

## 2주차 PHP + MySQL

### 1. PHP

- 변수(variable)

기본적으로 변수선언은

\$변수이름 = 데이터;

변수의 data type은

- scalar type : integer, float, string, boolean
- compound type인 array, object, callable, iterable

ex1) Scalar type

(코드)

```
1  <!-- Scalar type -->
2  <h2>[Scalar type]</h2>
3  <?php
4      $str = "name"; // string
5      $age = 25;      // integer
6      $height = 175.4; // float
7      $isResidenceCheonan = true; // boolean
8
9      var_dump($str);
10     var_dump($age);
11     var_dump($height);
12     var_dump($isResidenceCheonan);
13  ?>
```

(결과)

**[Scalar type]**

string(4) "name" int(25) float(175.4) bool(true)

ex2) Array

(코드)

```

15 <!-- Array -->
16 <h2>[Array]</h2>
17 <?php
18     $fruit = array(
19         'fruit0' => "Banana",
20         'fruit1' => "Apple",
21         'fruit2' => 3
22     );
23
24     var_dump($fruit);
25
26     for($i=0;$i<count($fruit);$i++)
27     {
28         echo '<h3>'.$fruit['fruit'.$i].'</h3>';
29     }
30     // key : string or integer
31     // value : any value of any type
32 ?>

```

(결과)

## [Array]

```
array(3) { ["fruit0"]=> string(6) "Banana" ["fruit1"]=> string(5) "Apple" ["fruit2"]=> int(3) }
```

**Banana**

**Apple**

**3**

ex3) Object

(코드)

```

33 <!-- Object -->
34 <p><h3>[Object]</h3></p>
35 <?php
36     class Car{
37         // default visibility : public
38         function Car(){
39             $this->model = "Maserati";
40         }
41         function print_car(){
42             echo "<p>$this->model</p>";
43         }
44     }
45     // create an object
46     $my_car = new Car();
47     // show object function
48     $my_car->print_car();
49 ?>

```

(결과)

**[Object]**

Maserati

---

- 조건문

- if문
- switch문

ex1) if문

(코드)

```
35 <!-- if -->
36 <h3>[if]</h3>
37 <?php
38     function compareNum($left,$right){
39         echo '<p>a='.$left.', b='.$right.'</p>';
40         if($left>$right) {
41             echo "a가 b보다 큼니다.";
42         } else if($left<$right) {
43             echo "a가 b보다 작습니다.";
44         } else {
45             echo "a와 b가 똑같습니다.";
46         }
47     }
48     compareNum(10,5);
49     compareNum(5,10);
50     compareNum(5,5);
51 ?>
```

(결과)

**[if]**

a=10, b=5

a가 b보다 큼니다.

a=5, b=10

a가 b보다 작습니다.

a=5, b=5

a와 b가 똑같습니다.

ex2) switch문

(코드)

```
53 <!-- switch -->
54 <h3>[switch]</h3>
55 <?php
56     $your_rank = 3;
57     echo '<p>';
58     switch($your_rank){
59         case 1: echo $your_rank."등에 당첨되었습니다.1등 상품은 맥북프로입니다.";
60             break;
61         case 2: echo $your_rank."등에 당첨되었습니다.2등 상품은 아이패드입니다.";
62             break;
63         case 3: echo $your_rank."등에 당첨되었습니다.3등 상품은 에어팟입니다.";
64             break;
65         default : echo "아쉽지만 다음 기회에~";
66             break;
67     }
68     echo '</p>';
69 ?>
```

(결과)

**[switch]**

3등에 당첨되었습니다.3등 상품은 에어팟입니다.

- 반복문

while, do while, for, foreach 를 사용

ex1) while, do while

(코드)

```

38 <?php
39     // while
40     echo '<p><h3>[while]</h3></p>';
41     $i = 4;
42     echo $i.'번 반복합니다.';
43     while($i--){
44         echo '<p>실행</p>';
45     }
46     // do-while
47     echo '<p><h3>[do-while]</h3></p>';
48     $j = 0;
49     echo $j.'번 반복합니다.';
50     do{
51         echo '<p>실행</p>';
52     }while($j--);
53 ?>

```

(결과)

**[while]**

4번 반복합니다.

실행

실행

실행

실행

**[do-while]**

0번 반복합니다.

