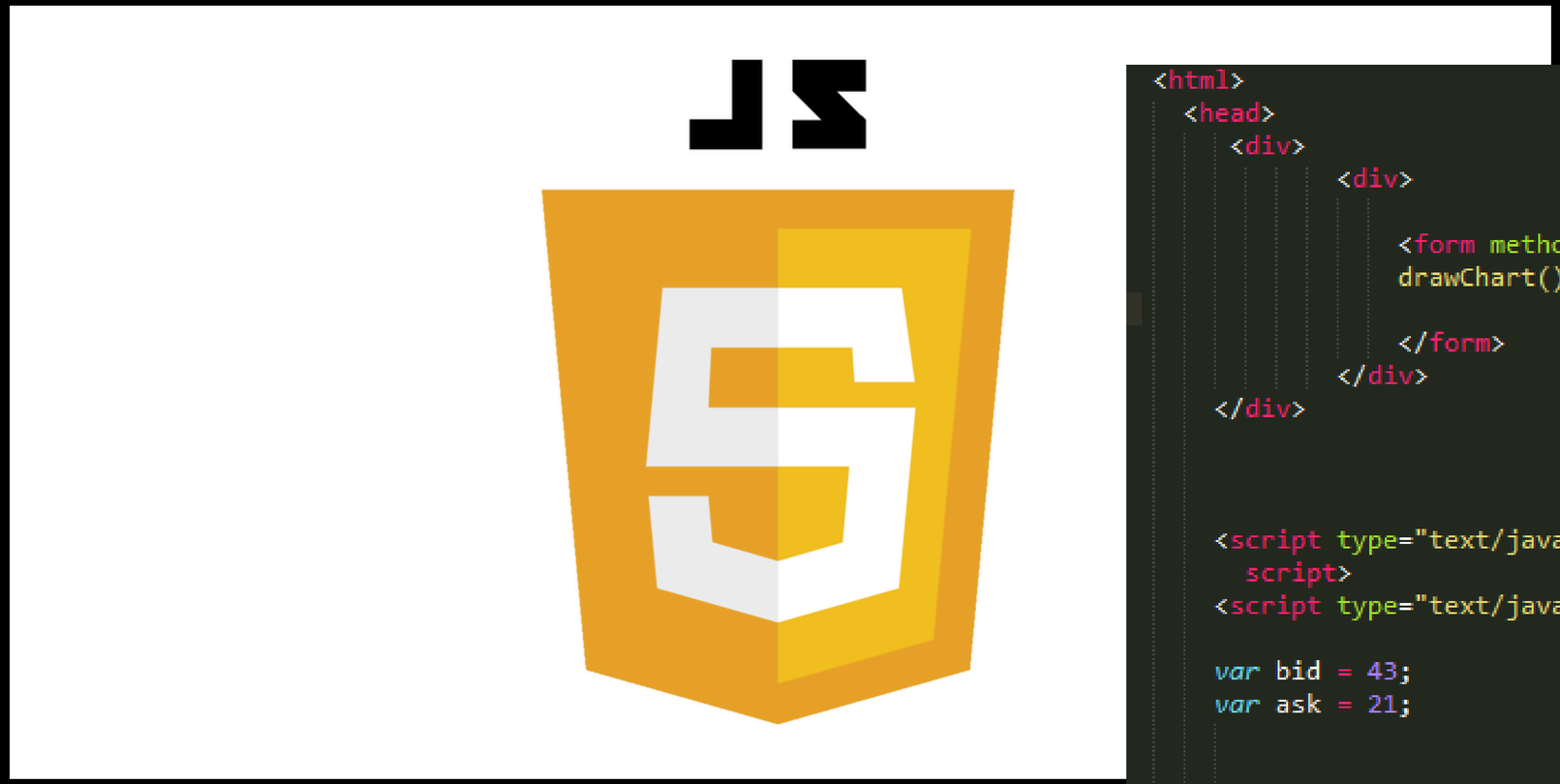
자 이것만 읽어도 대충 뭔지 감이 오죠

일단 공격 방식은 기본적으로 JAVAscript를 사용하고
가장 기초적인 취약점임

자바 스크립트. 이걸 알아야됨

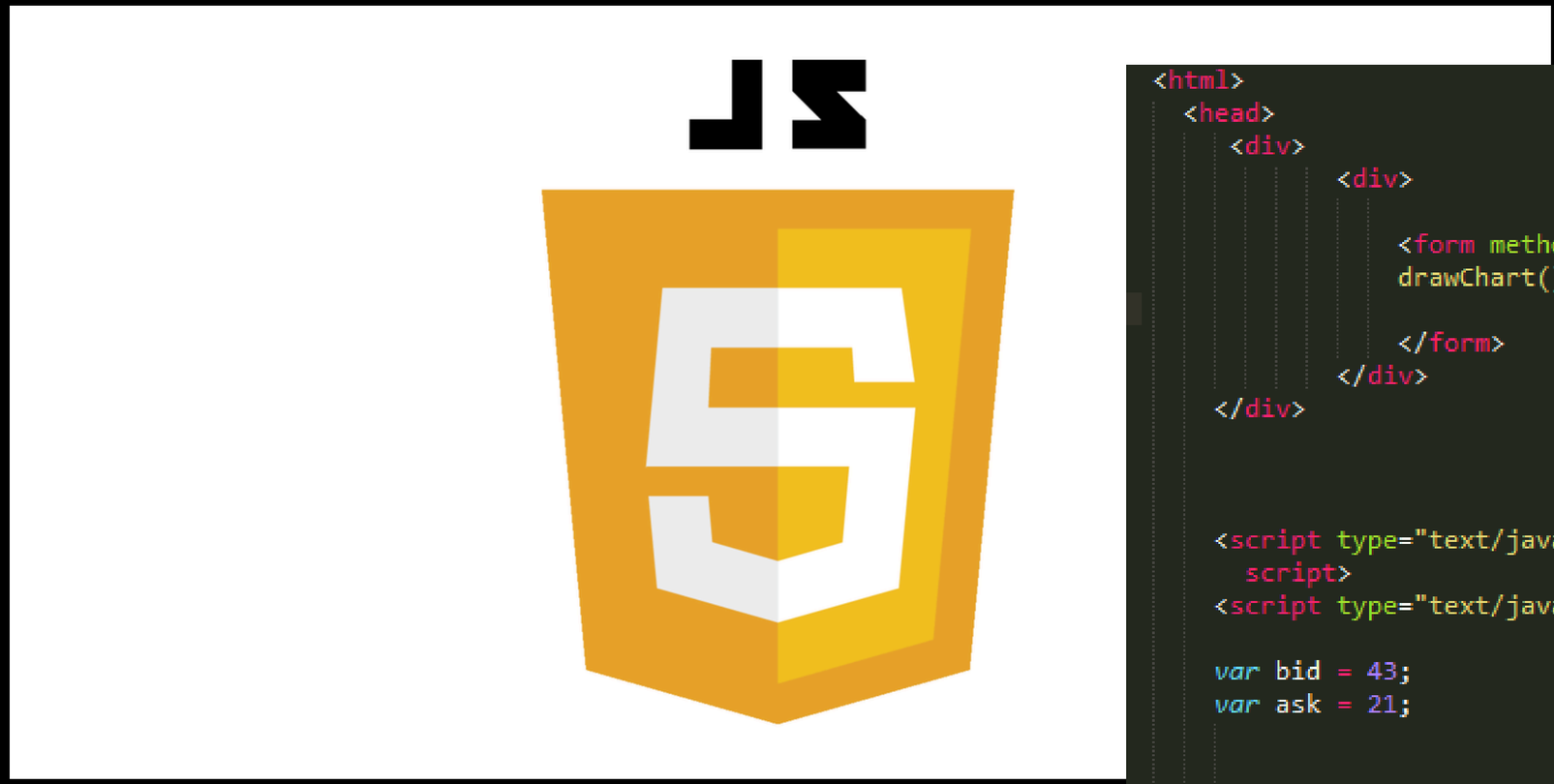


자바 스크립트. 이걸 알아야됨



```
<html>
  <head>
    <div>
          <div>

              <form method="post" action="#" id="formvalue" onkeyup="
              drawChart()" />

              </form>
          </div>
    </div>



    <script type="text/javascript" src="https://www.google.com/jsapi"></
      script>
    <script type="text/javascript">

    var bid = 43;
    var ask = 21;


      google.load("visualization", "1", {packages:["corechart"]});
      google.setOnLoadCallback(drawChart);
      function drawChart() {
        var data = google.visualization.arrayToDataTable([
          ['Price', 'Quantity'],
          ['Value #1', bid],
          ['Value #2', ask],
        ]);
```

자바 스크립트. 이걸 알아야됨



```
<html>
  <head>
    <div>
          <div>

              <form method="post" action="#" id="formvalue" onkeyup="
              drawChart()" />

              </form>
          </div>
    </div>



    <script type="text/javascript" src="https://www.google.com/jsapi"></
      script>
    <script type="text/javascript">

    var bid = 43;
    var ask = 21;


      google.load("visualization", "1", {packages:["corechart"]});
      google.setOnLoadCallback(drawChart);
      function drawChart() {
        var data = google.visualization.arrayToDataTable([
          ['Price', 'Quantity'],
          ['Value #1', bid],
          ['Value #2', ask],
        ]);
```

