

BEOMTAEK KIM

EDUSTACK/한국정보교육원
Leader in IT education

K 한국정보교육원

PHP 프로그래밍 입문

PHP 기초문법

PHP 기초 문법 (PHP의 시작과 끝)

```
<? php  
    echo "안녕하세요";  
?>
```

- <? php 는 php 프로그램의 시작, ?> 는 끝을 의미
- php 프로그램의 각 문장 끝에는 반드시 ";" 을 표기

PHP 기초 문법 (주석)

```
<?php
    // 한 행을 주석으로 처리
    echo "안녕하세요";
    /* 여러 줄을
    주석으로 처리할 때 사용 */
?>
```

PHP 기초 문법 (상수와 변수)

■ 상수

- 주어진 값을 일정하게 가진 것으로 문자열, 정수, 실수 등이 상수에 속한다
- 문자열 : 큰따옴표로 감싼 하나 또는 여러 개의 문자
- 정수 : 1000, 30, -203, -25, 0 과 같이 양수, 0, 음수로 구성된 수
- 실수 : 2.3, 3.14, -128.0, -3.14 와 같이 소수점 값을 가진 수

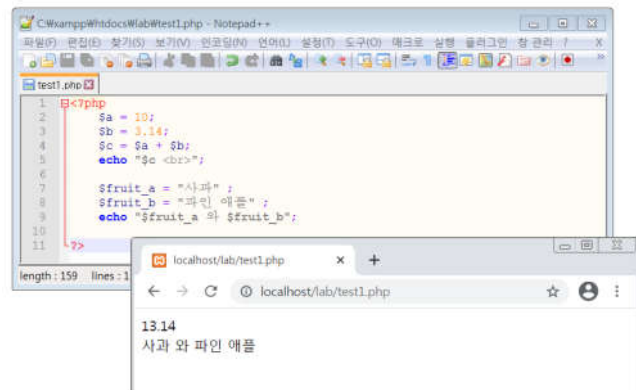
■ 변수

- 입력하는 값에 따라 내용이 변하는 것
- PHP에서는 \$로 시작하고, 변수명은 일반적으로 영문 소문자와 숫자, 밑줄을 조합해서 만든다

PHP 기초 문법 (상수와 변수)

상수와 변수 사용하기

- 문자열은 하나 또는 여러 개의 문자로 구성되며, 해당 문자를 큰 따옴표나 작은 따옴표로 감싸주어야 한다
- echo 문은 큰 따옴표 안의 내용을 출력하는데, 변수를 만나면 변수의 값을 출력하게 된다
- 변수명은 특수문자, 한글, 공백 등을 사용할 수 없고 숫자로 시작할 수 없다

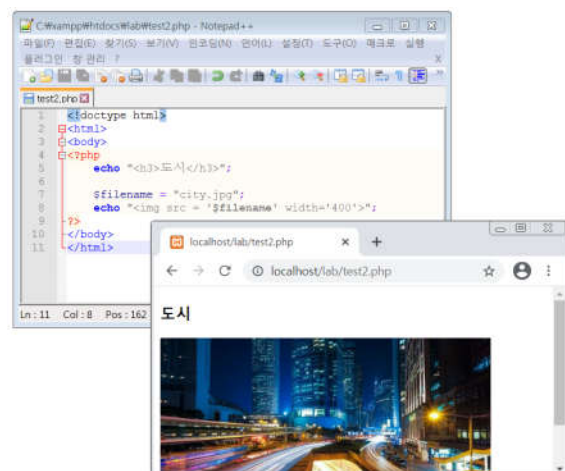


7

PHP 기초 문법 (echo문)

echo문의 기본문법

- 문자열과 변수의 값을 출력하며 문자열에 HTML 태그를 포함할 수 있다



8

PHP 기초 문법 (echo문)

■ 따옴표 사용하기

```

1 <?php
2 // 아래 1~6 중 비정상 출력되는 경우는?
3 // ----- 큰 따옴표 사용
4 $name = "홍길동" ;
5 echo $name ;
6
7 echo " 님 반갑습니다<br>" ; // 1
8 echo "$name 님 반갑습니다<br>" ; // 2
9 echo "$name'님 반갑습니다<br>" ; // 3
10
11 // ----- 작은 따옴표 사용
12 $name = '홍길동' ;
13 echo $name ;
14
15 echo ' 님 반갑습니다<br>' ; // 4
16 echo '$name 님 반갑습니다<br>' ; // 5
17
18
  
```

9

PHP 기초 문법 (echo문)

■ 따옴표 사용하기

- 작은 따옴표 안의 변수 \$name 은 문자열로 인식
- '\$name님' 이 변수로 인식 되므로 echo "{\$name}님.." ; 을 사용한다
- echo "<img src = 'city.jpg'" 는 <img src = 를 하나의 문자열로 처리하므로 오류가 발생한다. 이를 아래와 같은 방법으로 수정이 가능하다.

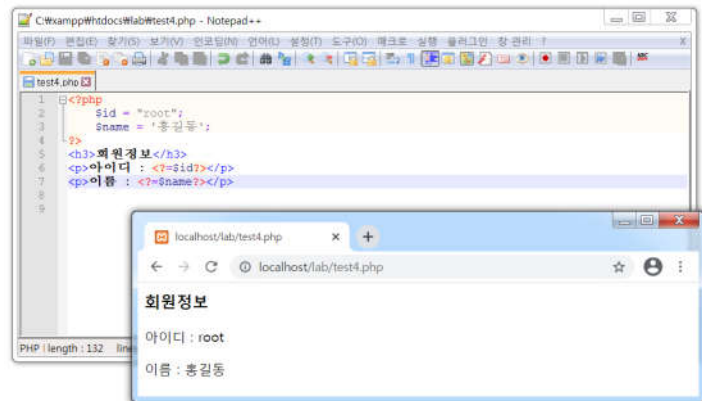
echo "<img src = 'city.jpg'" 또는 echo "<img src = ₩city.jpg₩";

10

PHP 기초 문법 (echo문)

■ echo 문의 약식 표기

- 작은 따옴표 안의 변수
\$name 은 문자열로 인식
- <?=\$id?> 는 <?php echo \$id?> 를 줄여서 표현한 것임

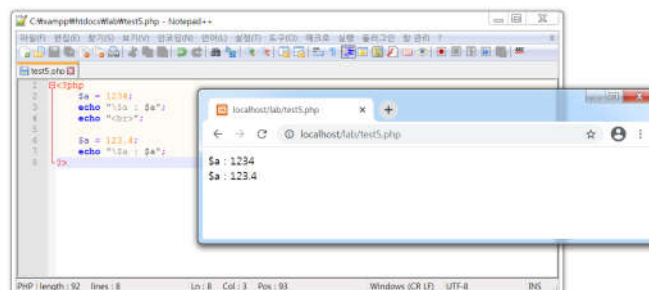


11

PHP 기초 문법 (기본 데이터형)

■ 정수와 실수

- 자연수, 0, 자연수의 음수를 통틀어 정수라 하고, 실수는 소수점 이하 자릿수가 포함된 수를 의미
- PHP에서 변수의 데이터 형은 변수 값의 데이터 형에 따라 결정

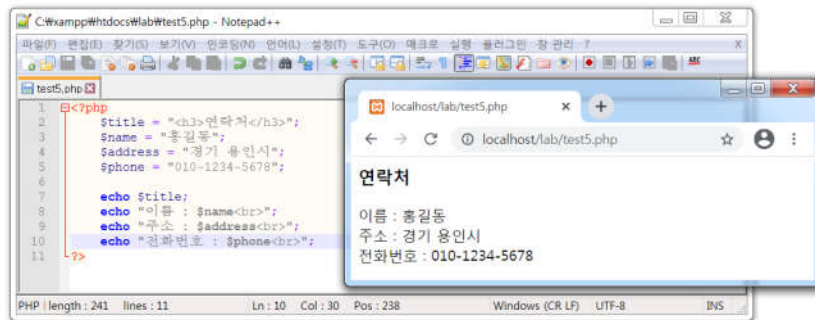


12

PHP 기초 문법 (기본 데이터형)

■ 문자열

- 하나 또는 여러 개의 문자로 구성된 데이터 형

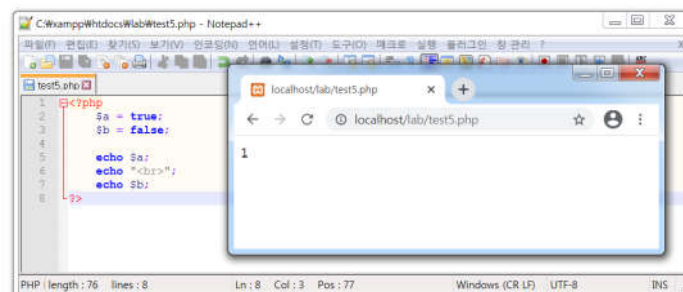


13

PHP 기초 문법 (기본 데이터형)

■ 불 형

- 참(true)과 거짓(false) 값을 표현하는 간단한 데이터 형
- \$b 와 같이 값이 없을 경우 null 을 갖는다



14

PHP 기초 문법 (연산자)

산술 연산자

- 참(true)과 거짓(false) 값을 표현하는 간단한 데이터 형

산술 연산자	기능	예	결과 값
+	더하기	10 + 20	30
-	빼기	20 - 10	10
*	곱하기	10 + 20 * 30	610
/	나누기	10 / 2	5
%	나머지	7 % 3	1 (7을 3으로 나눈 몫은 2, 나머지는 1)
++	1 증가	\$a++	현재 \$a 값에서 1 증가
--	1 감소	\$a--	현재 \$a 값에서 1 감소
**	승수	2**5	2^5

- 일반 사칙 연산과 동일하게 더하기, 곱하기, 나누기 연산 시 곱하기와 나누기를 먼저 계산

PHP 기초 문법 (연산자)

산술 연산자

The screenshot shows a Notepad++ window with the following PHP code in test6.php:

```

1 <?php
2 $a = 3 ;
3 $b = 5 ;
4 $c = $a + $b ;
5 $c++ ;
6 $c = $c + $a ;
7 $d = $a + $c * $b ;
8 echo "\$d : $d" ;
9 echo "<br>" ;
10
11 $a = 10 ;
12 $b = $a % 3 ;
13 $b-- ;
14 $c = $a - $b ;
15 $c = $c - 5 ;
16 echo "\$c : $c" ;
17

```

Overlaid on the bottom right is a web browser window showing the output of the script at localhost/lab/test6.php:

```

$d : 63
$c : 5

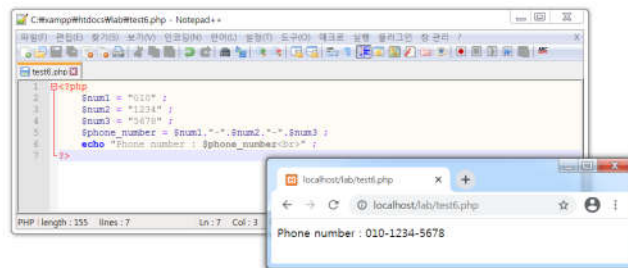
```


PHP 기초 문법 (연산자)

■ 문자열 연결 연산자

- .(dot) 는 문자열 연결 연산자로 2개 이상의 문자열을 하나로 연결하는 역할을 한다

문자열 연산자	예	설명
.	\$a.\$b	문자열 \$a 와 \$b 를 연결하여 하나의 문자열을 만들



17

PHP 기초 문법 (연산자)

