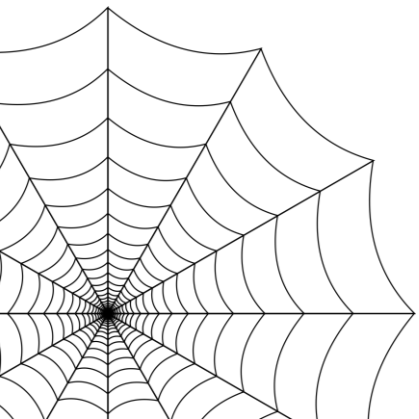
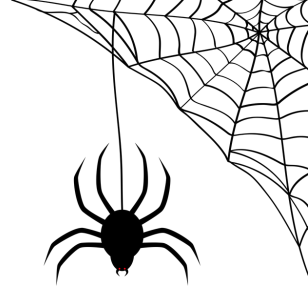


WEB Hacking

웹해킹 교육 2회차

18학번 서연주

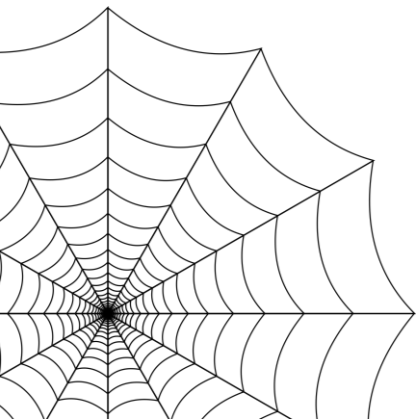


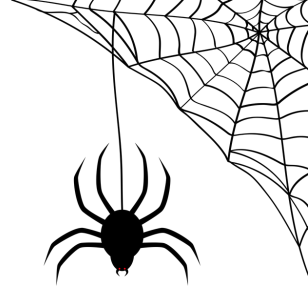


<목차>

- < 00. 복습 />
- < 01. PHP />
- < 02. PHP 심화 />
- < 03. MySQL />

</목차>





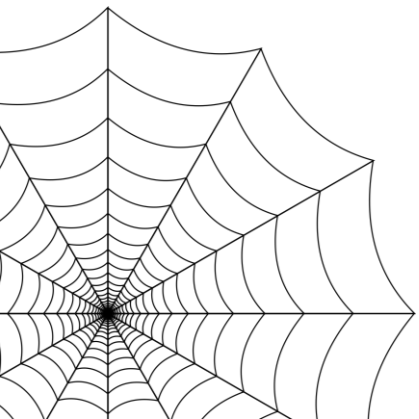
<00/>

<복습>

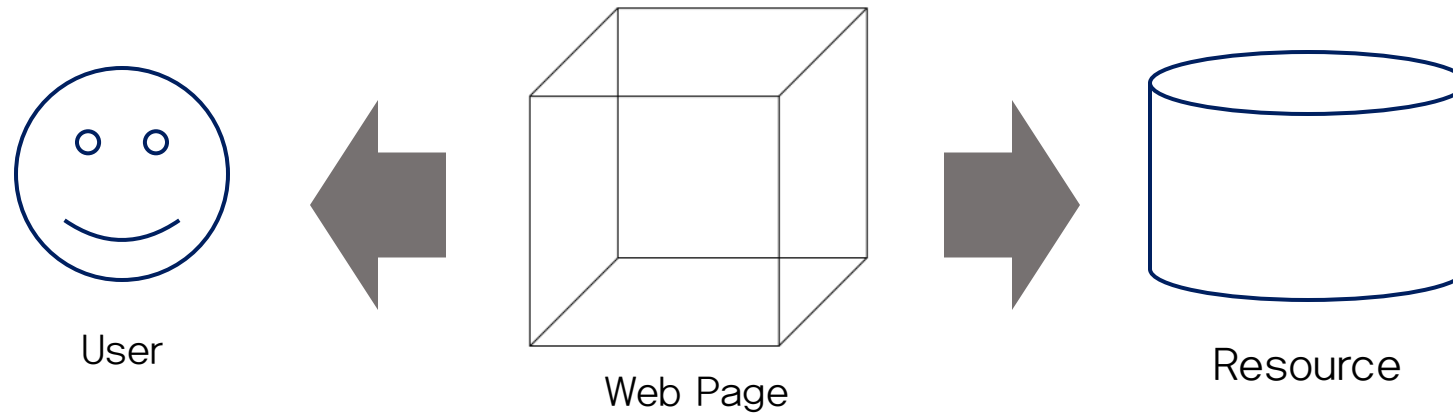
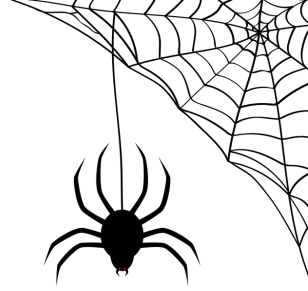
< Front End & Back End />

< HTML, CSS, JavaScript />

</복습>



00. 복습 - Front End & Back End



Front End

(Client side)
보여지는 영역

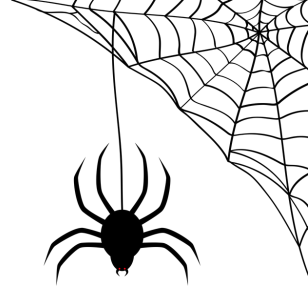
*html, css,
JavaScript*

Back End

(Server side)
보이지 않는 영역

*PHP, JSP, ASP, ASP.Net,
NodeJs, Flask*

00. 복습 - HTML, CSS, JavaScript



Front End를 구성하는 언어



<01 />

<PHP>

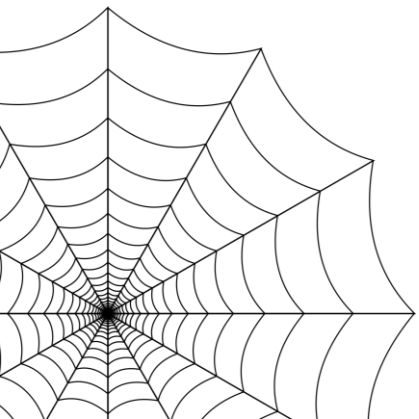
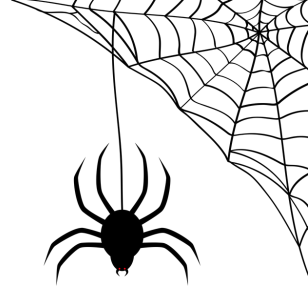
< PHP? />