이거 근데 걍 프로그래밍언어라
걍 알아야됨. 내가 알려줄순없음

그래서 보통 HTML을 쓰는 게시판에 많이 발생하는데

그래서 보통 HTML을 쓰는 게시판에 많이 발생하는데

그래서 보통 HTML을 쓰는 게시판에 많이 발생하는데

HTML(HyperText Markup Language)은
웹 페이지의 구조와 내용을 정의하는 마크업 언어

그래서 보통 HTML을 쓰는 게시판에 많이 발생하는데

그래서 보통 HTML을 쓰는 게시판에 많이 발생하는데

HTML(HyperText Markup Language)은
웹 페이지의 구조와 내용을 정의하는 마크업 언어

<< 꺾쇠 >> 사용함 : 마크업 언어

HTML 한가지 예를들 어보자면

HTML 한가지 예를들 어보자면

<input type=“button” > 버튼 </input>

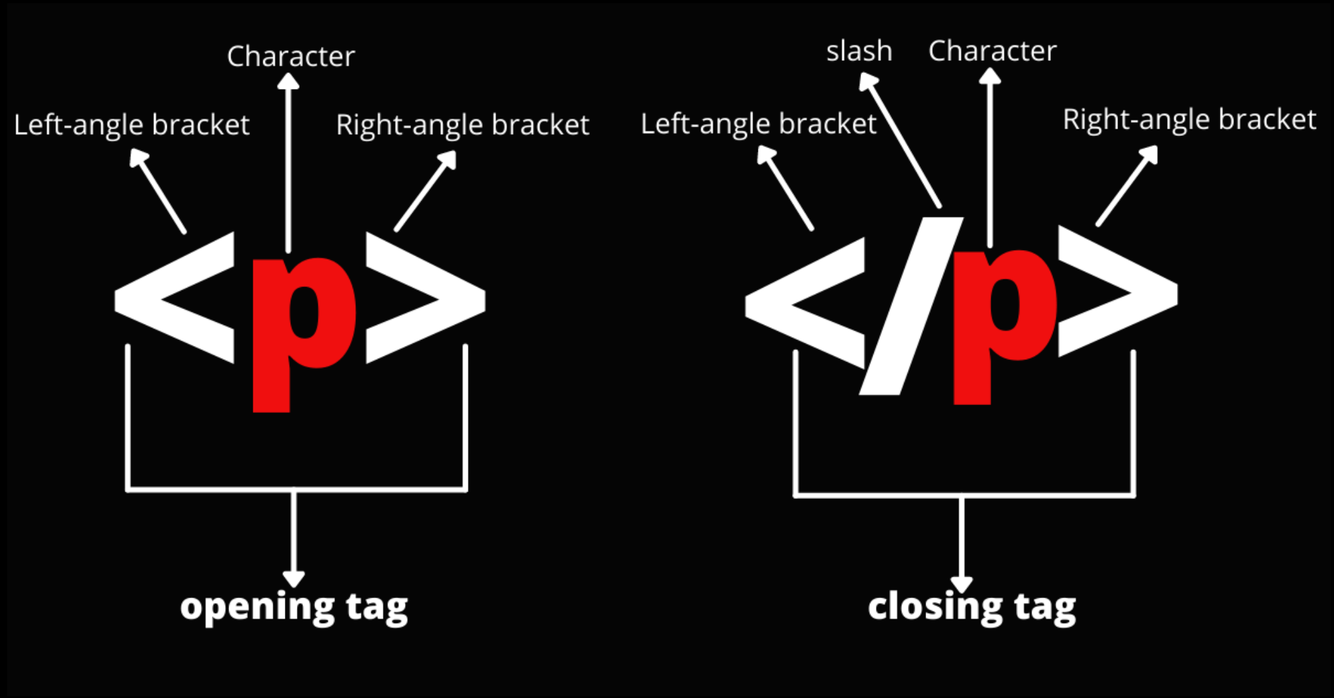
이게 버튼을 만드는 코드임

HTML 한가지 예를들 어보자면

태그

<input type=“button” > 버튼 </input>

이게 버튼을 만드는 코드임

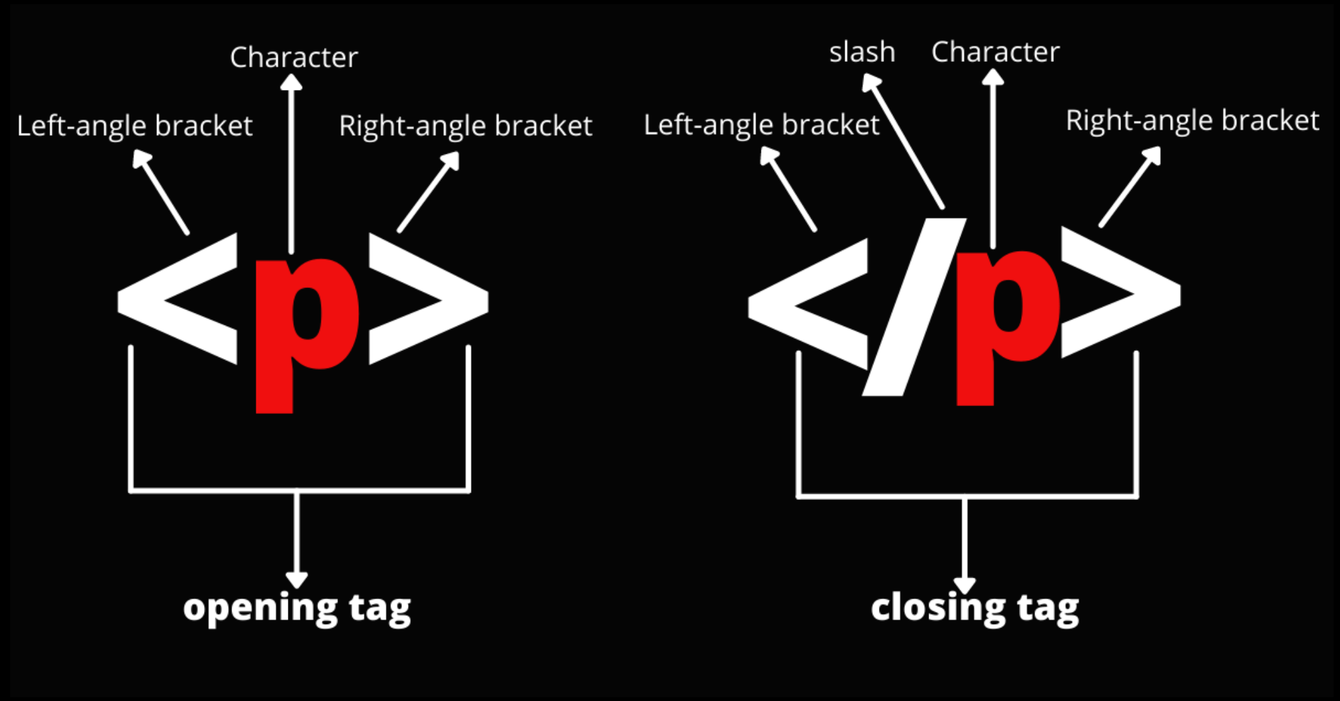


HTML 한가지 예를들 어보자면

속성

<input type="button" > 버튼 </input>

이게 버튼을 만드는 코드임

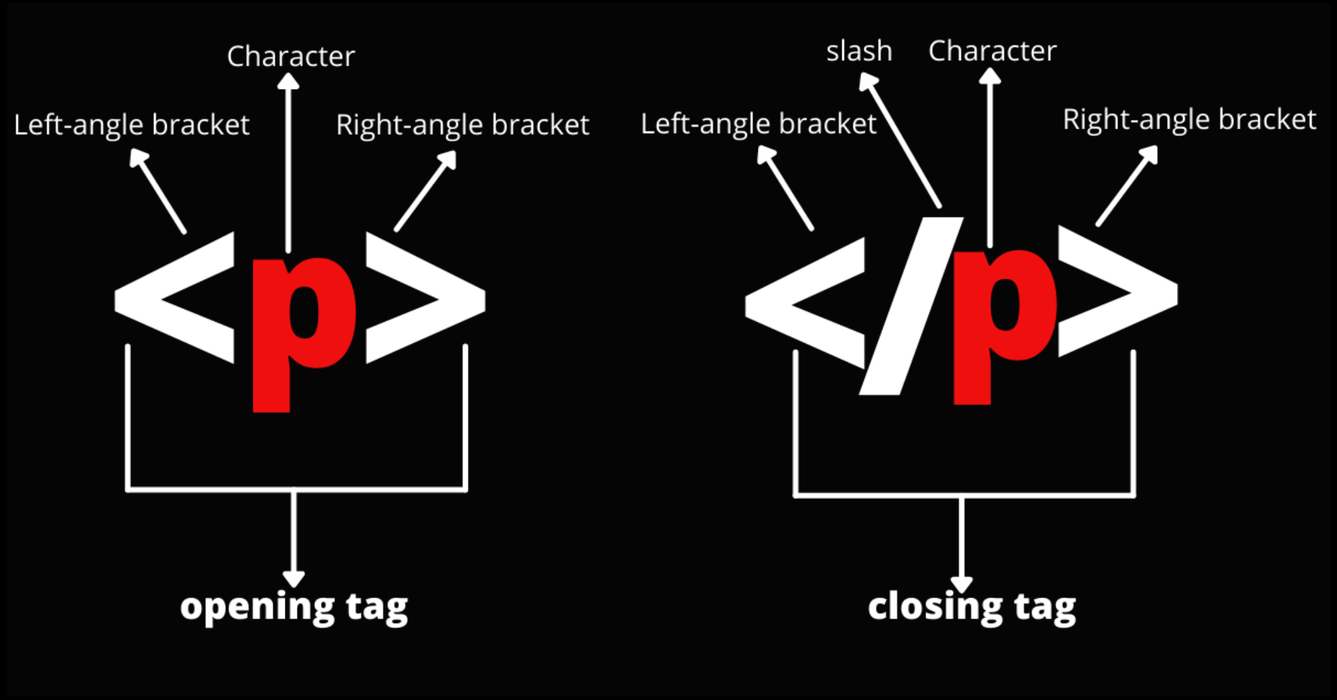