■ 대입 연산자

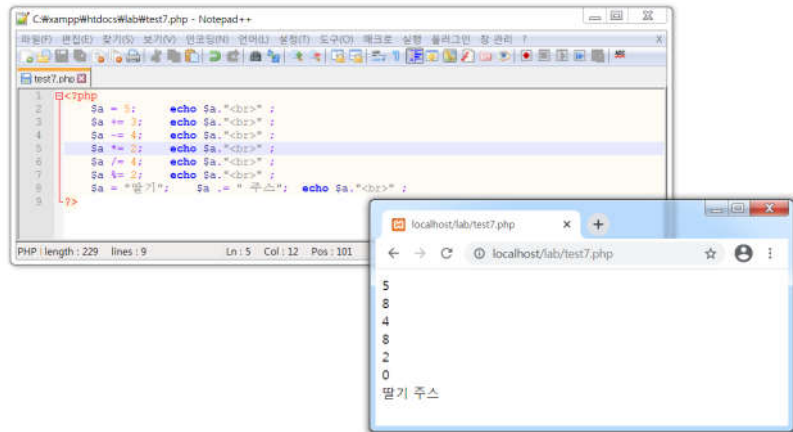
- 변수에 값을 대입할 때, 사용하는 연산자

대입 연산자	예	동일한 표현	설명
=	\$a = 10		10을 변수 a 에 저장
+=	\$a += 2	\$a = \$a + 2	\$a 에 2를 더한 결과 값을 \$a 에 저장
-=	\$a -= 2	\$a = \$a - 2	\$a 에서 2를 뺀 결과 값을 \$a 에 저장
*=	\$a *= 2	\$a = \$a * 2	\$a 에 2를 곱한 결과 값을 \$a 에 저장
/=	\$a /= 2	\$a = \$a / 2	\$a 를 2로 나눈 결과 값을 \$a 에 저장
%=	\$a %= 2	\$a = \$a % 2	\$a 를 2로 나눈 나머지 값을 \$a 에 저장
.=	\$a .= \$b	\$a = \$a.\$b	문자열 \$a 와 \$b 를 하나로 연결한 문자열을 다시 \$a 에 저장

18

PHP 기초 문법 (연산자)

■ 대입 연산자



19

PHP 기초 문법 (연산자)

■ 비교 연산자

- 참과 거짓을 판별하는 조건문이나 반복문의 조건식에서 사용

비교 연산자	예	설명
==	\$a == 3	\$a 가 3과 같으면 참, 그렇지 않으면 거짓
!=	\$a != \$b	\$a 값과 \$b 값이 서로 다르면 참, 그렇지 않으면 거짓
>	\$a > 100	\$a 가 100보다 큰 값이면 참, 그렇지 않으면 거짓
<	\$a < 100	\$a 가 100보다 작으면 참, 그렇지 않으면 거짓
>=	\$a >= 5	\$a 가 5보다 크거나 같으면 참, 그렇지 않으면 거짓
<=	\$a <= 5	\$a 가 5보다 작거나 같으면 참, 그렇지 않으면 거짓

20

PHP 기초 문법 (연산자)

■ 논리 연산자

논리 연산자	예	설명
!	!\$a	\$a가 참이면 거짓, 거짓이면 참
& &	(\$a > 10) & & (\$a < 20)	\$a가 10보다 크고 20보다 작으면 참, 그렇지 않으면 거짓
	(\$a > 0) (\$b > 0)	\$a가 0보다 크거나 \$b가 0보다 크면 참, 그렇지 않으면 거짓

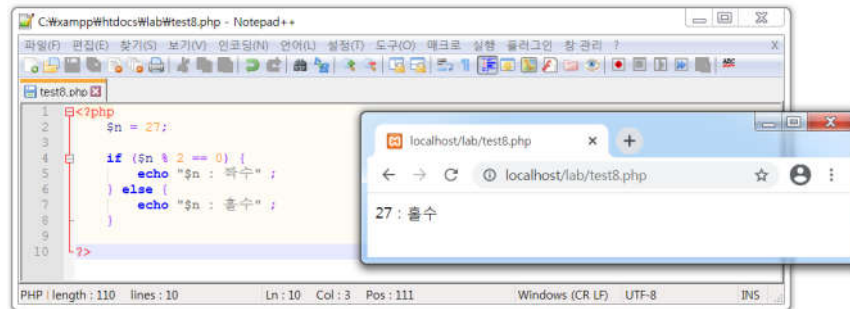
- !, & &, || 는 각각 NOT, AND, OR 를 의미한다

조건문

조건문 (조건문)

■ 조건문

- 조건식의 참과 거짓에 따라 실행하는 문장을 다르게 할 때는 조건문을 사용
- PHP 에서 사용하는 조건문에는 if 문과 switch 문이 있다



23

조건문 (if문)

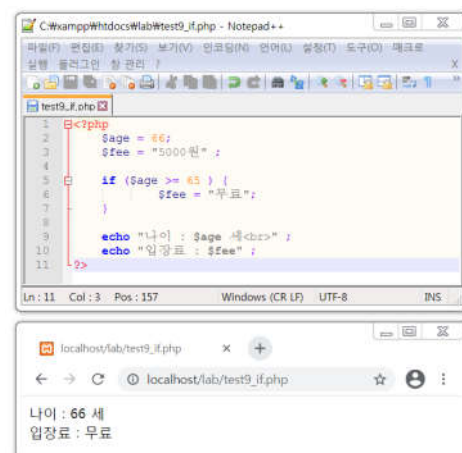
■ if ~ 구문

- if ~ 구문의 문법 형식은 다음과 같다

```

if (조건식) {
    문장1;
    문장2;
    ...
}
    
```

- 나이에 따라 입장료 판별하기



24

조건문 (if문)

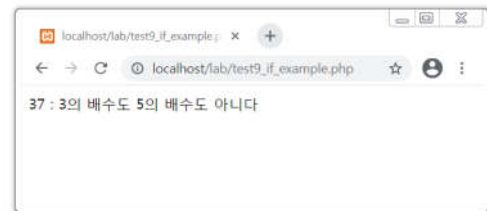
- if ~ 구문(예제)
 - 오른쪽 결과가 출력되도록 () 의 내용을 작성하시오

```
<?php
$num = 37 ;
$result = "3의 배수도 5의 배수도 아니다" ;

if (                ) $result = "3의 배수이다" ;
if (                ) $result = "5의 배수이다" ;
if (                )
    $result = "3의 배수이면서 5의 배수" ;

echo "$num : $result" ;

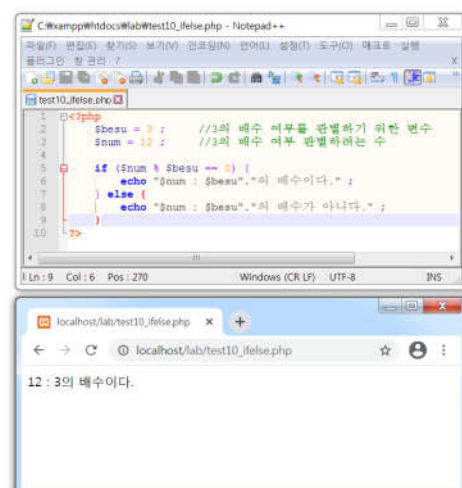
?>
```



조건문 (if문)

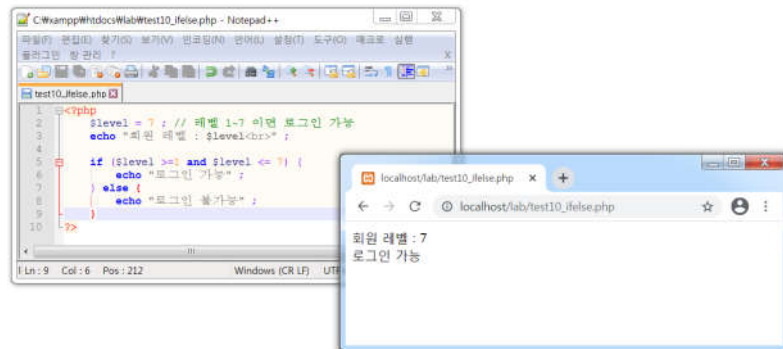
- if ~ else ~ 구문
 - if ~ else ~ 구문의 문법 형식

```
if (조건식) {
    문장1 ;
    문장2 ;
    ...
}
else {
    문장A ;
    문장B ;
    ...
}
```



조건문 (if문)

- if ~ else ~ 구문
 - 회원 레벨에 따라 로그인 가능 판별하기



27

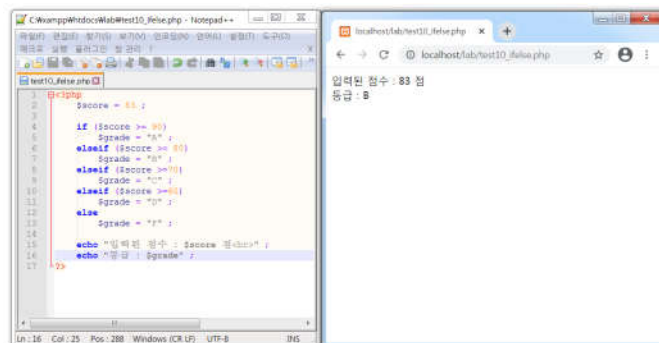
조건문 (if문)

- if ~ elseif ~ else ~ 구문
 - 하나의 if 문에 조건식이 여러 개일 때 사용하는 것으로, 각 조건에 따라 실행되는 문장을 달리할 수 있다. 일반적인 문법 형식은 다음과 같다.

```

if (조건식) {
    문장1;
    문장2;
    ...
} elseif (조건식2) {
    문장A;
    문장B;
    ...
}
...
else {
    문장 i;
    문장 ii;
    ...
}

```



28

조건문 (if문)

▪ if ~ elseif ~ else ~ 구문 (예제)

- 다음 표의 내용을 이용하여 놀이공원의 입장료를 계산하는 프로그램을 작성하시오

구분	입장료	조건
입장료 무료		- 3세미만
특별 할인	4,000원	- 3~13세 - 17시 이후 입장객
할인	8000원	- 14~18세 - 70세 이상 - 복지카드 소지자 - 국가유공자증 소지자
일반	10,000원	- 위의 조건에 해당하지 않는 모든 경우

```

1 <?php
2
3 $age = 45;
4 $welfare = "no";
5 $veteran = "yes";
6 $welfare = "no";
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

```

조건문 (if문)

▪ 중첩 if 문

- if 문을 2개 이상 섞어서 사용할 경우 이를 '중첩 if문' 이라고 하며 일반적인 형태는 아래와 같다

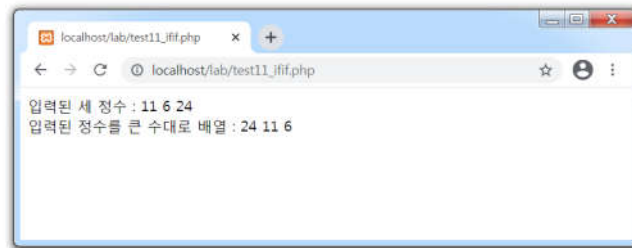
```

if (조건식) {
    문장
} elseif (조건식) {
    if (조건식) {
        문장
    } else {
        문장
    }
} else {
    문장
}

```

조건문 (if문)

- 중첩 if 문 (예제)
 - 정수 3개를 값이 큰 순서로 정렬하기



조건문 (if문)