< 작동 원리 />

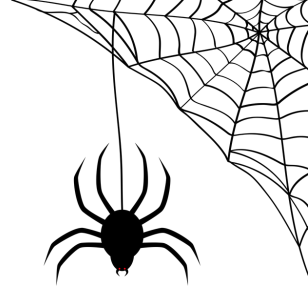
< 문법 />

< 실습 />

</PHP>



01. PHP – PHP?

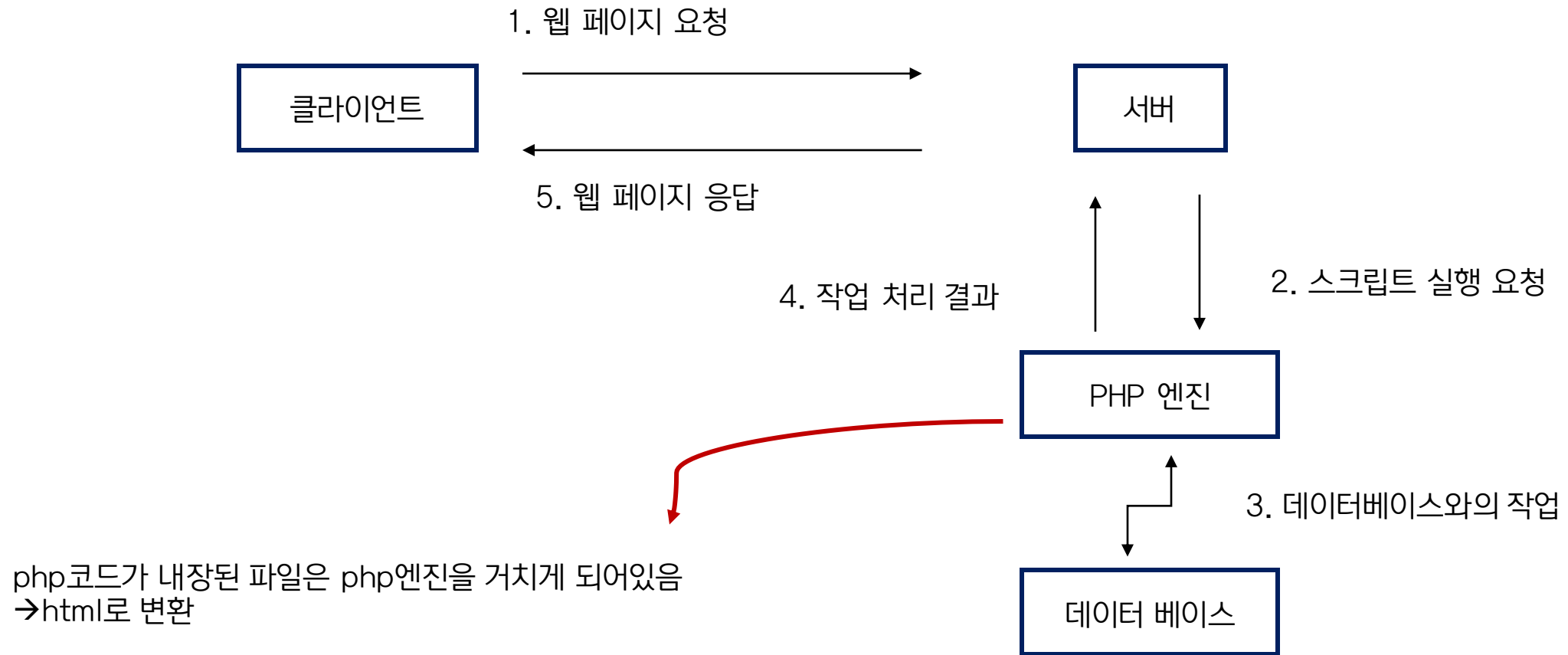
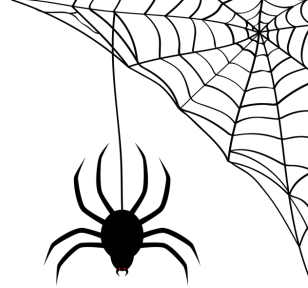


Back End를 구성하는 언어

Server-side HTML-embedded script 언어

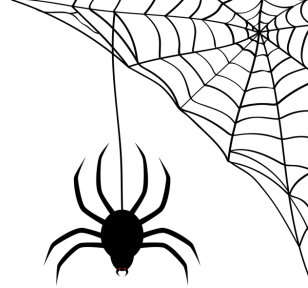
== PHP는 서버에서 실행됨,
HTML을 포함함
(Html내용을 .php로 저장해도 사용 가능)

01. PHP – 작동원리



< PHP 동작 원리 >

01. PHP – 작동원리



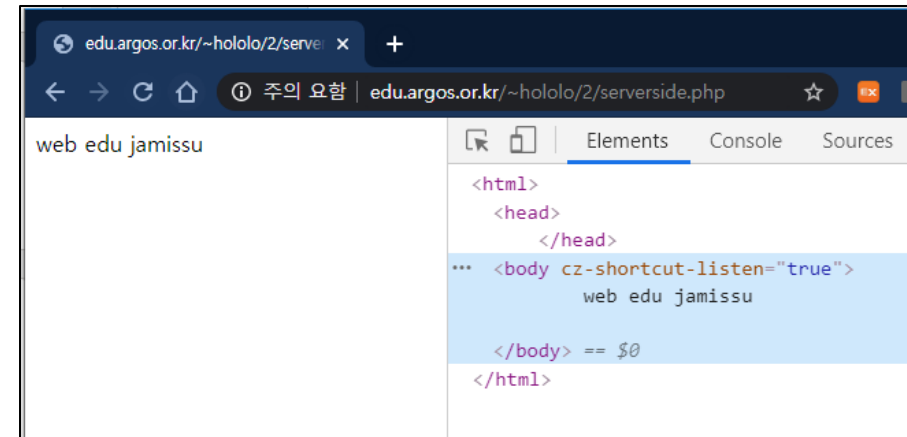
php코드가 내장된 파일은 php엔진을 거치게 되어있음
→ **html로 변환**

→ 클라이언트 측에서는 볼 수 없음

서버 사이드 언어이기 때문

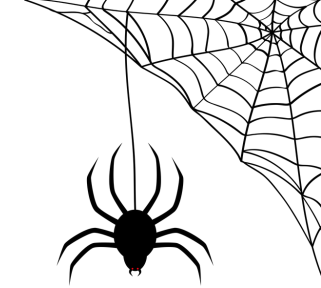
```
<html>
  <head>
  </head>
  <body>
    <?php      echo "web edu jamissu";
    ?>
  </body>
</html>
```