HTML 한가지 예를들 어보자면

텍스트

<input type=“button” > 버튼 </input>

이게 버튼을 만드는 코드임

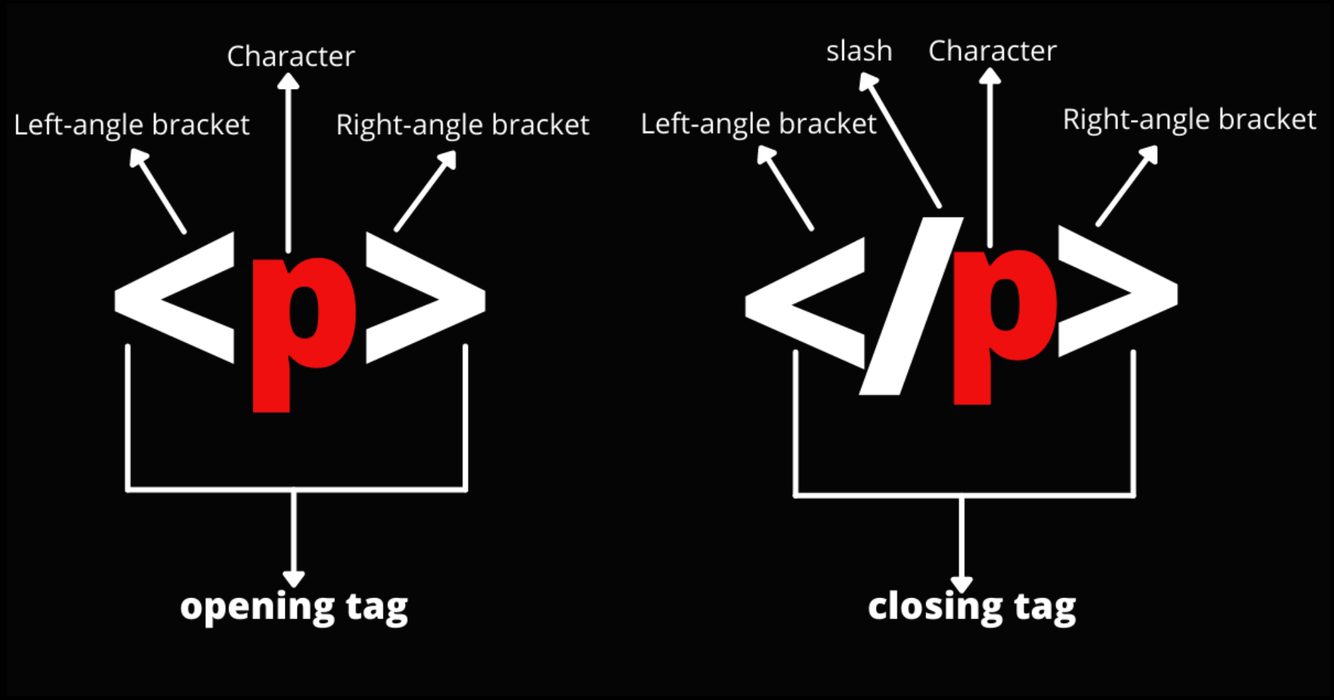


HTML 한가지 예를들 어보자면

태그 끝

<input type=“button” > 버튼 </input>

이게 버튼을 만드는 코드임



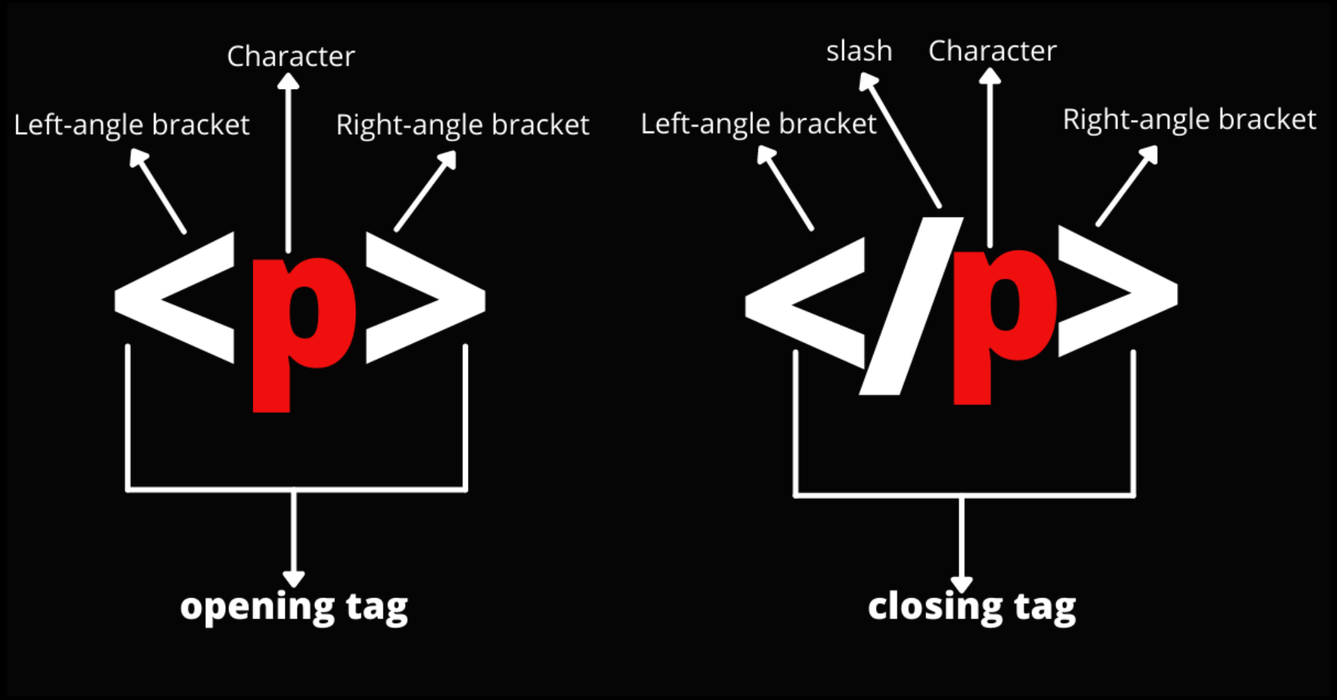
HTML 한가지 예를들 어보자면

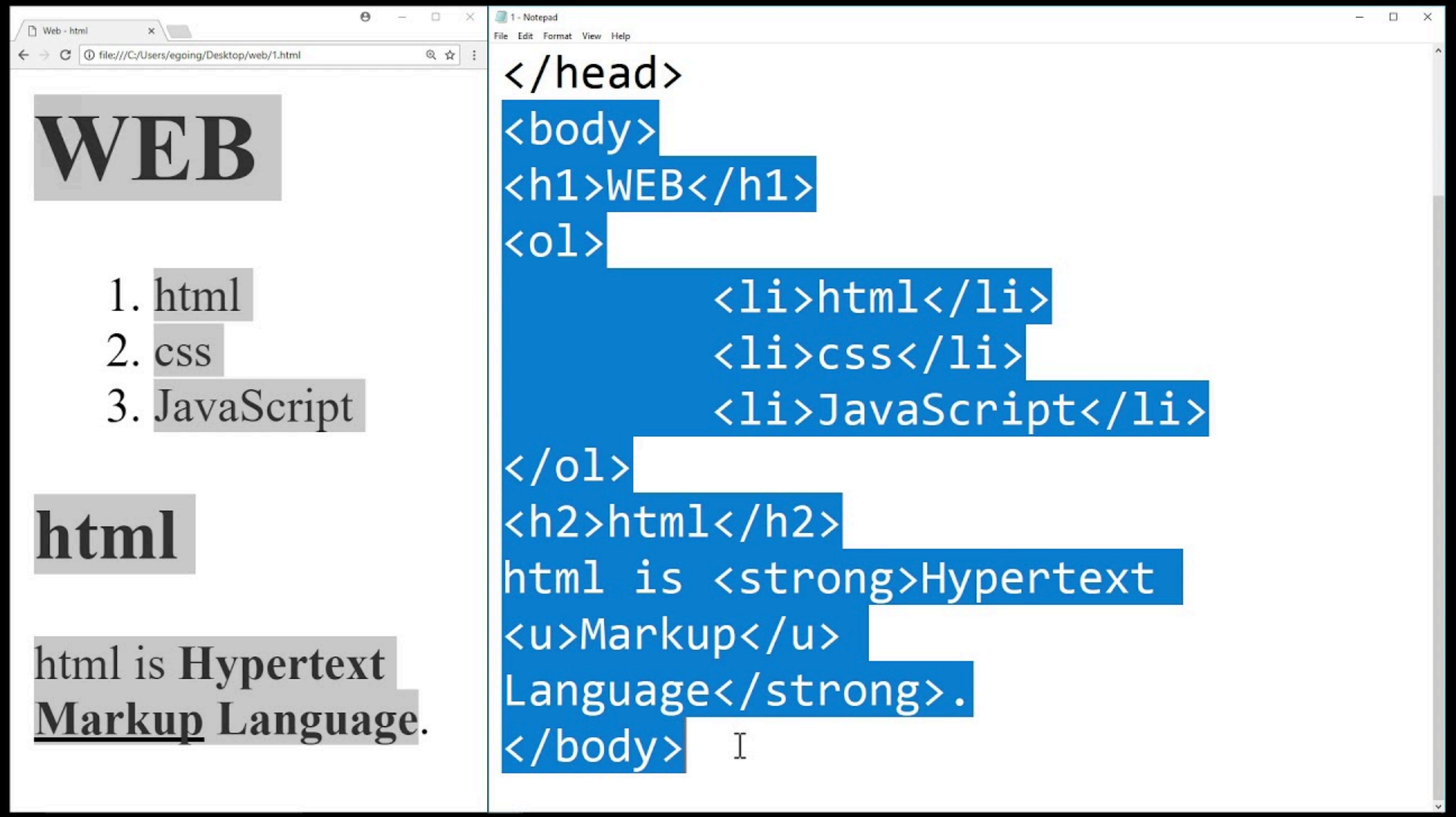
태그 끝

<input type=“button” > 버튼 </input>

이게 버튼을 만드는 코드임

쉽죠?



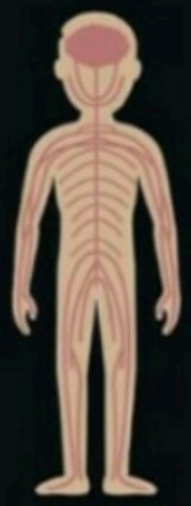


이런식으로 태그 여러개 써서 페이지 만드는 거임 ㅇㅇ


JAVA <SCRIPT>

HTML
The Skeleton

CSS
The Skin



JavaScript
The Brains

흔히 script는 '동작'으로 많이 불림
: 실제 값을 변/위조/조작/실행/외부통신
이런거 함

그 HTML 태그들중에 <script>라는 태그가 있음

이걸로 XSS가 발동되는거

이걸로 XSS가 발동되는거

- XSS 종류는 저번에 말했던것 처럼 여러 종류와
  - 파생된 기법들이 있는데
    - 생략하겠다.