- 중첩 if 문 (예제)
 - 오늘 날짜를 2019년 03월 10일, 출생일은 1990년 03월 03일 이라고 가정하고 아래의 알고리즘을 이용한 "만 나이 계산 프로그램" 을 작성할 것이다. () 에 적절한 내용을 작성하라

출생월	출생일	만 나이 계산식
1~2월		만 나이 = 오늘 연도 - 출생연도
3월	1~10 일	만 나이 = 오늘 연도 - 출생연도
	11 ~ 31 일	만 나이 = 오늘 연도 - 출생연도 - 1
4~12월		만 나이 = 오늘 연도 - 출생연도 - 1

조건문 (if문)

```
<?php
    $now_year = 2019 ;
    $now_month = 3;
    $now_day = 10;
    $birth_year = 1990;
    $birth_month = 3;
    $birth_day = 30;

    if(
        )
        $age = $now_year - $birth_year;
    elseif (
        )
    {
        if (
            )
            $age = $now_year - $birth_year;
        else
            $age = $now_year - $birth_year - 1;
    }
    else
        $age = $now_year - $birth_year - 1;

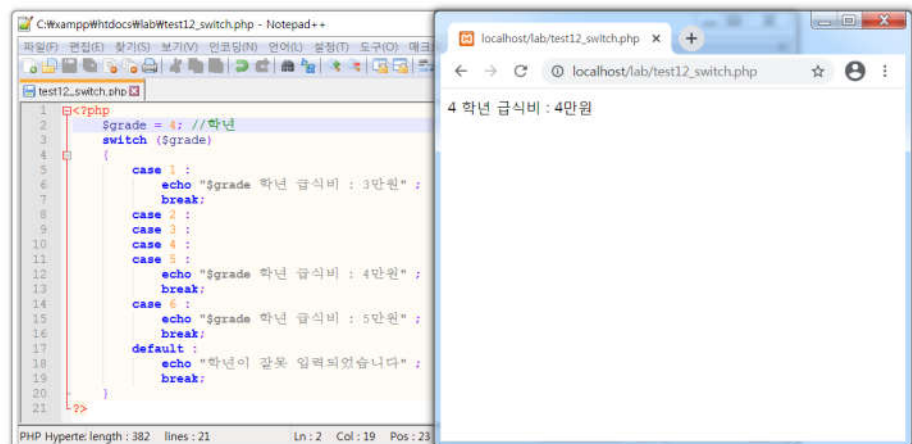
    echo "오늘 날짜 : $now_year 년 $now_month 월 $now_day 일<br>";
    echo "출생 년월일 : $birth_year 년 $birth_month 월 $birth_day 일생<br>";
    echo "만 나이 : $age 세";
?>
```

33

조건문 (switch문)

switch 문

```
switch (변수)
{
    case 값1 :
        문장1;
        ...
        break;
    case 값2 :
        문장a;
        ...
        break;
    default :
        문장;
        ...
        break;
}
```



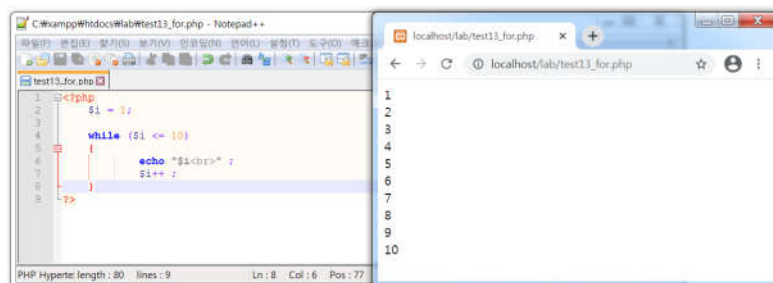
34

반복문

35

반복문 (while문)

- while
 - 반복되는 작업을 빠르게 처리하고자 하는 경우에는 반복문을 사용할 수 있으며 PHP에서는 while, for, do ~ while 문을 사용할 수 있다.
 - 1부터 10까지의 정수를 출력하는 프로그램

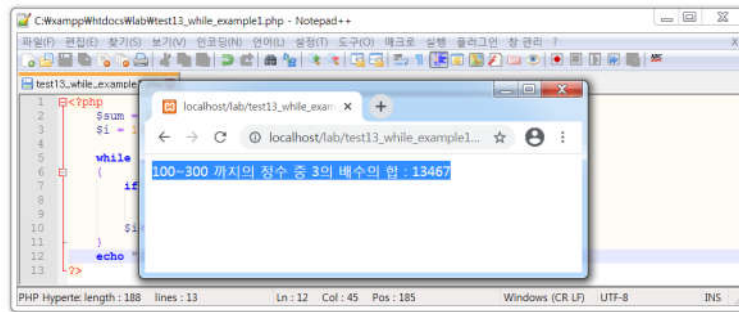


36

반복문 (while문)

▪ while 문 (예제)

- while 문으로 정수 100~300 중 3의 배수만 골라 합계를 구하는 프로그램을 작성하라

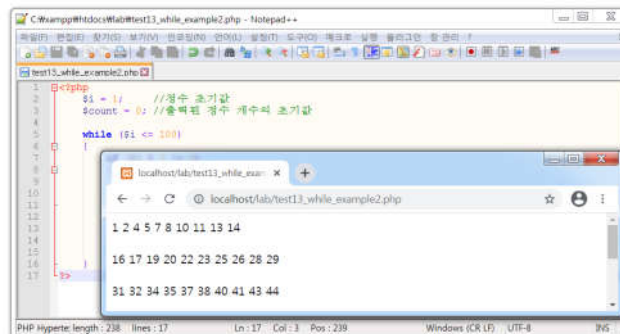


37

반복문 (while문)

▪ while 문 (예제)

- while 문으로 정수 1~100 중 3의 배수가 아닌 정수를 한 줄에 10개씩 출력하는 프로그램

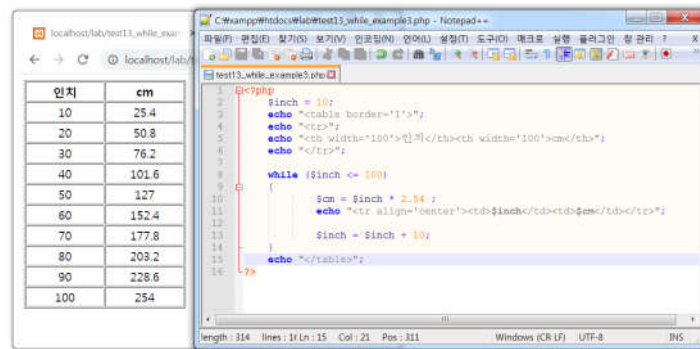


38

반복문 (while문)

▪ while 문 (예제)

- while 문을 이용하여 인치를 cm 로 변환하는 프로그램을 작성하되 인치는 10,20...100

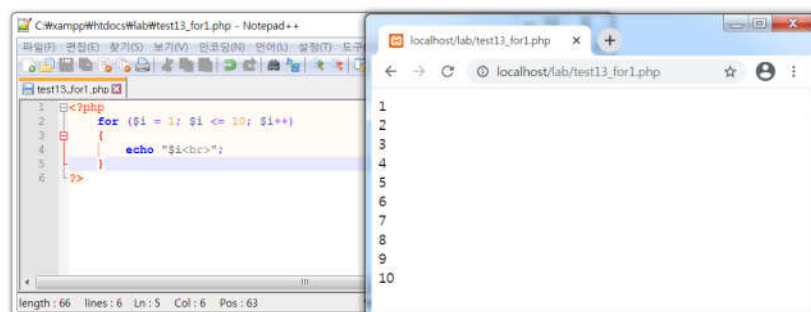


39

반복문 (for문)

▪ for문의 기본 구조

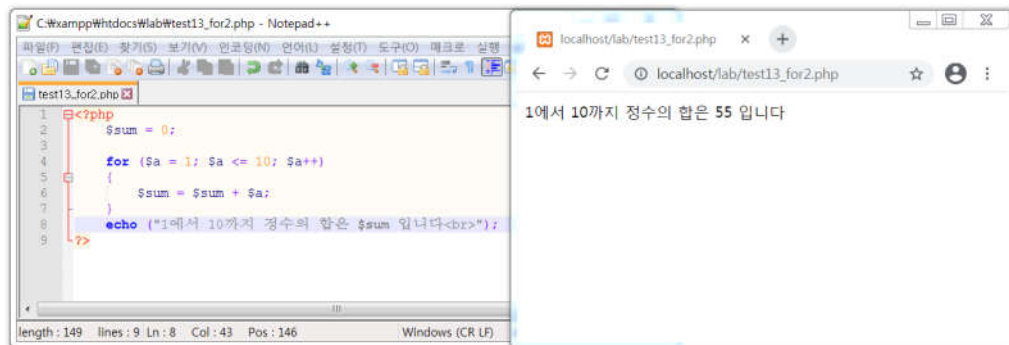
- while 과 마찬가지로 하나 또는 여러 문장을 반복할 때 사용한다. for 문은 while 문 보다 간결하고 축약된 형태라 코드가 간결해지기 때문에 while 문보다 더 자주 사용된다



40

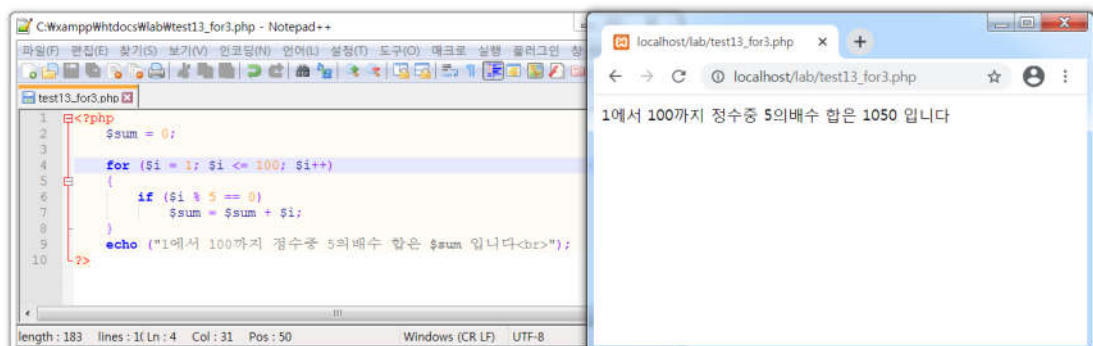
반복문 (for문)

- for문을 이용하여 구하는 정수/배수 합계
 - 1~10의 합계 구하기



반복문 (for문)

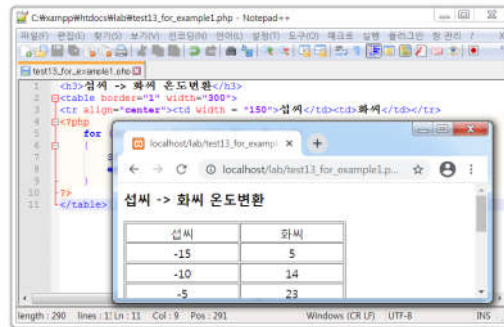
- for문을 이용하여 구하는 정수/배수 합계
 - for 문으로 정수 1~100 중 5의 배수 합계 구하기



반복문 (for문)

▪ for문을 이용하여 구하는 정수/배수 합계 (예제)

- 섭씨(C) 를 화씨(F) 변환하기 위해서는 $F = (C * 9/5) + 32$ 의 식이 필요하다. 이를 이용하여 섭씨 -15도부터 35도까지, 극간 5도로 하여 이에 대응하여 화씨 온도를 출력하는 프로그램을 작성하시오