서버측에서 작성



클라이언트 측에서는 볼 수 없음

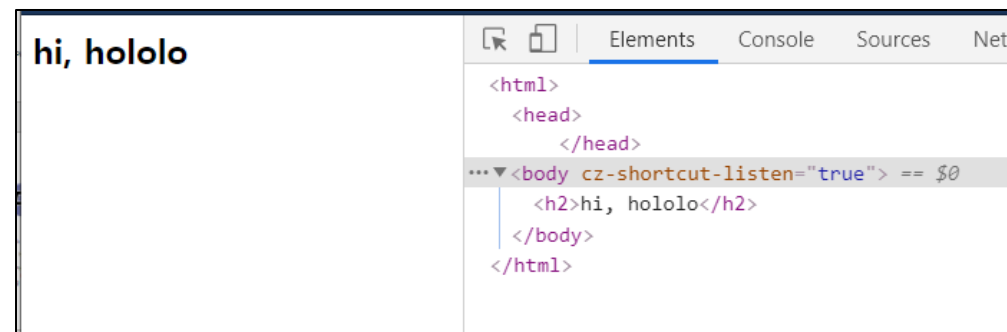
01. PHP – 문법



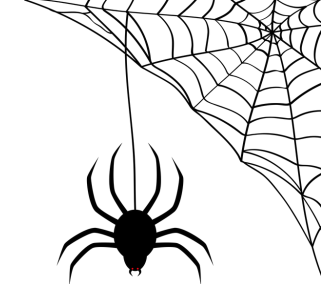
1. PHP의 시작과 끝은 `<?php ~~~ ?>` 또는 `<? ~~~ ?>`
2. Html을 PHP파일 안에 표현할 때는 echo “(html 태그/내용)”;
즉, echo 안에 표현
3. 문장의 끝은 “ ; “
4. 변수는 타입 필요 X
문자열과 함께 사용할 때는 {}안에 넣어서 (지향)
5. 주석은 // 또는 /* ~ */
6. 문자열 연결은 .

```
<html>
  <head>
  </head>
  <body>
    <?php
      $name = 'hololo';

      //print name
      echo "<h2>hi, ".$name.</h2>";
    ?>
  </body>
</html>
```



01. PHP – 실습



public_html폴더 안에 php파일 만들기 (phpBasic.php)

```
<?php
    $name = '서연주';
    echo [redacted];

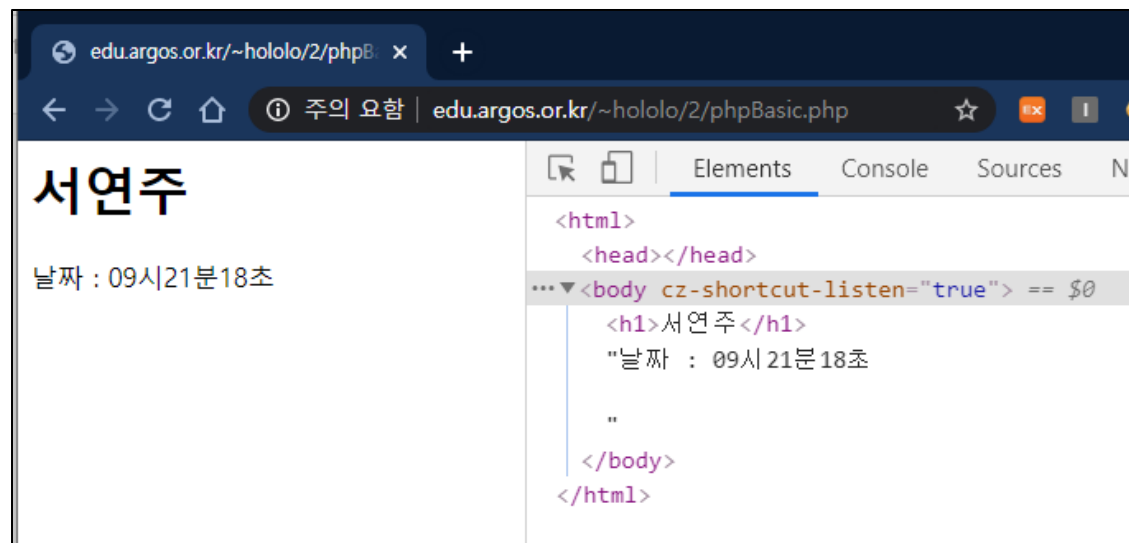
    echo "날 짜 : ".date("h시 i분 s초 ");
?>
```

[해야 할 것]

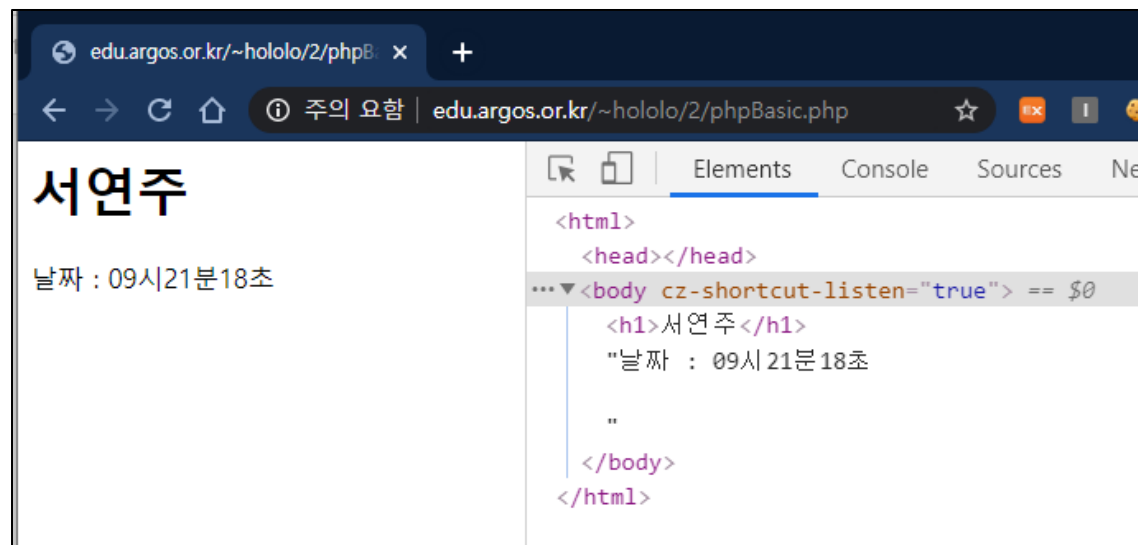
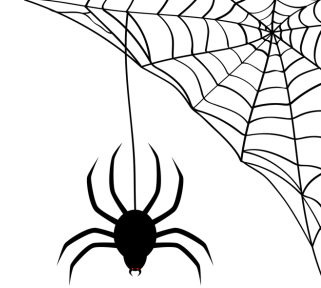
\$name이라는 변수에 이름을 넣어,
echo를 이용해 출력하되,
html태그 중 <h1>태그 안에 이름을 넣어 출력할 것