실행

// do-while문의 경우 {} 내부를 먼저 한번 실행한 후 조건문이 참인지 확인하고  
반복문 실행됨

ex2) for, foreach - 반복문을 통한 숫자 index 배열과 연관 배열 접근

(공통 코드 - 배열 선언)

```

54 <!-- for, foreach -->
55 <?php
56 // 숫자 인덱스 배열
57 $arr1 = array(
58     'Math','Physics','C++'
59 );
60 // 연관 배열
61 $arr2 = array(
62     'name'=>'IU',
63     'number'=>'0100000000',
64     'height'=>'162'
65 );

```

(코드-for)

```

66 // for
67 echo '<p><h3>[for]</h3></p>';
68 var_dump($arr1);
69 for($i=0;$i<count($arr1);$i++)
70 {
71     echo "<p>$arr1[$i]</p>";
72 }
73 // 연관배열은 for문으로 나타내지 못함

```

(결과-for)

## [for]

```
array(3) { [0]=> string(4) "Math" [1]=> string(7) "Physics" [2]=> string(3) "C++" }
```

Math

Physics

C++

(코드-foreach)

```

74 // foreach
75 echo '<p><h3>[foreach]</h3></p>';
76 var_dump($arr1);
77 foreach($arr1 as $lectures)
78 {
79     echo "<p>$lectures</p>";
80 }
81 var_dump($arr2);
82 foreach($arr2 as $key => $value)
83 {
84     echo "<p>$key : $value</p>";
85 }
86 ?>

```

(결과-foreach)

## [foreach]

```
array(3) { [0]=> string(4) "Math" [1]=> string(7) "Physics" [2]=> string(3) "C++" }
```

Math

Physics

C++

```
array(3) { ["name"]=> string(2) "IU" ["number"]=> string(10) "0100000000" ["height"]=> string(3) "162" }
```

name : IU

number : 0100000000

height : 162

- 함수

ex) sum

(코드)

```
70 <!-- Function -->
71 <h3>[Function]</h3>
72 <?php
73     function sum($left,$right){
74         return $left+$right;
75     }
76     echo "<p>2+4=" .sum(2,4). '</p>';
77 ?>
```

(결과)

**[Function]**

2+4=6

## 2. MySQL

: 관계형 데이터베이스 관리 시스템

- MySQL 접속

(in console)

```
mysql -u 사용자이름 -p[password] dbname
```

- DATABASE 관련

- DATABASE 생성

```
CREATE DATABASE dbname;
```

(코드 및 결과)

```
mysql> CREATE DATABASE test;
Query OK, 1 row affected (0.00 sec)
```

- 사용할 DB선정

```
USE dbname;
```

(코드 및 결과)

```
mysql> USE test;
Database changed
```

- DB특성변경

```
ALTER DATABASE dbname CHARACTER SET=문자집합이름COLLATE =
콜레이션이름;
```

(코드 및 결과)

```
mysql> ALTER DATABASE test CHARACTER SET=utf8 COLLATE=utf8_general_ci;
Query OK, 1 row affected (0.01 sec)
```

- 삭제

```
DROP DATABASE [IF EXISTS]
dbname;
```

(코드 및 결과)

```
mysql> drop database test;
Query OK, 0 rows affected (0.02 sec)
```

- 목록보기

```
SHOW DATABASES;
```

(코드 및 결과)

```
mysql> SHOW databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| test |
+-----+
5 rows in set (0.00 sec)
```



- TABLE 관련
  - TABLE 생성

```
CREATE TABLE tablename(
    column_name1 INT(3) AUTO_INCREMENT,
    column_name2 VARCHAR(15) NOT NULL,
    column_name3 INT(11)
)ENGINE=INNODB;
```

+ 제약 조건(constraint)

: 데이터의 무결성을 지키기 위해 데이터를 입력받을 때 실행되는 검사 규칙

NOT NULL	해당 필드(=column)는 NULL값을 저장할 수 없음
UNIQUE	해당 필드는 서로 다른 값을 가져야 됨
PRIMARY KEY	NOT NULL && UNIQUE
FOREIGN KEY	하나의 테이블을 다른 테이블에 의존하게 만듦
DEFAULT	해당 필드의 기본값을 설정