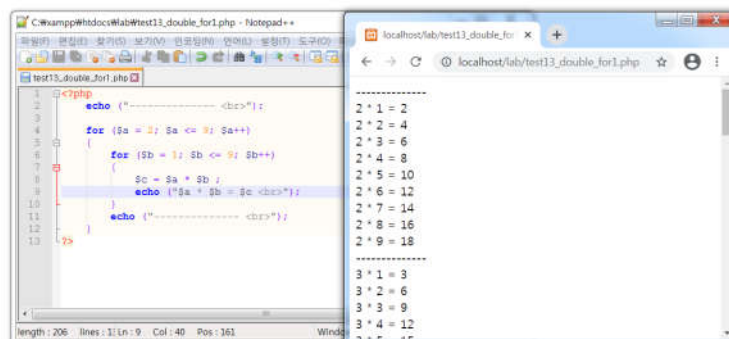


43

반복문 (이중 for문)

▪ 이중 for 문으로 구구단 표 만들기

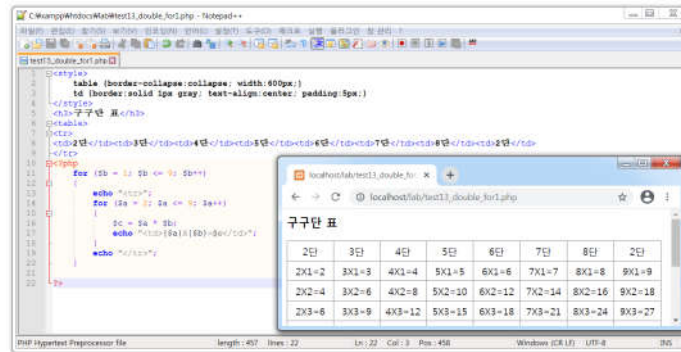
- step1. 2중 for 문을 이용하여 간단히 구구단을 작성해 본다



44

반복문 (이중 for문)

- 이중 for 문으로 구구단 표 만들기
 - step2. 추가적으로 CSS 를 적용하여 표 형태로 표현하기



45

반복문 (do ~ while문)

- while 과 do~while
 - do~while 은 while 과 거의 동일하다. 아래의 예는 정수 1~10의 합계를 while 과 do~while 을 이용하여 작성한 내용이다

<pre><?php \$i=1; \$sum=0; while (\$i <= 10) { \$sum = \$sum + \$i; \$i++; } echo "합계 : \$sum"; ?></pre>	<pre><?php \$i=1; \$sum=0; do { \$sum = \$sum + \$i; \$i++; } while (\$i <= 10); echo "합계 : \$sum"; ?></pre>
--	--

- 일반적으로 while 문을 사용하는 것이 직관적이고 편리하지만, 조건식 확인전에 루프내의 문장을 무조건 한 번은 실행해야 한다면 do~while 문을 사용한다

46

배열

47

배열 (배열)

배열의 기본 형식

- 배열은 하나의 변수명에 여러 개의 데이터를 저장할 수 있게 해주는 데이터 형
- 100명의 한 과목 성적 합계와 평균을 구하는 프로그램의 경우 100명 각각의 성적은 `$score[0] ... $score[99]` 와 같은 배열에 저장할 수 있다
- 배열의 미 사용 VS. 배열의 사용

```
<?php
    $eng = 90;
    $computer = 80;
    $program = 85;
?>
```

```
<?php
    $score[0] = 90;
    $score[1] = 80;
    $score[2] = 85;
?>
```

48

배열 (배열)

- 배열의 기본 형식
 - 배열을 이용하여 성적의 합계와 평균 구하기

The screenshot shows a Notepad++ window with a PHP script named `test_array1.php`. The script initializes an array `$score` with five elements: 90, 80, 85, 95, and 93. It then calculates the sum of these values and the average. The output is displayed in a web browser window at `localhost/lab/test_array1.php`.

```

1 <?php
2 $score[0] = 90; // 영어
3 $score[1] = 80; // 컴퓨터 계층
4 $score[2] = 85; // 기초 프로그래밍
5 $score[3] = 95; // 기초 수학
6 $score[4] = 93; // 네트워크 계층
7
8 $sum = 0;
9 for ($a = 0; $a <= 4; $a++)
10 {
11     $sum = $sum + $score[$a];
12 }
13 $avg = $sum/5;
14
15 echo "과목점수 : $score[0], $score[1], $score[2], $score[3], $score[4] <br>";
16 echo "합계 : $sum, 평균 : $avg <br>";
17
    
```

과목점수 : 90, 80, 85, 95, 93
합계 : 443, 평균 : 88.6

49

배열 (배열)

- `array()` 함수를 이용한 배열의 초기화
 - `array()` 함수를 이용하면 배열에 값을 간편히 입력할 수 있다. `array()`는 배열에 초기값을 입력할 때 사용하는 함수이다.
 - 20명의 성적을 입력하고 합계와 평균을 구하는 프로그램 작성하기

The screenshot shows a Notepad++ window with a PHP script named `test_array2.php`. The script uses the `array()` function to initialize an array `$scores` with 20 elements. It then calculates the sum of these values and the average. The output is displayed in a web browser window at `localhost/lab/test_array2.php`.

```

1 <?php
2 $scores = array(77, 74, 87, 94, 82, 87, 74, 87, 94, 82,
3               87, 74, 87, 94, 82, 87, 74, 87, 94, 82);
4
5 $sum = 0;
6 for ($a = 0; $a < 20; $a++)
7 {
8     $sum = $sum + $scores[$a];
9 }
10
11 $avg = $sum/20;
12
13 echo "합계 : $sum, 평균 : $avg";
14
    
```

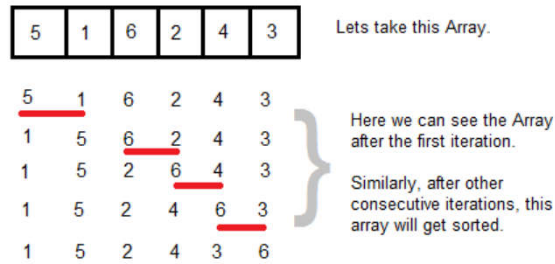
합계 : 1624, 평균 : 81.2

50

배열 (배열)

배열과 버블 정렬

- 버블 정렬은 인접한 2개의 원소를 비교하면서 값을 정렬하는 방식이다. 배열을 이용하여 배열의 원소 값을 서로 바꿔가면서 거품이 부풀어 오르듯 정렬하기 때문에 버블 정렬이라고 부른다.



배열 (배열)

오름차순으로 버블 정렬하기

```

1  $num = array(15, 12, 9, 7, 6, 12, 19, 30, 28, 26);
2  $count = 10; //배열의 원소 개수
3  echo "정렬 전 : ";
4  for ($a = 0; $a < $count; $a++)
5      echo $num[$a]. " ";
6  echo "\n";
7
8  for ($i = $count-1; $i >= 0; $i--) // $i 는 0부터 0까지 1씩 감소
9  {
10     for ($j = 0; $j <= $i; $j++) // $j는 0부터 $i까지 1씩 증가
11     {
12         if ($num[$j] > $num[$j+1]) // 인접한 두 수 비교
13         {
14             $tmp = $num[$j]; // 값의 위치를 $tmp에 잠시 대입
15             $num[$j] = $num[$j+1]; // $j의 위치를 앞의 배열 원소에 저장
16             $num[$j+1] = $tmp; // $tmp를 뒤의 배열 원소에 저장
17         }
18     }
19     echo "버블 정렬 (오름차순) 후 : ";
20     for ($a = 0; $a < $count; $a++) // 버블 정렬 후 배열의 원소 출력
21         echo $num[$a]. " ";
22     echo "\n";
23 }
24

```

정렬 전 : 15 12 9 7 6 12 19 30 28 26
 버블 정렬(오름차순) 후 : 6 7 9 12 13 15 19 26 28 30

배열 (2차원 배열)

2차원 배열

- 학생 3명이 받은 5개 과목 성적의 합계와 평균을 구할 때에는 2차원 배열을 사용해야 한다

2차원 배열
int arr[2][3]

arr[0][0]	arr[0][1]	arr[0][2]
arr[1][0]	arr[1][1]	arr[1][2]

배열 (2차원 배열)

2차원 배열

- 학생 3명이 받은 5개 과목 성적의 합계와 평균을 구하는 프로그램 작성하기

```

1 // test_array_2d.cpp
2
3 #include <iostream>
4 #include <vector>
5
6 using namespace std;
7
8 // 성적 배열 선언
9 int arr[3][5];
10
11 // 성적 입력 함수
12 void input_scores(int arr[3][5]) {
13     for (int i = 0; i < 3; i++) {
14         for (int j = 0; j < 5; j++) {
15             int score;
16             do {
17                 cout << "학생 " << i + 1 << "의 " << j + 1 << "번째 과목 성적: ";
18                 cin >> score;
19                 while (cin.get() != '\n') continue;
20             } while (score < 0 || score > 100);
21             arr[i][j] = score;
22         }
23     }
24 }
25
26 // 학생 3명의 성적 합계와 평균
27 void calculate_sum_and_avg(int arr[3][5]) {
28     for (int i = 0; i < 3; i++) {
29         int sum = 0;
30         for (int j = 0; j < 5; j++) {
31             sum += arr[i][j];
32         }
33         double avg = sum / 5;
34         cout << "학생 " << i + 1 << "의 성적 합계: " << sum << ", 평균: " << avg << endl;
35     }
36 }
37
38 int main() {
39     input_scores(arr);
40     calculate_sum_and_avg(arr);
41     return 0;
42 }
  
```

```

Score[0][0] = 88
Score[0][1] = 88
Score[0][2] = 90
Score[0][3] = 77
Score[0][4] = 63
Score[1][0] = 86
Score[1][1] = 77
Score[1][2] = 66
Score[1][3] = 96
Score[1][4] = 93
Score[2][0] = 74
Score[2][1] = 83
Score[2][2] = 95
Score[2][3] = 86
Score[2][4] = 97
1번 학생의 점수 => 합계 : 422, 평균 : 84.4
2번 학생의 점수 => 합계 : 408, 평균 : 81.6
3번 학생의 점수 => 합계 : 435, 평균 : 87
  
```

함수

함수 (함수의 개념)

■ 함수의 이해

■ 내장 함수

- 배열에 초기값 입력 : `array()`
- `round(3.7)`의 결과값은 반올림하여 4 : `round()`
- `abs(-3)`의 절대값 3 : `abs()`
- 프로그래밍 내부에 기본적으로 포함되어 있는 함수

■ 사용자 함수

- 필요한 기능을 직접 만들어서 사용하는 함수
- 함수 정의 : `"function hello()"`
- 함수 호출 : `"hello()"`

```
<?php
function hello()
{
    echo "안녕하세요";
}

hello();
?>
```

함수 (함수의 매개변수)