실습 1에서 한 것처럼 php파일 주소를 브라우저 주소창에다 입력

~~~~~/파일명.php

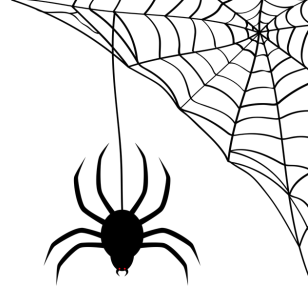


# 01. PHP – 실습



<알 수 있는 것>

1. 서버측에서 작성한 php코드는 클라이언트 측에서 보이지 않음
2. 동적인 페이지를 작성 가능



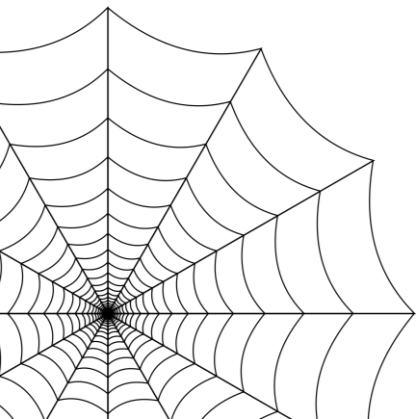
<02/>

<PHP 심화>

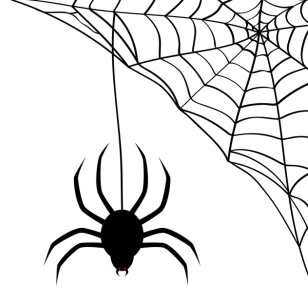
< form />

< Cookie & Session />

</PHP 심화>



## 02. PHP 심화 - form



: 입력을 받아 한번에 서버로 보내는 경우 사용

폼 데이터를 서버가 처리  
처리결과에 따라 새로운 html페이지를 보여줌

<속성>

**action** : 폼을 전송할 서버 쪽 파일을 지정

**name** : 폼을 식별하기 위한 이름

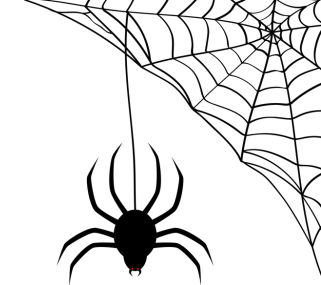
accept-charset : 폼 전송에 사용할 문자 인코딩

target : action에서 지정한 스크립트 파일을 현재 창이 아닌 다른 위치에 열도록 지정

**method** : 폼을 서버에 전송할 http 메소드 (GET 또는 POST)

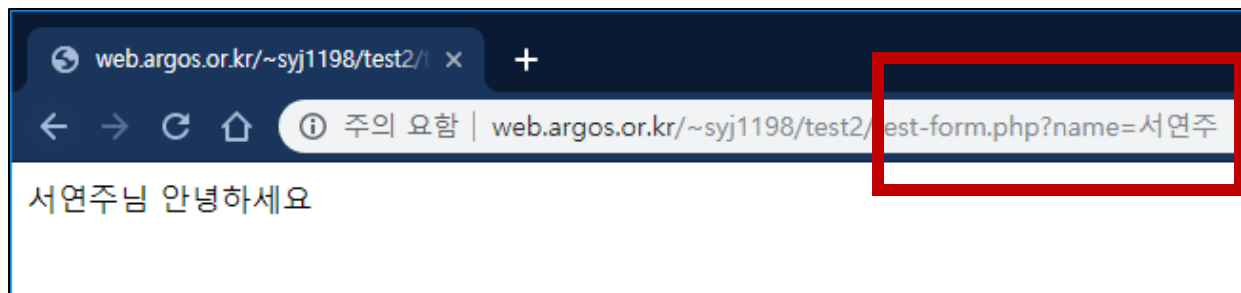
<form **action** = “처리할 주소” **method**=“get | post”> </form>

## 02. PHP 심화 - form



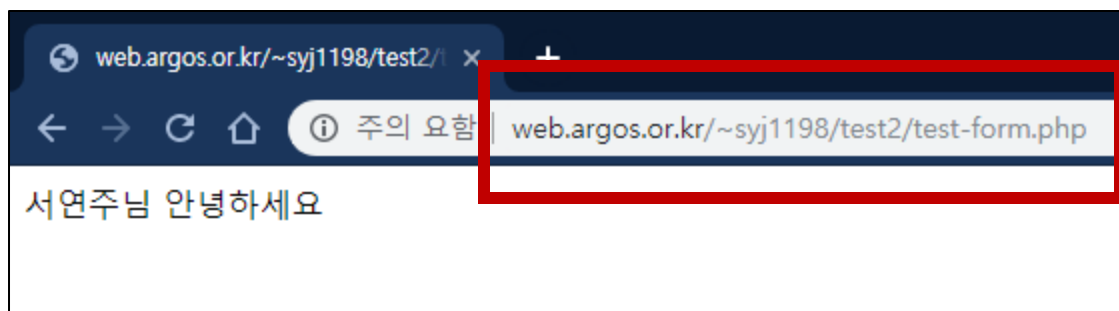
<form action = “처리할 주소” method=“get | post”> </form>

- ① GET → 주소에 데이터를 추가하여 전달하는 방식  
(보통 지정된 리소스에서 데이터를 요청하는 경우)

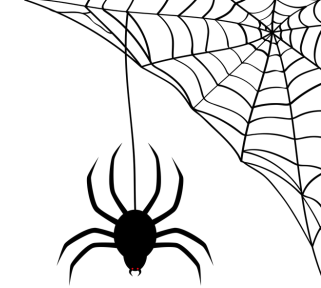


- ② POST → 주소에 추가 X, 보안성과 활용성이 get보다 좋음  
(보통 리소스에서 데이터를 쓰고, 수정, 삭제할 경우)

(로그인!!)



## 02. PHP 심화 - form



form 자체는 화면에 나타나지 않으나, **input태그**를 통해 입력할 수 있는 공간을 만들 수 있음

〈속성〉

**type** : 태그 모양을 마음대로 지정 → text, radio, checkbox, password, button, submit 등

(참고 - <https://hololo-kumo.tistory.com/9?category=801279> )

**name** : 태그 이름을 지정

required : 입력 필수란으로 지정

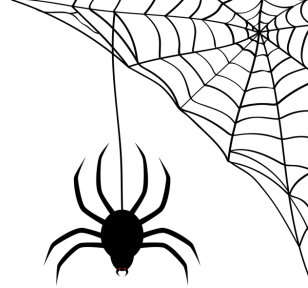
placeholder : 폼을 서버에 전송할 http 메소드 (GET 또는 POST)

autofocus : 웹 페이지가 로딩되자마자 포커스 자동 이동

```
<html>
  <head>
  </head>
  <body>
    <form>
      ID : <input type="text" name="id"/>
      <br>
      PW : <input type="password" name="pw"/>
      <br>
      <input type="submit" name="submit_b"/>
    </form>
  </body>
</html>
```



## 02. PHP 심화 - form



form 자체는 화면에 나타나지 않으나, **input태그**를 통해 입력할 수 있는 공간을 만들 수 있음