(코드 및 결과)

```
mysql> CREATE TABLE table1(
  -> id INT(2) AUTO_INCREMENT,
  -> name VARCHAR(3) NOT NULL,
  -> residence VARCHAR(10) NOT NULL,
  -> PRIMARY KEY(id)
  -> );
Query OK, 0 rows affected (0.05 sec)

mysql> DESC table1;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(2) | NO   | PRI | NULL    | auto_increment |
| name  | varchar(3) | NO   |     | NULL    |               |
| residence | varchar(10) | NO   |     | NULL    |               |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql> SELECT * FROM TABLE1;
Empty set (0.00 sec)
```

- TABLE 읽기
  - 원하는 컬럼명만 읽기

```
// 원하는 컬럼만 포함해서 레코드 읽기
SELECT col_name1,col_name2,... FROM 테이블이름 ;

// 조건을 만족하는 레코드 중 원하는 컬럼만 읽기
SELECT col_name1,col_name2,... FROM 테이블이름
[WHERE 조건];
```

(코드 및 결과)

```
mysql> INSERT INTO table1
-> VALUES(2,'Bob','두정동');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM table1;
+----+-----+-----+
| id | name | residence |
+----+-----+-----+
| 1 | Mike | 병천면 가전리 |
| 2 | Bob | 두정동 |
+----+-----+-----+
rows in set (0.00 sec)
```

→ 모든 레코드 읽기

```
SELECT * FROM 테이블이름 ;
```

(코드 및 결과)

```
mysql> SELECT * FROM table1;
+----+-----+-----+
| id | name | residence |
+----+-----+-----+
| 1 | Mike | 병천면 가전리 |
| 2 | Bob | 두정동 |
+----+-----+-----+
rows in set (0.00 sec)
```

- TABLE 변경

→ 테이블명 변경

```
RENAME TABLE 테이블이름_변경전 TO 테이블이름_변경후;
```

(코드 및 결과)

```
mysql> SHOW TABLES;
+-----+
| Tables_in_test |
+-----+
| table1 |
+-----+
1 row in set (0.00 sec)

mysql> RENAME TABLE table1 TO profile;
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> SHOW TABLES;
+-----+
| Tables_in_test |
+-----+
| profile        |
+-----+
1 row in set (0.00 sec)
```

→ record 추가

1)

```
INSERT INTO 테이블이름(필드이름1, 필드이름2, 필드이름3, ...)
VALUES (데이터값1, 데이터값2, 데이터값3, ...);
```

(코드 및 결과)

```
mysql> INSERT INTO table1(name,residence)
-> VALUES('Mike','병천면 가전리');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM table1;
+-----+-----+-----+
| id | name | residence |
+-----+-----+-----+
| 1 | Mike | 병천면 가전리 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

2)

```
INSERT INTO 테이블이름
VALUES (데이터값1, 데이터값2, 데이터값3, ...);
```

+ 테이블이름 뒤 필드이름을 쓰지 않으면 테이블에 있는  
순서대로 VALUES뒤의 데이터값이 입력됨.

(코드 및 결과)

```
mysql> INSERT INTO table1
-> VALUES(2,'Bob','두정동');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM table1;
+-----+-----+-----+
| id | name | residence |
+-----+-----+-----+
| 1 | Mike | 병천면 가전리 |
| 2 | Bob  | 두정동     |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

→ record 값 변경

```
UPDATE 테이블이름  
SET 필드이름1=변경값1, 필드이름2=변경값2, ...  
WHERE 필드이름=데이터값[AND 필드이름=데이터값 ...];
```

- + WHERE 뒷부분이 없으면 해당 테이블의 모든 레코드의 필드이름1, 필드이름2의 값이 SET에서 설정한 값으로 바뀜.

(코드 및 결과)

```
mysql> UPDATE table1  
-> SET name='Jessie'  
-> WHERE id=2;  
Query OK, 1 row affected (0.01 sec)  
Rows matched: 1 Changed: 1 Warnings: 0  
  
mysql> SELECT * FROM table1;  
+-----+-----+-----+  
| id | name | residence |  
+-----+-----+-----+  
| 1 | Mike | 병천면 가전리 |  
| 2 | Jessie | 두정동 |  
+-----+-----+-----+  
rows in set (0.00 sec)
```