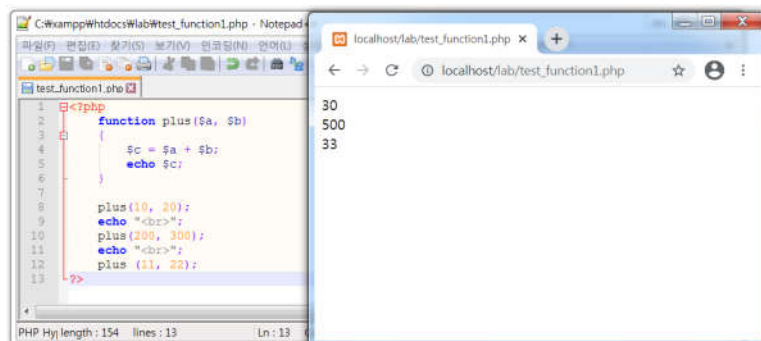
- 매개변수 사용하는 형식
 - 함수명의 괄호안에 매개변수를 사용하는 경우의 형식

```
function 함수명(매개변수1, 매개변수2, ... )
{
    문장1;
    문장2;
    ...
}
...
함수명(입력값1, 입력값2, ...);
...
```

57

함수 (함수의 매개변수)

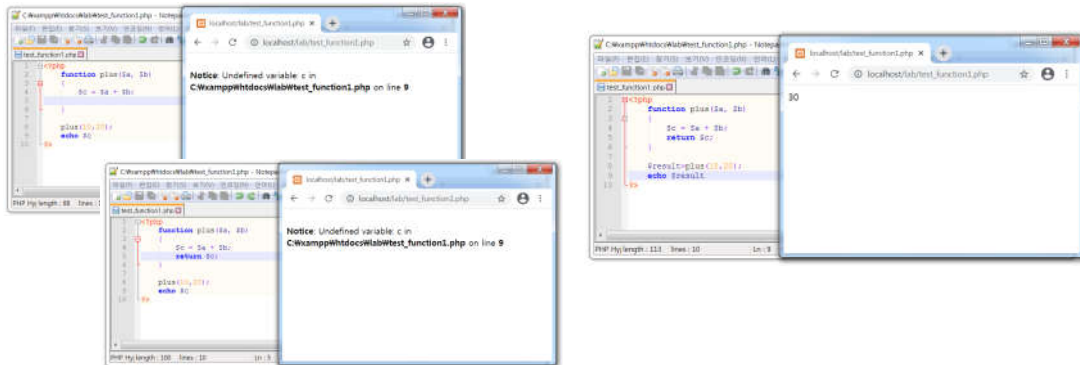
- 매개변수 사용하는 형식
 - 함수의 매개변수를 이용하여 두 수의 합 구하기



58

함수 (함수의 반환 값)

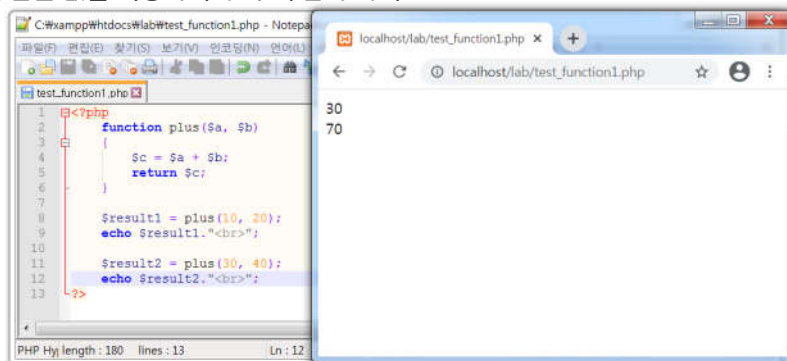
- 함수에 의해 실행된 결과값을 호출함수에서 사용할 수 있다



59

함수 (함수의 반환 값)

- 함수에 의해 실행된 결과값을 호출함수에서 사용할 수 있다
 - 함수의 반환 값을 이용하여 두 수의 합 구하기



60

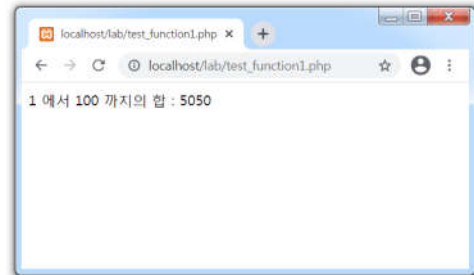
함수 (함수의 반환 값)

■ 함수의 반환 값을 이용한 정수의 합 계산

```
<?php
// sum($a, $b) 함수는 $a 부터 $b 까지의 합을 구한다
function sum($a, $b)
{
    $sum = 0;
    while ($a <= $b)
    {
        
    }
    return $sum;
}

$from = 1;
$to = 100;

$total = sum($from, $to);
echo("$from 에서 $to 까지의 합 : $total");
?>
```



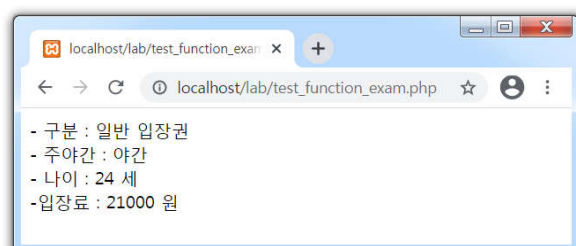
함수 (함수의 반환 값)

■ 함수를 이용한 입장료 계산 (평가)

- 함수를 이용하여 다음과 같은 놀이공원 입장료를 계산하는 프로그램을 작성하시오

구분		대인	소인
일반 입장권	주간	26,000원	19,000원
	야간	21,000원	16,000원
자유 이용권	주간	33,000원	24,000원
	야간	28,000원	21,000원
2일 자유 이용권		55,000원	40,000원
콤비권		54,000원	40,000원

* 소인은 12세 이하 또는 65세 이상



함수 (내장 함수)

PHP 내장 함수 예

- 내장 함수는 별도의 정의작업이 필요 없으며 즉시 호출하여 사용할 수 있다

함수	기능	함수	기능	함수	기능	함수	기능
abs()	절대값 구하기	round()	반올림 값 구하기	echo()	문자열 출력하기	copy()	파일 복사하기
sin()	삼각형의 사인 값 구하기	log()	로그 값 구하기	explode()	특정 문자를 기준으로 문자열 나누기	mkdir()	디렉터리 생성하기
cos()	삼각형의 코사인 값 구하기	max()	최대값 구하기	strlen()	문자열의 길이 얻기	chdir()	디렉터리 변경하기
tan()	삼각형의 탄젠트 값 구하기	sqrt()	제곱근 구하기	substr()	문자열에서 일부 문자 추출하기	fopen()	파일 열기
date()	현재 날짜 구하기	rand()	임의의 난수 생성하기	nl2br()	행바꿈(\n)을 (br)태그로 변경	fread()	파일 읽기
ceil()	소수점 아래에서 올린 정수	pi()	파이의 근사값 구하기	sprint()	문자열을 특정 양식에 맞추기	fwrite()	파일 쓰기
floor()	소수점 아래에서 내린 정수						

함수 (내장 함수)

PHP 내장 함수 예

- PHP에서 리눅스 명령의 사용과 shell 의 실행

```
<?php
system("mkdir /var/www/html/image");
system("python aaa.py");
?>
```

```
<?php
$output = shell_exec('ls -lart');
echo "<pre>$output</pre>";
?>
```

```
<?php
exec("ls /etc -al", $output);
echo "<pre>$output</pre>";
?>
```


함수 (내장 함수)

- PHP 내장함수 (예제)
 - 문자열 관련 내장 함수 사용하기

The screenshot shows a Notepad++ editor window with the following PHP code:

```

1 $tel = "010-1234-5678";
2 echo "\$tel : \$tel<br>";
3 $num_tel = strlen($tel);
4
5 echo "\$tel의 길이 : \$num_tel<br>";
6
7 $tel1 = substr($tel, 0, 3); // 앞에서 3개의 문자 가져오기
8 echo "\$tel1<br>";
9
10 $tel2 = substr($tel, 4, 4); // 4번째 문자에서 4개의 문자 가져오기
11 echo "\$tel2<br>";
12
13 $tel3 = substr($tel, 8, 4); // 9번째 문자에서 4개의 문자 가져오기
14 echo "\$tel3<br>";
15
16 $phone = explode("-", $tel); // -을 기준으로 문자열 분리
17 echo "전화번호 : \$phone[0] \$phone[1] \$phone[2]<br>";
18
19

```

The web browser window shows the output of the script:

```

$tel : 010-1234-5678
$tel의 길이 : 13
010
1234
5678
전화번호 : 010 1234 5678

```

form 과 PHP

form과 PHP (HTTP)

■ HTTP

■ 일반적인 웹 트래픽 처리 과정

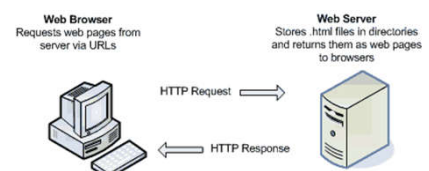


form과 PHP (HTTP)

■ HTTP

■ 데이터 흐름

- ① 사용자가 웹 브라우저 주소 창에 URL 을 입력하거나 웹 페이지의 링크를 클릭
- ② HTTP 통신 규약에 의해 사용자의 요청이 웹 서버로 전달
- ③ 웹 서버는 사용자의 요청을 처리하여 그 결과를 HTTP 통신 규약에 의해 사용자의 웹 브라우저로 전송
- ④ 웹 브라우저는 웹 서버가 보낸 데이터를 분석하여 화면에 출력
- ⑤ 웹 브라우저 화면에 텍스트, 글자, 동영상 등의 형태로 나타난 요청 결과를 사용자가 확인



form과 PHP (HTTP)

▪ POST 방식과 GET 방식

▪ POST 방식

회원 가입 양식이나 게시판에 글을 쓸 때 처럼 사용자가 폼 양식에 입력한 데이터를 웹 서버로 전달할 때 사용하며, 이미지와 같은 바이너리 파일도 웹 서버로 전달할 수 있다

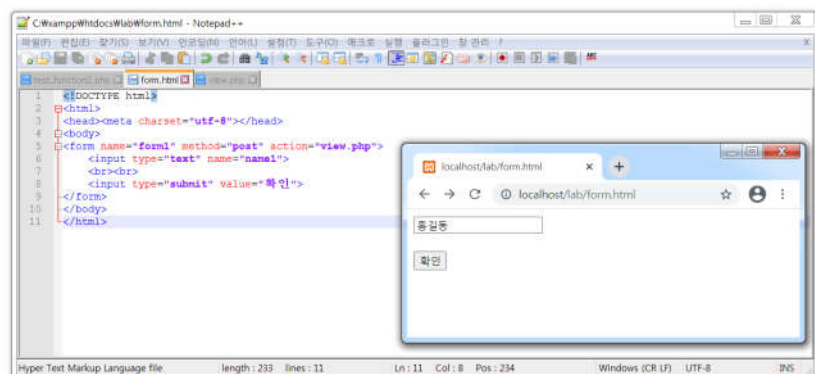
▪ GET 방식

링크를 보낼 때 URL 뒤에 데이터를 입력하여 웹 서버로 전달한다

form과 PHP (POST 방식)

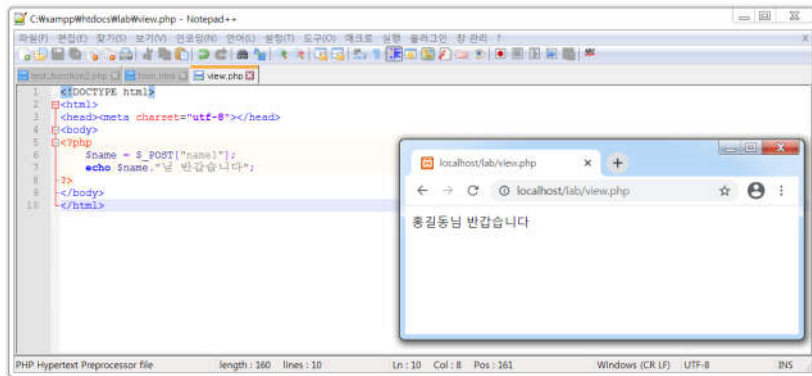
▪ 폼 양식의 데이터 처리

▪ 폼 양식 생성하기



form과 PHP (POST 방식)