```
<html>
  <head>
  </head>
  <body>
    <form>
      ID : <input type="text" name="id"/>
      <br>
      PW : <input type="password" name="pw"/>
      <br>
      <input type="submit" name="submit_b"/>
    </form>
  </body>
</html>
```

edu.argos.or.kr/~hololo/2/formBasic.html

ID :

PW :

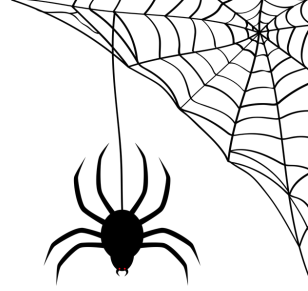
제출

텍스트문을 전달 → type = “text”

비밀번호를 전달 → type = “password”

제출 버튼은 → type = “submit”

## 02. PHP 심화 - form (실습)



```
<html>
  <head>
    <meta charset="utf-8">
    <title>simpleLogin</title>
  </head>
  <body>
    <h1>안녕하세요</h1>
    <form>
      ID : <br>
      <input type="text" name="id"/><br>
      PW : <br>
      <input type="password" name="pw"/><br>
      <input type="submit" name="submit_b"/>
    </form>
  </body>
</html>
```

simpleLogin.html

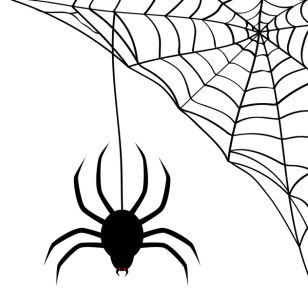
[해야 할 것]

- 데이터를 전송할 주소로 'login\_result.php'로 지정
- 데이터 전송 방식을 "POST"로 할 것

```
?php
    echo "Your ID is {$_POST['id']}<br>";
    echo "Your PW is {$_POST['pw']}<br>";
?>
```

login\_result.php

## 02. PHP 심화 - form (실습)



simpleLogin

← → ↻ ⌂ 주의 요함 | edu.argos.or.kr/~hololo/2/simpleLogin.html

안녕하세요

ID :  
argos

PW :  
.....

제출



edu.argos.or.kr/~hololo/2/login x +

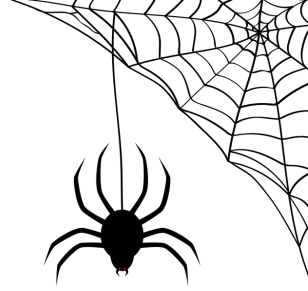
← → ↻ ⌂ 주의 요함 | edu.argos.or.kr/~hololo/2/login\_result.php

Your ID is argos.  
Your PW is argos.

id와 pw 모두 “argos”를 입력

제출 버튼을 누를 경우?

## 02. PHP 심화 - form (실습)



login\_result.php

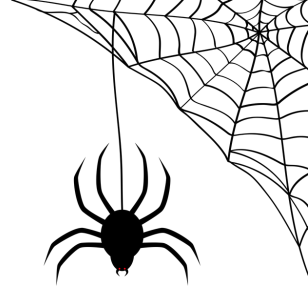
```
<?php
    echo "Your ID is {"$_POST['id']}.<br>";
    echo "Your PW is {"$_POST['pw']}.<br>";
?>
```

PHP의 슈퍼전역변수 : 미리 정의된 변수들  
(POST, GET, SESSION, COOKIE, SERVER 등)

배열처럼 접근 가능

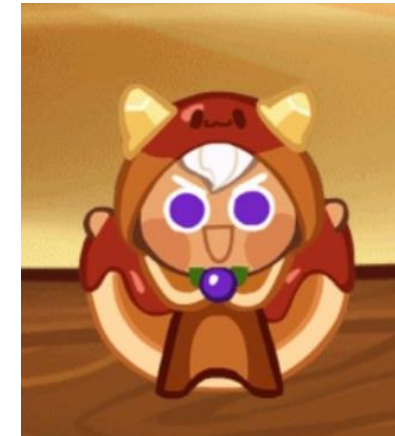
`$_POST['id']` → POST로 넘어온 값 중 name이 id인 것

## 02. PHP 심화 - Cookie



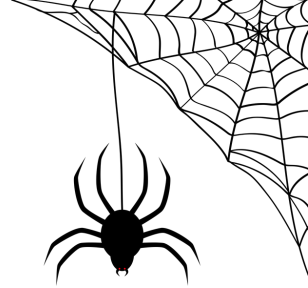
쿠키(cookie) : 웹 사이트에 접속할 때 서버에 의해 사용자의 컴퓨터에 저장되는 정보  
저장된 사용자의 정보를 남겨서 필요할 때마다 재사용  
남아 있는 과자 부스러기!!

- 로그인 정보 등을 저장하는 용도로 많이 활용 (자동 로그인)
- 하지만 사용자의 정보가 고스란히 → 위험



귀여워

## 02. PHP 심화 - Cookie(실습)



PHP에서는 `setcookie()` 함수를 이용하여 쿠키 생성

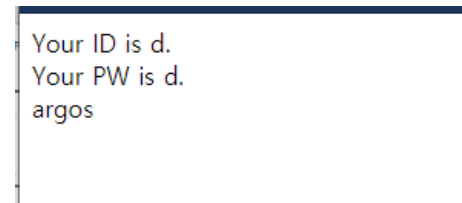
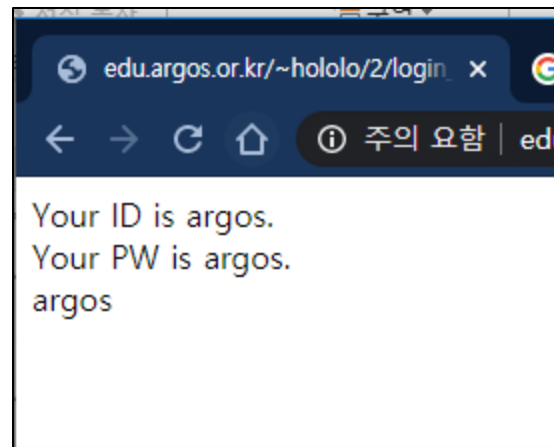
`setcookie` ("쿠키로 쓸 이름", 쿠키로 지정해줄 값, 언제까지 지속 할지);

(시간은 `time()`으로 현재 시간 + 초단위)