그 HTML 태그들중에 <script>라는 태그가 있음

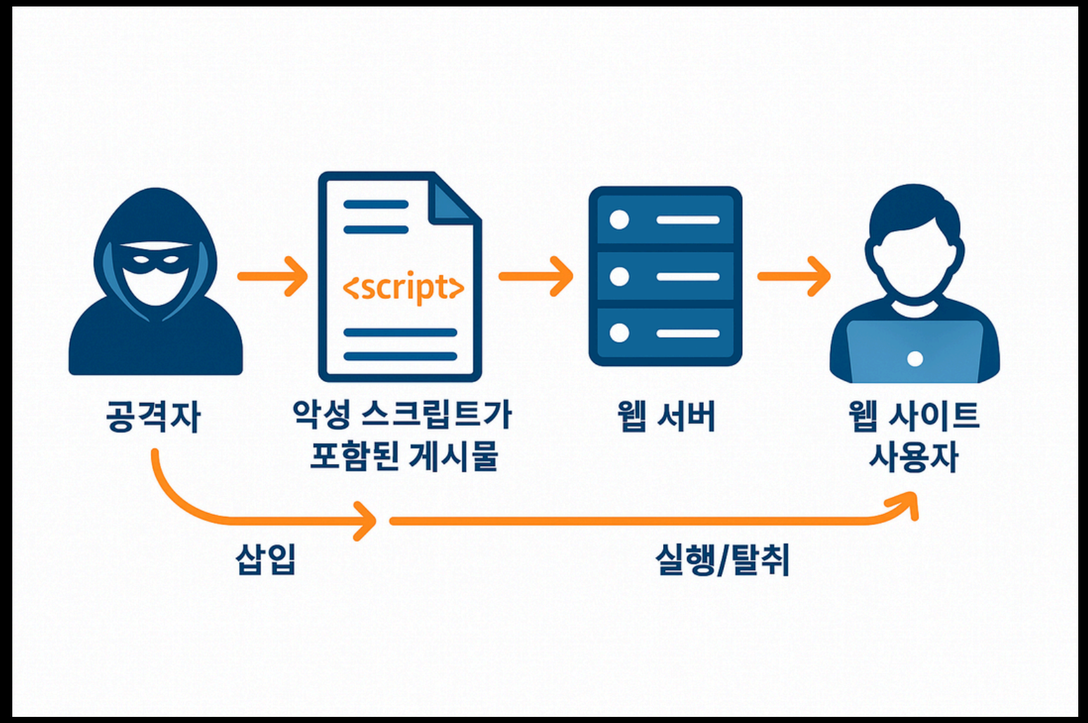
이걸로 XSS가 발동되는거

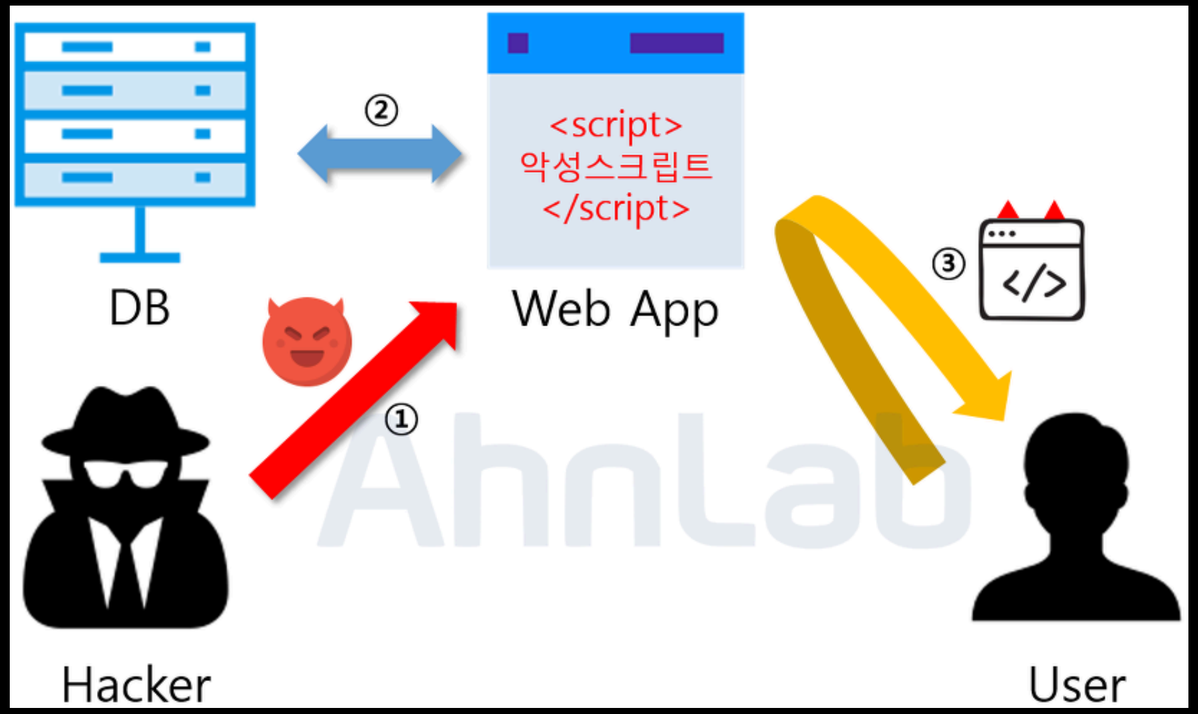
```
<script>
  console.log('Hello, World!');
  alert('페이지가 로드되었습니다.');
</script>
```