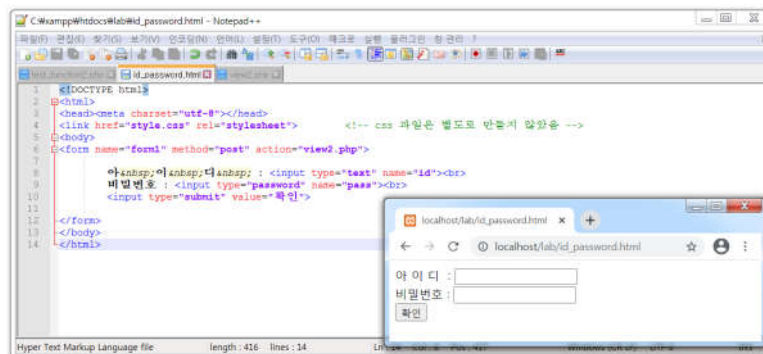
- 폼 양식의 데이터 처리
 - 폼 양식 생성하기



71

form과 PHP (POST 방식)

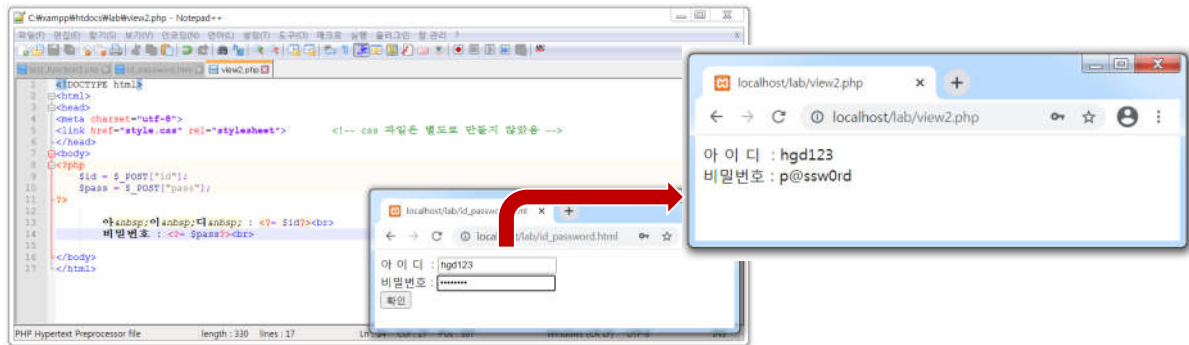
- 입력된 아이디와 비밀번호 처리
 - 아이디와 패스워드 입력 창 생성하기



72

form과 PHP (POST 방식)

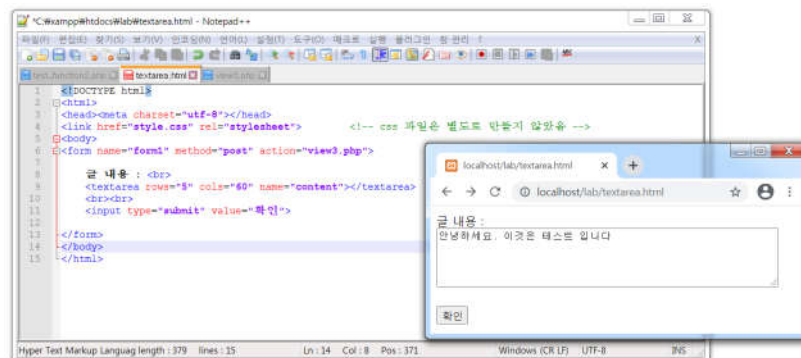
- 입력된 아이디와 비밀번호 처리
 - 아이디와 비밀번호로 입력된 데이터 처리하기



73

form과 PHP (POST 방식)

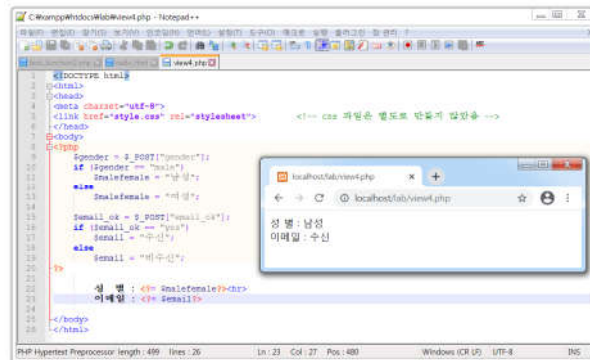
- 입력된 텍스트 처리
 - 텍스트 입력 창 생성하기



74

form과 PHP (POST 방식)

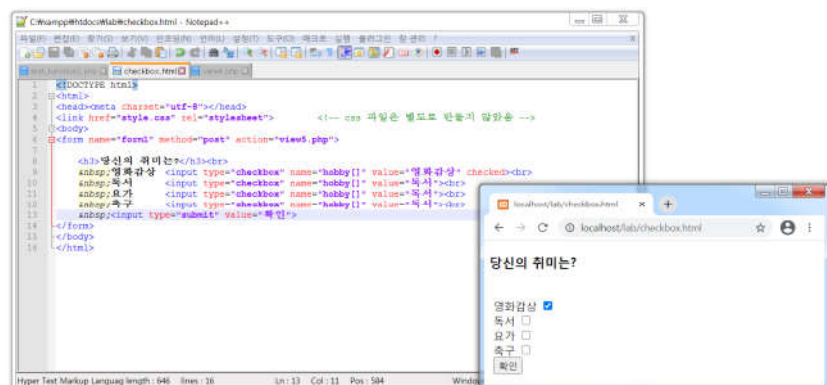
- 라디오 버튼에서 선택된 데이터 처리
 - 라디오 버튼에서 선택된 데이터 처리하기



77

form과 PHP (POST 방식)

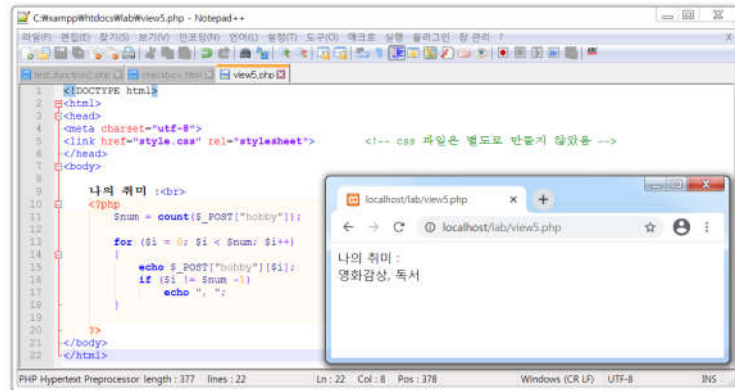
- 체크박스에서 선택된 데이터 처리
 - 체크박스 생성하기



78

form과 PHP (POST 방식)

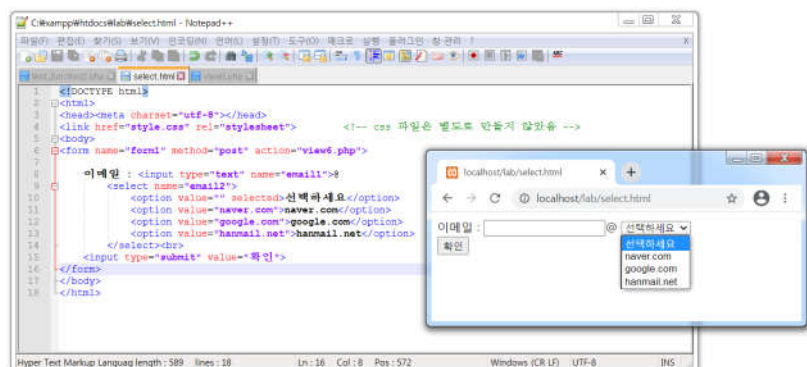
- 체크박스에서 선택된 데이터 처리
 - 체크박스 생성하기



79

form과 PHP (POST 방식)

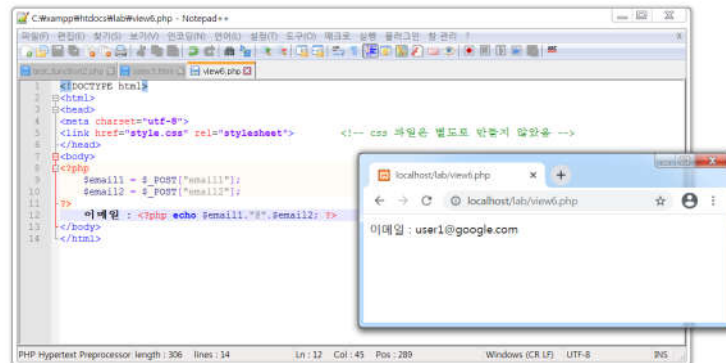
- 선택 박스에서 선택된 데이터 처리
 - 이메일 선택 박스 생성하기



80

form과 PHP (POST 방식)

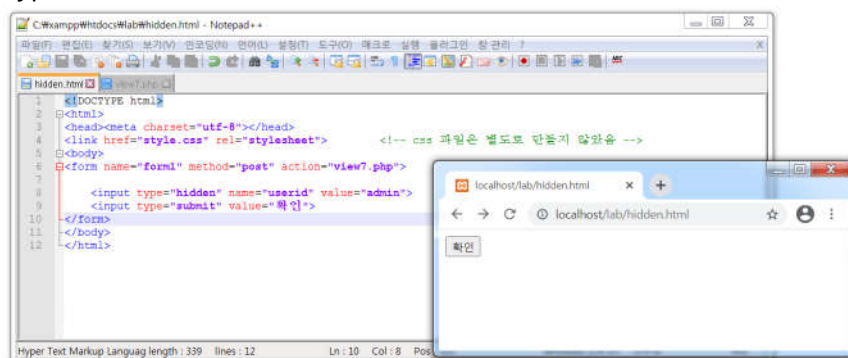
- 선택 박스에서 선택된 데이터 처리
 - 이메일 선택 박스에서 선택된 데이터 처리하기



81

form과 PHP (POST 방식)

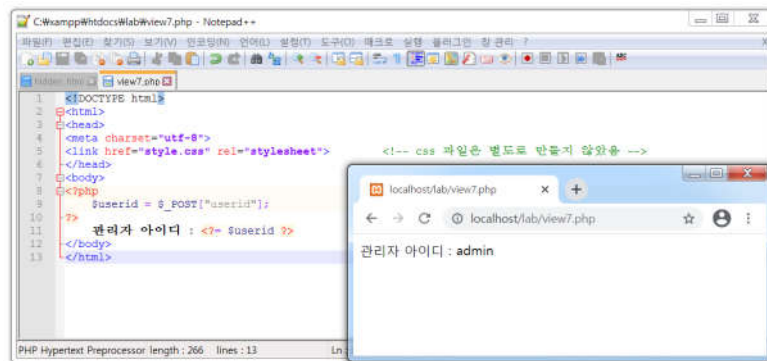
- hidden으로 입력된 데이터 처리
 - <input type="hidden"..> 은 웹 페이지를 표시하지 않고 숨김 상태로 웹 서버에 데이터를 전달