```
<?php
    setcookie('user_cookie',$_POST['id'],time()+60);
    echo "Your ID is ".$_POST['id']."<br>";
    echo "Your PW is ".$_POST['pw']."<br>";
    echo $_COOKIE['user_cookie']."<br>";
?>
```

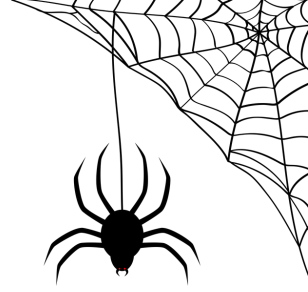
기존의 login\_result.php파일 수정하기

(1분간 지속)



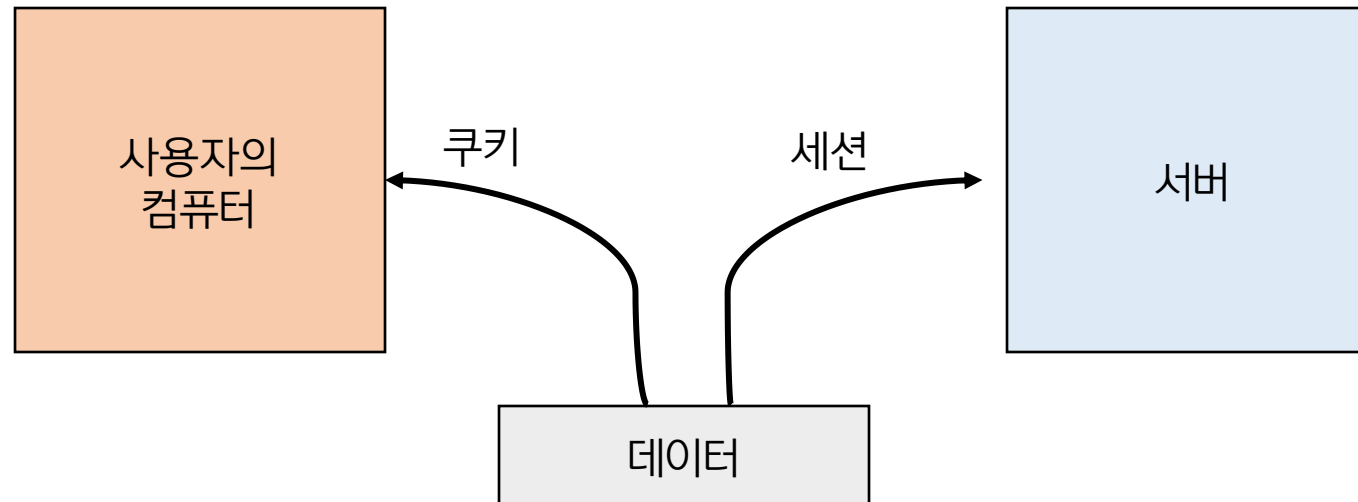
먼저 argos / argos로 로그인 후,  
다르게 로그인을 하면?  
→쿠키는 그대로 남아있음

## 02. PHP 심화 - 세션

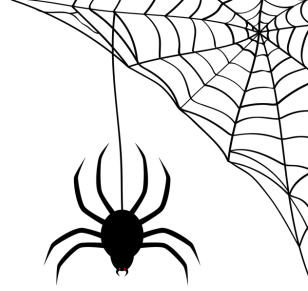


Session : 여러 페이지에 걸쳐 사용되는 사용자 정보를 저장하는 방법  
사용자가 브라우저를 닫아 서버와의 연결을 끝내는 시점

**쿠키**는 사용자의 컴퓨터에 데이터를 저장, **세션**은 서버 측에 데이터를 저장 (세션의 키값만 사용자에게)  
→ 사용자는 키값을 이용해서 서버의 데이터 사용



## 02. PHP 심화 - 세션



```
<?php
    session_start();
    $_SESSION["user_session"] = $_POST["username"];
?>
```

위 처럼 세션을 만들어 준 후,  
로그인 후에만 볼 수 있는 페이지 등을 만들 수 있음

삭제를 하기 위해서는 **session\_destroy();**  
(로그아웃에 이용) (과제에서 이용 ㅎㅎ)



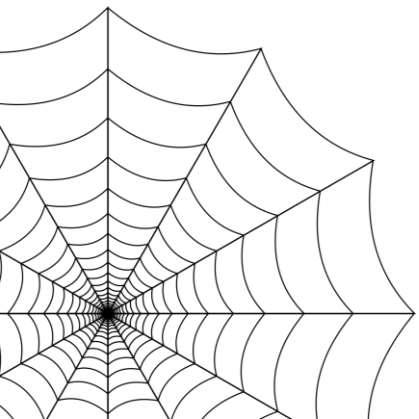
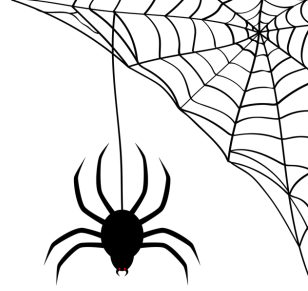
<03/>

<MySQL>

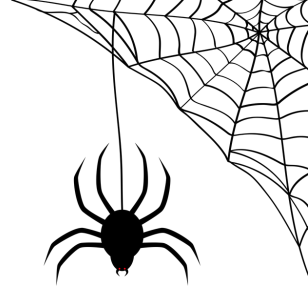
< MySQL? />

< 실습 />

</MySQL>



# 03. MySQL - MySQL



PHP와 함께 사용되는 가장 대중적인 **관계형 데이터베이스 시스템**

데이터베이스(DB:database) : 통합하여 관리되는 데이터의 집합체

→ 효율적 처리, 여러 업무에서 여러 사용자가 사용 가능

ex) 학교라는 데이터베이스

-학생  
-선생님  
-수업  
테이블

열(column) : 관계된 데이터의 묶음

| NO | ID  | PW  |
|----|-----|-----|
| 1  | id1 | pw1 |
| 2  | id2 | pw2 |
| 3  | id2 | pw3 |

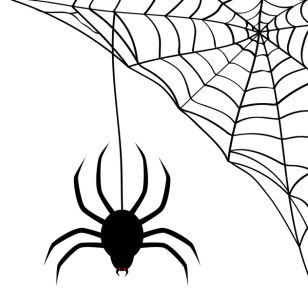
행(row)

<테이블>

→ **key**로 구분, 테이블에서 행의 식별자로 이용되는 열  
설계자가 지정해줌

ex) NO를 key로 설정

## 03. MySQL - MySQL(실습)



- ① mysql -u(id) -p를 입력 (p 다음에 비밀번호를 바로 입력해도 된다)
- ② password를 입력 (기본 비밀번호는 argos)

```
hololo@argos-edu:~/public_html/2$ mysql -uhololo -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 5.7.31-0ubuntu0.18.04.1 (Ubuntu)

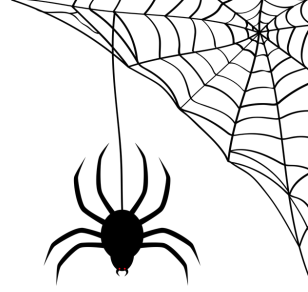
Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

## 03. MySQL - MySQL(실습)



- 데이터베이스 목록 보기 : `show databases;` (앞으로 쿼리(요청문)는 항상 세미콜론으로 끝나게)

(모르고 그냥 엔터를 눌렀다면 세미콜론을 입력하고 엔터!!)