이걸로 XSS가 발동되는거

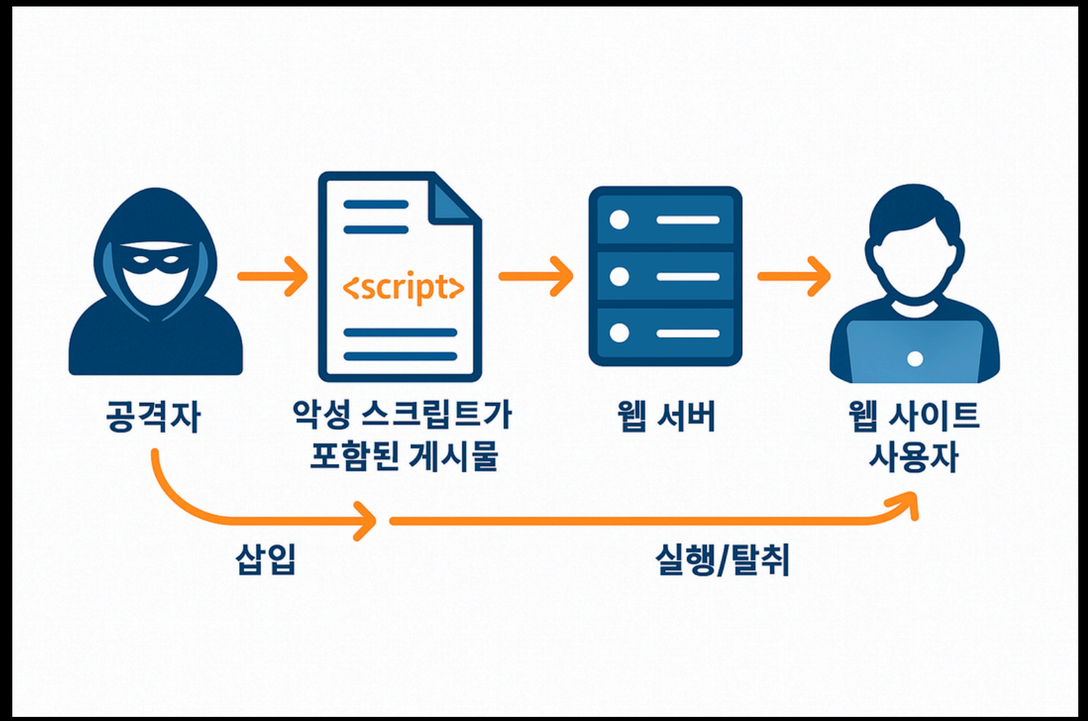
- XSS 종류는 저번에 말했던것 처럼 여러 종류와
  - 파생된 기법들이 있는데
    - 생략하겠다.

그 HTML 태그들중에 <script>라는 태그가 있음



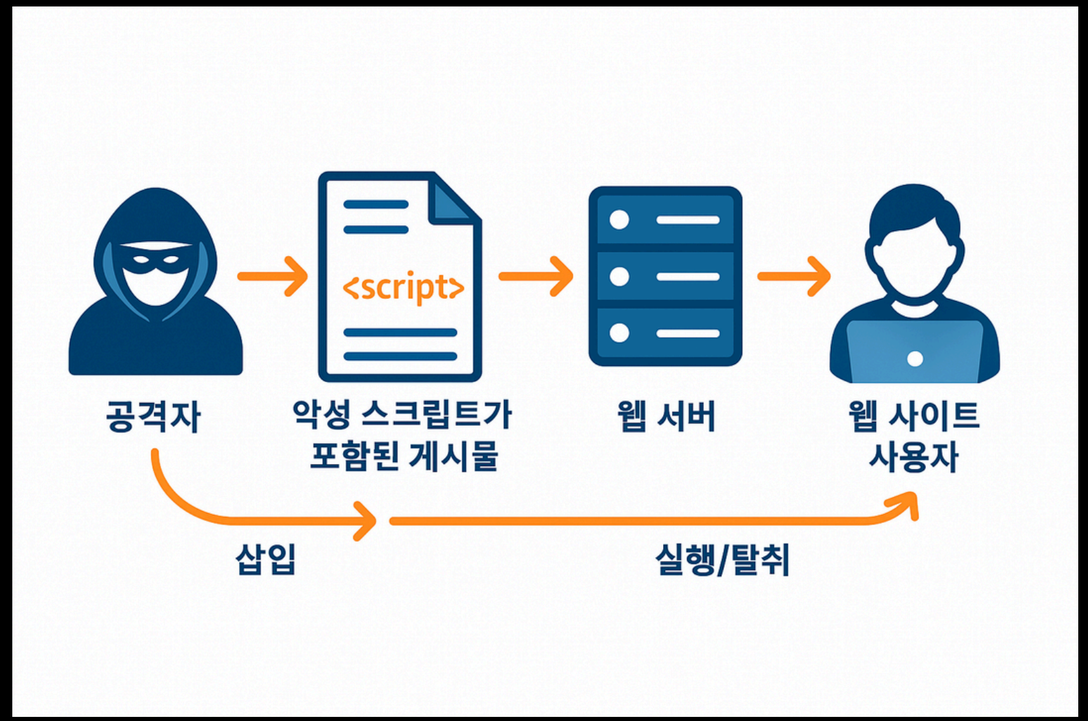


그 HTML 태그들중에 <script>라는 태그가 있음



근데 이 script 태그를 쓰려면
이게 작동하는지 알아야 하잖슴

그 HTML 태그들중에 <script>라는 태그가 있음



근데 이 script 태그를 쓰려면
이게 작동하는지 알아야 하잖슴

그래서 쓰는게 :
<script>alert(1);<script>

실습 :

https://xss-game.appspot.com/


OLIVER QUEEN
1985-2019
Beloved son, brother,
husband, and father
Hero of Star City
The Green Arrow

THANK YOU

끝임