82

form과 PHP (POST 방식)

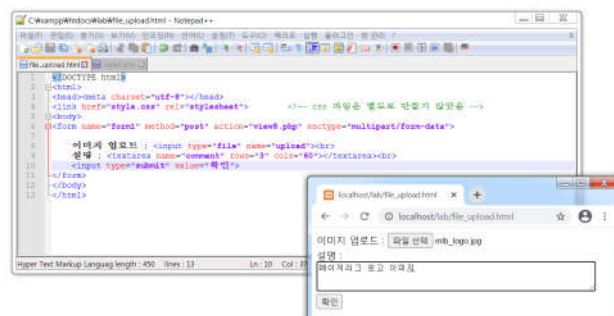
- hidden으로 입력된 데이터 처리
 - hidden 으로 입력된 데이터 처리하기



83

form과 PHP (POST 방식)

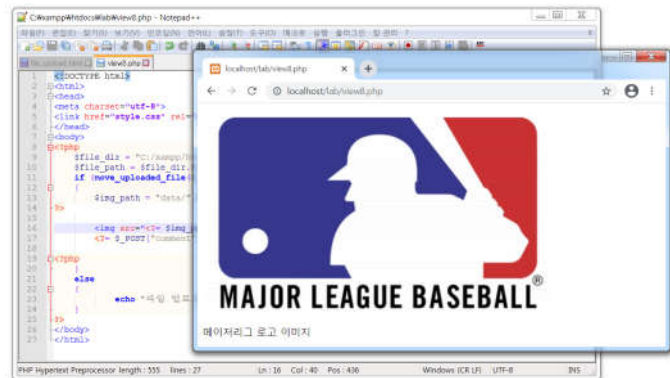
- 파일 업로드
 - 파일을 업로드 하려면 <form> 태그의 enctype 속성을 multipart/form-data 로 설정
 - <input> 태그의 type 속성을 file 로 설정해야 파일 선택가능



84

form과 PHP (POST 방식)

- 파일 업로드
 - 파일 업로드 처리하기



85

form과 PHP (POST 방식)

- 파일 업로드
 - 파일 업로드 처리하기
 - 업로드된 이미지 파일을 저장할 경로가 \$file_dir 에 설정된다
 - \$ _FILES ["upload"] ["name"] 은 업로드된 파일 이름을 나타낸다. \$file_path 에는 저장할 폴더와 파일 이름을 포함한 경로가 저장된다
 - move_up...file_path) 는 업로드된 파일을 설정한 파일 경로에 저장한다는 의미이다. \$ _FILE ["upload"] ["tmp_name"] 은 서버에 임시로 저장된, 업로드된 파일명이고, \$file_path 에는 ①에서 설정한, 업로드된 파일을 저장할 경로를 의미한다

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<link href="style.css" rel="stylesheet" <!-- css 파일은 별도로 만들지 않았음 -->
</head>
<body>
<?php
    $file_dir = "C:/xampp/htdocs/lab/data/"; ①
    $file_path = $file_dir.$_FILES["upload"]["name"]; ②
    if (move_uploaded_file($_FILES["upload"]["tmp_name"], $file_path)) ③
    {
        $img_path = "data/".$_FILES["upload"]["name"]; ④

        <br> ⑤
        <?=$_POST["comment"]?> ⑥
    }
    else
    {
        echo "파일 업로드 오류 발생";
    }
}
</body>
</html>
```

86

form과 PHP (POST 방식)

■ 파일 업로드

■ 파일 업로드 처리하기

- ④ \$img_path 에는 ⑤ 에서 태그의 src 속성으로 설정한 폴더와 이미지 파일명을 포함한 경로를 의미한다
- ⑤ 업로드된 이미지를 태그를 이용하여 화면에 출력한다
- ⑥ \$_POST["comment"] 는 앞서 html 에서 설정한 comment, 즉 사용자가 텍스트 영역에 입력한 내용을 의미한다.

■ move_uploaded_file()

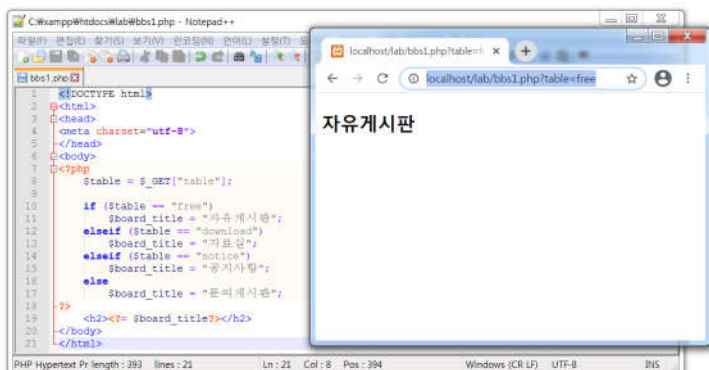
- move_uploaded_file() 함수의 형식은 "move_uploaded_file(업로드된파일명, 저장할경로)
- \$_FILES["upload"]["name"] : 업로드된 파일명
- \$_FILES["upload"]["type"] : 업로드된 파일의 데이터 형
- \$_FILES["upload"]["size"] : 업로드된 파일의 크기(바이트)
- \$_FILES["upload"]["tmp_name"] : 서버에 저장된 파일명
- \$_FILES["upload"]["error"] : 업로드 시 발생한 오류 코드

87

form과 PHP (GET 방식)

■ GET 방식으로 데이터 전달하기

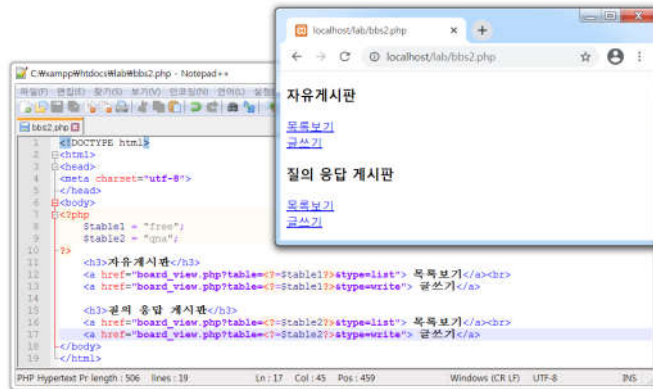
- 파일명 뒤에 ?table=free 가 있는데 이는 table 키를 이용하여 free 라는 데이터를 bbs1.php 에 전달 한다는 의미
- URL? 키1=값&키2=값 ...



88

form과 PHP (GET 방식)

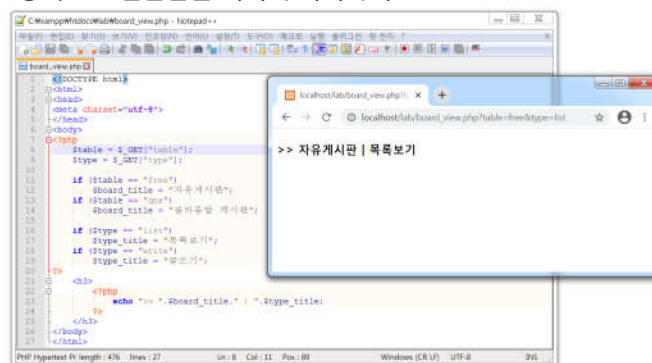
- 링크시 GET 방식으로 데이터 전달하기
 - 파일 내용과 결과 페이지



89

form과 PHP (GET 방식)

- 링크시 GET 방식으로 데이터 전달하기
 - 링크시 GET 방식으로 전달받은 데이터 처리하기



90

기타

- 쿠키
- 세션
- phpMyAdmin

쿠키

▪ 쿠키의 목적

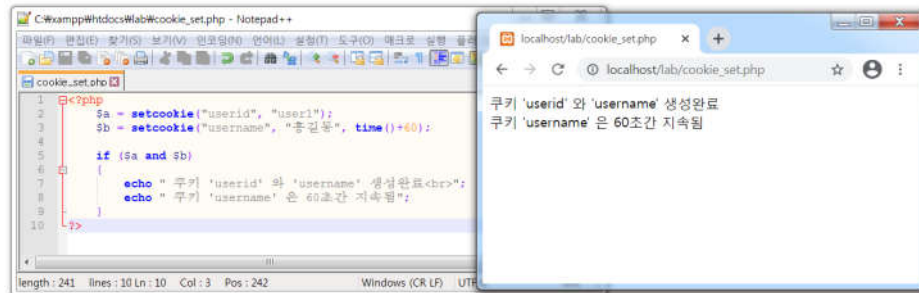
- 특정 웹 사이트를 재방문, 웹 사이트 내의 다른 페이지로 이동할 때 재 로그인 할 필요없음
- 사용자의 컴퓨터에서 아이디나 패스워드를 기억
- 사용자의 웹페이지 이용 패턴 분석



- 웹 서버는 사용자의 컴퓨터에 쿠키를 저장하고 필요할 때 사용자의 컴퓨터에 요청하고 사용자의 컴퓨터는 쿠키를 웹 서버에 전송한다

쿠키

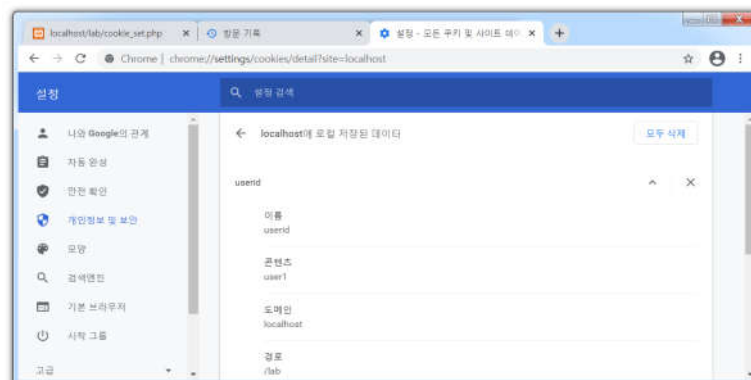
■ 쿠키생성 예제



93

쿠키

■ 생성된 쿠키 확인



94

세션

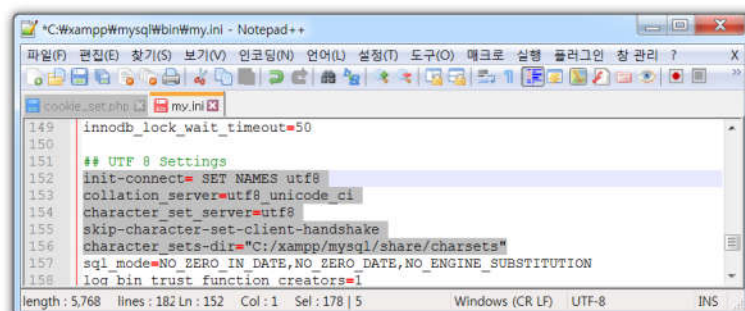
■ 세션의 정의

- 보안을 이유로 사용자의 컴퓨터와 웹 서버에 모두 정보를 저장
- 웹 사이트를 방문하는 사용자의 컴퓨터에는 세션ID 정보를 저장, 웹 서버에는 사용자의 컴퓨터 세션 ID에 대응되는 세션 정보를 저장
- 사용자의 컴퓨터 세션 ID가 유출되더라도 별다른 정보가 없고, 주요 정보가 웹 서버에 저장되어 있기 때문에 쿠키보다 보안성이 강하다

phpMyAdmin

■ MySQL 서버 구동

- MySQL에서 문자셋을 UTF-8로 설정하기 위해 C:\xampp\mysql\bin 내에 있는 my.ini 파일을 아래와 같이 수정한 후 xampp에서 mysql을 실행한다



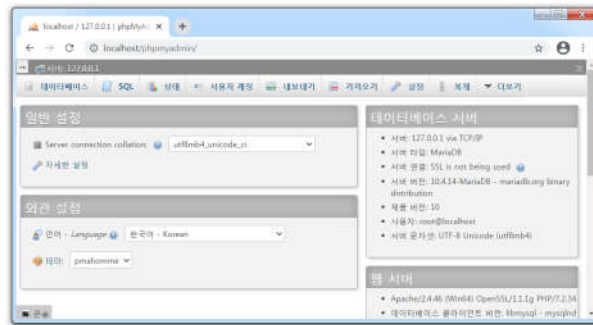
```