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| syj1198 |
+-----+
2 rows in set (0.02 sec)
```

주의 !  
databases 입니다 !  
항상 마지막엔 세미콜론!!

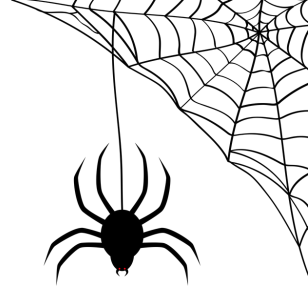
Database : information\_schema, 자기id  
데이터베이스가 있음

- 데이터베이스 사용하기 : `use 사용할DB이름;`

```
mysql> use syj1198;
Database changed
```

내가 syj1198라는 데이터베이스를 사용하겠다!

# 03. MySQL - MySQL(실습)



- 테이블 목록 보기 : `show tables;` 데이터베이스는 테이블들로 구성

```
mysql> show tables;  
Empty set (0.03 sec)
```

→ 아무 테이블도 안만들어준 경우  
현재는 이렇게 뜹니다...

우린 아직 테이블이 없습니다!!  
만들어줍시다!!

```
mysql> SHOW TABLES;
```

```
+-----+
```

```
| Tables_in_NADDRU |  
+-----+  
데이터베이스
```

```
+-----+
```

```
| class |  
| class2 |  
| power |  
+-----+  
테이블
```

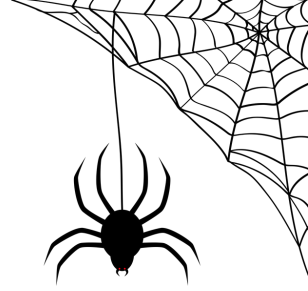
```
+-----+
```

```
3 rows in set (0.00 sec)
```

→ 테이블이 있는 경우  
: NADDRU라는 데이터베이스 안에  
class  
class2  
power라는  
테이블들이 있다!

출처 : <https://ra2kstar.tistory.com/1>

# 03. MySQL - MySQL(실습)



- 테이블 만들기 : create table(~~);

```
create table (테이블 이름) (  
    (컬럼명1) (Datatype) (옵션들) ,  
    (컬럼명2) (Datatype) (옵션들) ,  
    (컬럼명3) (Datatype) (옵션들)  
);
```

**주의!**  
**한 줄 쓰고 실행!!**

- 테이블 삭제하기 : drop table (테이블 이름)

ex) drop table table1;

(실습아님)

\* 화살표는 직접 입력하는 것이 아니라

① 한 줄 입력하고 ENTER누르면 자동으로 나오는 것

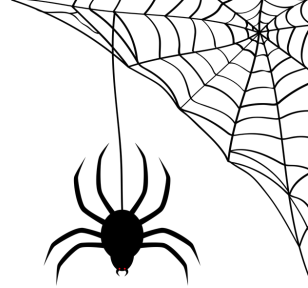
| NO | ID  | PW  |
|----|-----|-----|
| 1  | id1 | pw1 |
| 2  | id2 | pw2 |
| 3  | id2 | pw3 |

```
mysql> create table table1(  
-> no int unsigned AUTO_INCREMENT,  
-> id varchar(10) not null,  
-> pw varchar(10) not null,  
-> primary key(no)  
-> );
```

```
mysql> show tables;  
+-----+  
| Tables_in_hololo |  
+-----+  
| table1            |  
+-----+  
1 row in set (0.00 sec)
```

<table1이라는 테이블 만들기>

# 03. MySQL - MySQL(실습)



- 테이블 만들기 : create table;

```
mysql> create table table1(  
  -> no int unsigned AUTO_INCREMENT, ①  
  -> id varchar(10) not null, ②  
  -> pw varchar(10) not null,  
  -> primary key(no) ③  
  -> );
```

varchar?  
유형이 가변길이이므로,  
입력한 텍스트 크기에 맞게 가변적으로 크기가 줄어듦

varchar(n) → 최대 n바이트 만큼 사용

ex)

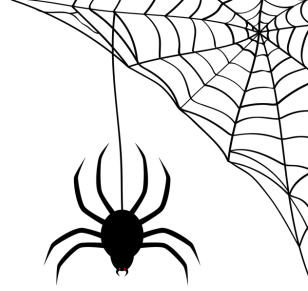
varchar(100), 'hello' 를 저장? 5바이트만 사용

① no / int / unsigned / AUTO\_INCREMENT  
→ no이라는 행, int형, 양수값으로, 자동으로 증가하게!

② id / varchar(10) / not null,  
→ id라는 행, varchar로 10바이트, null저장 안됨

③ primary key (no)  
→ 많은 열들을 key로 구분한다고 했음  
→ key를 no으로 설계자가 설정

## 03. MySQL - MySQL(실습)



- 테이블에 값 넣기 : Insert into 테이블명 (컬럼명1, ...) values("값", ...);

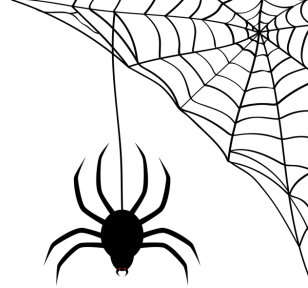
①

```
mysql> insert into table1 (id,pw) values ("argos","argos");  
Query OK, 1 row affected (0.05 sec)  
  
mysql> insert into table1 (id,pw) values ("web","hacking");  
Query OK, 1 row affected (0.04 sec)  
  
mysql> insert into table1 (id,pw) values ("hololo","hololo");  
Query OK, 1 row affected (0.04 sec)
```

| NO | ID     | PW      |
|----|--------|---------|
| 1  | argos  | argos   |
| 2  | web    | hacking |
| 3  | hololo | hololo  |



### 03. MySQL - MySQL(실습)



#### - 테이블 조회하기

: 테이블의 모든 데이터 → select \* from (테이블명);

특정 컬럼 데이터만 → select (컬럼명) from (테이블명);

조건에 따라 특정 컬럼만 → select (컬럼명) from (테이블명) where (조건);

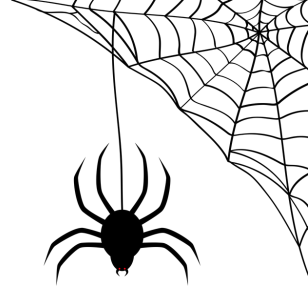
| NO | ID     | PW      |
|----|--------|---------|
| 1  | argos  | argos   |
| 2  | web    | hacking |
| 3  | hololo | hololo  |