→ column 관련

◆ column 추가

```
ALTER TABLE ADD 필드이름 필드타입;
```

(코드 및 결과)

```
mysql> ALTER TABLE table1 ADD age INT(2);  
Query OK, 0 rows affected (0.09 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> DESC table1;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| id | int(2) | NO | PRI | NULL | auto_increment |  
| name | varchar(3) | NO | | NULL | |  
| residence | varchar(10) | NO | | NULL | |  
| age | int(2) | YES | | NULL | |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```

◆ column 삭제

```
ALTER TABLE DROP 필드이름;
```

(코드 및 결과)

```
mysql> ALTER TABLE table1 DROP age;
Query OK, 0 rows affected (0.09 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> DESC table1;
```

Field	Type	Null	Key	Default	Extra
id	int(2)	NO	PRI	NULL	auto_increment
name	varchar(3)	NO		NULL	
residence	text	YES		NULL	

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

#### ◆ column 데이터타입 변경

**ALTER TABLE** 테이블이름 **MODIFY COLUMN** 필드이름  
필드타입;

(코드 및 결과)

```
mysql> ALTER TABLE table1 MODIFY COLUMN residence text;
Query OK, 0 rows affected (0.10 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

#### - TABLE 삭제

**DELETE FROM** tablename **WHERE**  
fieldname=**value**;

// column 중 fieldname이 value인 레코드를 삭제

(코드 및 결과)

```
mysql> SELECT*FROM test;
```

id	fruit
2	watermelon
100	apple

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

```
mysql> DELETE FROM test WHERE id=2;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT*FROM test;
```

id	fruit
100	apple

```
1 row in set (0.00 sec)
```

```
DELETE FROM tablename;
```

// 테이블에 저장된 모든 데이터 삭제

// 테이블은 사라지지 않는다.

(코드 및 결과)

```
mysql> DELETE FROM test;
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM test;
Empty set (0.00 sec)
```

```
mysql> SHOW TABLES;
+-----+
| Tables_in_test |
+-----+
| profile        |
| test           |
+-----+
2 rows in set (0.00 sec)
```

// 테이블까지 삭제하기 위해서는 아래의 코드 사용

```
DROP TABLE tablename;
```

(코드 및 결과)

```
mysql> DROP TABLE test
-> ;
Query OK, 0 rows affected (0.02 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_test |
+-----+
| profile        |
+-----+
1 row in set (0.00 sec)
```

### 3. PHP와 MySQL의 연동( by mysqli )

- Connection

- mysqli\_connect()

```
mysqli_connect([string host [, string username [,
string password [, string dbname [, int port [, string
socket]]]]]);
```

(코드)

```
$conn = mysqli_connect("localhost", "root", "111111",
"test");
```

: php와 mysql의 특정 user의 DB에 연결

- Query 실행

- **mysqli\_query()**

```
mysqli_query(mysqli link, string query [, int  
resultmode]);
```

(코드)

```
$conn = mysqli_connect("localhost", "root", "111111",  
"test");  
$res = mysqli_query($conn,  
    "INSERT INTO test(id,name,phone)  
    VALUES(2, 'Mike', '01000000000');"
```

: php와 연결된 DB의 test라는 곳에 id=2,name="Mike",  
phone= "01000000000"의 값을 가지는 record 추가  
+ mysqli\_query는 SELECT, EXPLAIN, SHOW와 같은  
읽기 명령문이 query문으로 오면 mysqli\_result 객체를  
반환

- **mysqli\_result 다루기** (ex - assoc)

- **mysqli\_fetch\_...()**

함수명	리턴값
mysqli_fetch_row	일반 배열
mysqli_fetch_assoc	연관 배열
mysqli_fetch_array	일반 + 연관 배열

- **mysqli\_fetch\_assoc()**

```
mysqli_fetch_assoc(mysqli_result res);
```

(코드)

```

$conn = mysqli_connect("localhost", "root", "111111",
"test");
$sql = "SELECT * FROM test;"
$res = mysqli_query($conn,$sql);
$row = mysqli_fetch_array($res);
while($row = mysqli_fetch_assoc($res))
{
    foreach($row as $key => $value)
    {
        echo "<p>row[$key]=$value</p>";
    }
    echo "<hr>";
}

```

(결과)

```
row[id]=1
```

```
— row[name]=Mike
```

```
row[residence]=병천면 가전리
```

---

```
row[id]=2
```

```
row[name]=Jessie
```

```
row[residence]=두정동
```