```
mysql> select * from table1;
+-----+-----+-----+
| no | id   | pw   |
+-----+-----+-----+
| 1  | argos | argos |
| 2  | web  | hacking |
| 3  | hololo | hololo |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> select id from table1;
+-----+
| id   |
+-----+
| argos |
| web  |
| hololo |
+-----+
3 rows in set (0.00 sec)
```

```
mysql> select id from table1 where pw="hacking";
+-----+
| id   |
+-----+
| web  |
+-----+
1 row in set (0.01 sec)
```

## 03. MySQL - MySQL(실습)



- 테이블에서 값 수정하기 : update (테이블명) set (컬럼명)="넣고 싶은 값" where (컬럼명)="원래 값";

```
mysql> select * from table1;
```

| no | id     | pw      |
|----|--------|---------|
| 1  | argos  | argos   |
| 2  | web    | hacking |
| 3  | hololo | hololo  |

3 rows in set (0.00 sec)

```
mysql> update table1 set id="webedu" where id="hololo";  
Query OK, 1 row affected (0.04 sec)  
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from table1;
```

| no | id     | pw      |
|----|--------|---------|
| 1  | argos  | argos   |
| 2  | web    | hacking |
| 3  | webedu | hololo  |

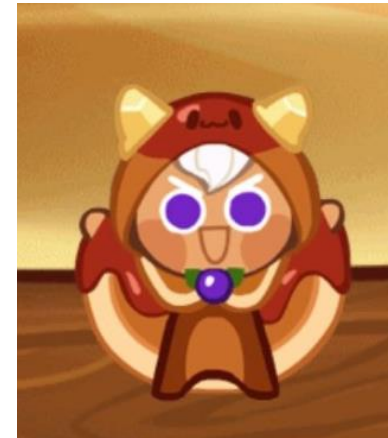
3 rows in set (0.00 sec)

## 03. MySQL - MySQL(실습)

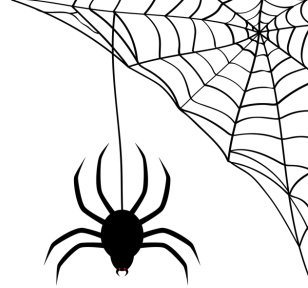
- MySQL 종료 : quit

```
mysql> quit  
Bye  
hololo@argos-edu:~/public_html/2$ █
```

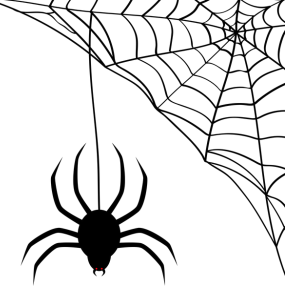
깜찍하게 인사까지 함



귀여워



## 03. MySQL - MySQL(실습)



이제 우리가 만든 테이블을 페이지에 출력해봅시다 (두둥)

```
<?php
$conn = mysqli_connect("localhost", "hololo", "argos", "hololo");
if($conn)
    echo "ok";
else
    echo "no";

echo "<br>";

$query = "localhost", id, pw, db 이름 순으로 적음
$result = mysqli_query($conn, $query);

while($row = mysqli_fetch_array($result)){
    echo $row['id'].":".$row['pw']."<br>";
}
?>
```

php파일

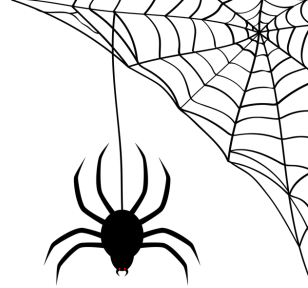
[해야 할 것]

1. 외부 php파일 만들기  
(아까 우리가 만든 MySQL의 테이블을 출력하는 파일)
2. 18페이지 simpleLogin의 연결파일을 수정

```
<html>
  <head>
    <meta charset="utf-8">
    <title>simpleLogin</title>
  </head>
  <body>
    <h1>안녕하세요</h1>
    <form action="test.php" method="post">
      ID : <br>
      <input type="text" name="id"/><br>
      PW : <br>
      <input type="password" name="pw"/><br>
      <input type="submit" name="submit_b"/>
    </form>
  </body>
</html>
```

simpleLogin.html

### 03. MySQL - MySQL(실습)



← → ↻ ⬆ ⚠ 주의 요함 | e

## 안녕하세요

ID :  
d

PW :  
•

제출

누르면

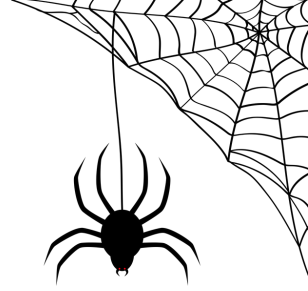
edu.argos.or.kr/~hololo/2/test.p x +

← → ↻ ⬆ ⓘ 주의 요함 | edu.argos.or.kr/~hololo/2/test.php

ok  
argos:argos  
web:hacking  
webedu:hololo

현재는 로그인과는 상관없지만, 이 둘을 이용해서 로그인, 로그아웃 페이지를 만들 예정

## 03. MySQL - MySQL(실습)



```
<?php
    $conn = mysqli_connect("localhost", "hololo", "argos", "hololo");
    if($conn)
        echo "ok";
    else
        echo "no";

    echo "<br>";
    $query = "select * from table1";
    $result = mysqli_query($conn, $query);
    while($row = mysqli_fetch_array($result)){
        echo $row['id'].": ".$row['pw']."<br>";
    }
?>
```

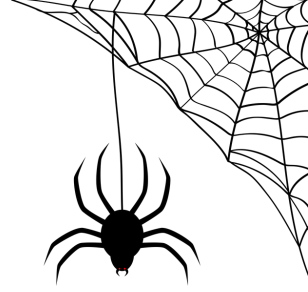
mysqli\_connect("localhost", id, pw, DB)

전달해줄 쿼리문을 \$query에 저장 (모든 값을 table1으로 부터 고를 것)

\$conn변수를 이용해서 나의 접속 정보를 전달, 연결한 db에 \$query전달

**mysqli\_fetch\_array()** : 쿼리 결과를 배열형태로 저장  
→ 하나의 row를 변수에 저장

## 04. 과제



38페이지의 제일 마지막 실습을 확장

→ 로그인을 정말로 구현되도록 만들기

→ 조건 : 테이블 table1에 있는 id와 동시에 맞는 pw를 입력해야만 로그인하도록

→ 로그인에 성공하면, 다른 html페이지를 열어도 되고,  
그냥 로그인에 성공했다고 출력하는 php페이지를 열어도 됨

참고 : 구글링을 통해서!! 엄청 많은 자료가 있음

검색 ex) mySQL php 로그인

힌트 : POST변수를 이용, Session이용, mySQL에서 반환되는 결과가 있냐 없냐?

# Q&A

과제는 없지만, 제대로 공부하고 싶다면,  
구글링 해보며 직접 만들어 보는 것이 좋습니다!