149 innodb_lock_wait_timeout=50
150
151 ## UTF 8 Settings
152 init-connect= SET NAMES utf8
153 collation_server=utf8_unicode_ci
154 character_set_server=utf8
155 skip-character-set-client-handshake
156 character_sets_dir="C:/xampp/mysql/share/charsets"
157 sql_mode=NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGINE_SUBSTITUTION
158 log_bin trust function creators=1
  
```


phpMyAdmin

■ phpMyAdmin 시작

- <http://localhost/dashboard> 에서 우측 상단의 메뉴중 "phpMyAdmin" 클릭

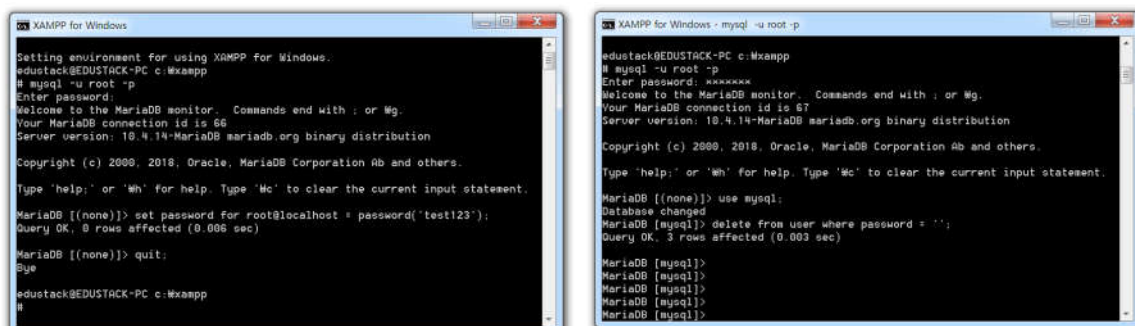


97

phpMyAdmin

■ phpMyAdmin 시작

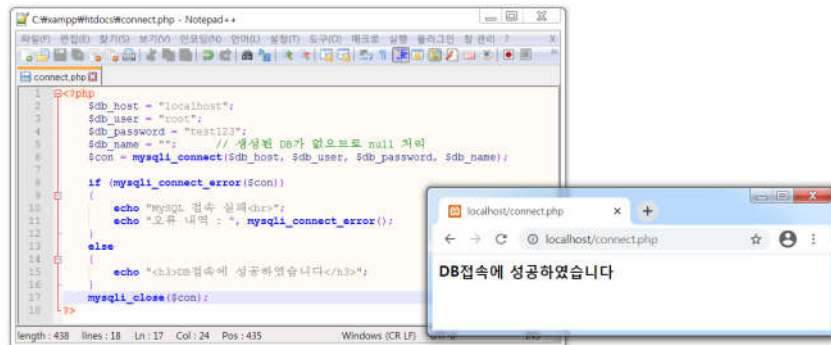
- 보안을 위해 root 패스워드를 설정하고, 익명사용자 삭제



98

phpMyAdmin

- phpMyAdmin 시작
 - php와 MySQL의 연동 확인하기

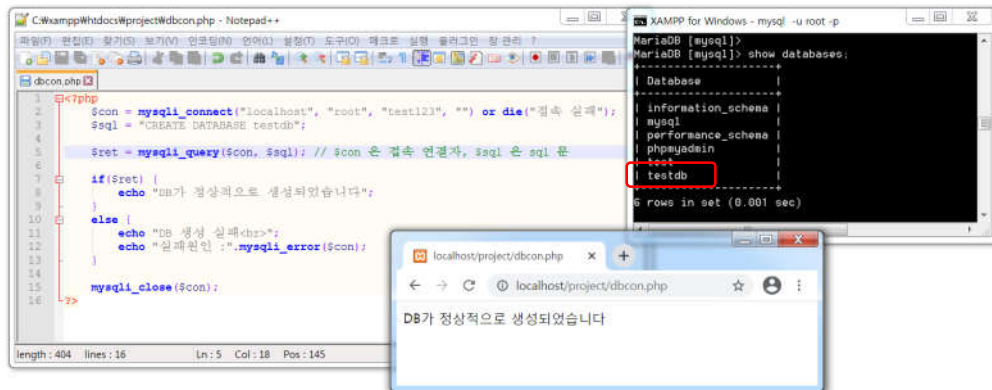


99

간단 프로젝트

간단 프로젝트 (사전 준비)

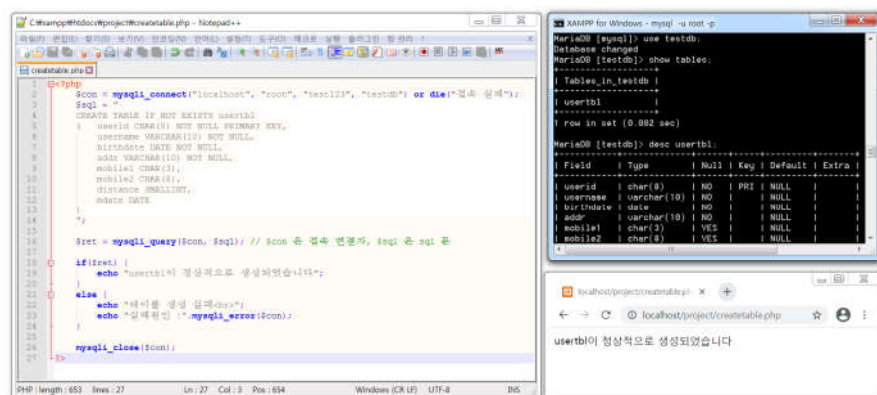
■ 데이터베이스 생성



101

간단 프로젝트 (사전 준비)

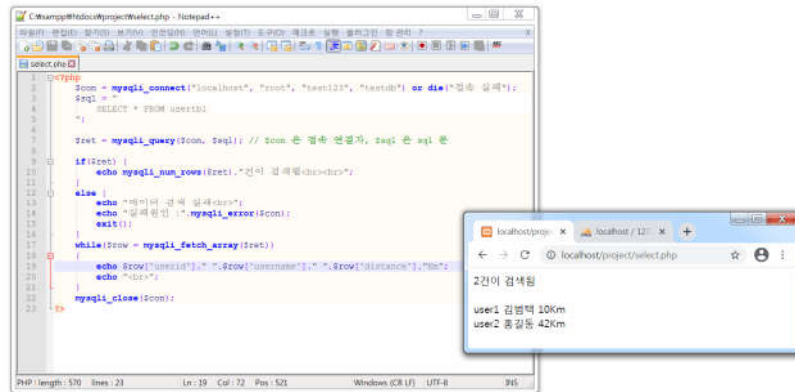
■ 데이터베이스 생성



102

간단 프로젝트 (사전 준비)

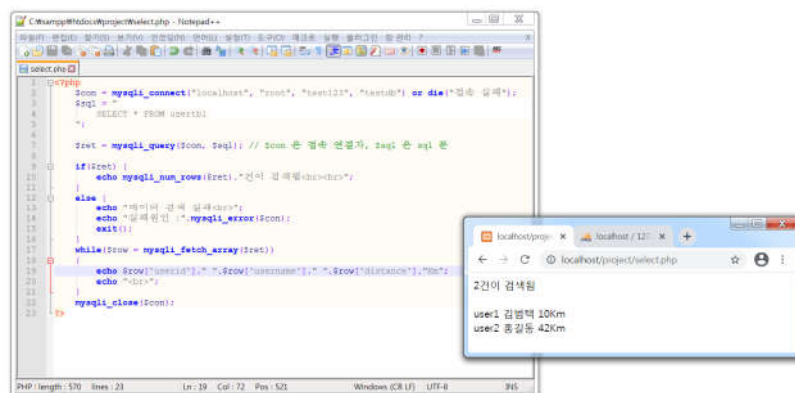
- phpMyAdmin 을 이용하여 데이터를 입력한 뒤, 데이터 검색



103

간단 프로젝트 (사전 준비)

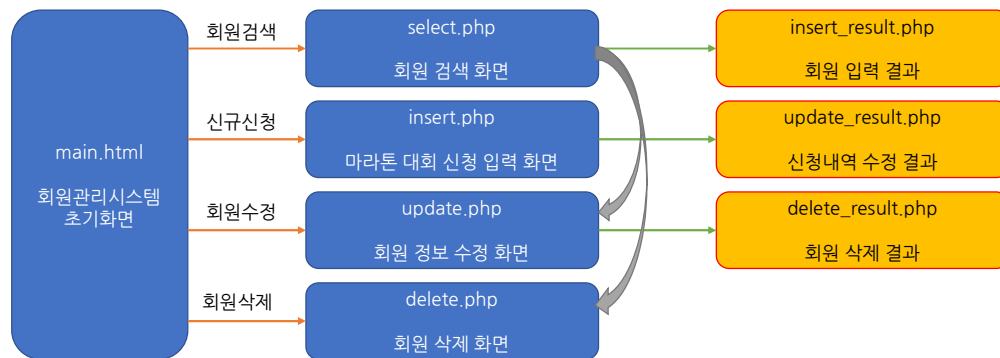
- phpMyAdmin 을 이용하여 데이터를 입력한 뒤, 데이터 검색



104

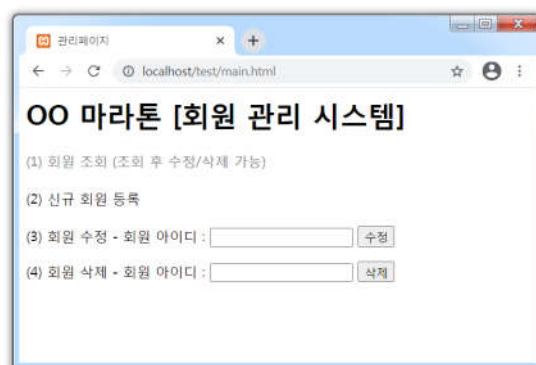
간단 프로젝트 (계획수립)

▪ OO 마라톤 회원관리 시스템의 구조



간단 프로젝트 (초기화면)

▪ 회원관리 시스템



간단 프로젝트 (회원검색)

- 회원검색 클릭시 전체 가입 회원의 정보를 조회

아이디	이름	출생연도	지역	국번	전화번호	등록(거리)	가입일	수정	삭제
user1	김병태	1978-03-06	경기 용인시	010	63134166	10	2021-01-07	수정	삭제
user2	홍길동	1990-12-25	서울 관악구	010	12345678	42	2021-01-05	수정	삭제

[<< 추가 화면](#)

간단 프로젝트 (회원가입, 마라톤 신청)

- 신규회원 등록화면을 통해 마라톤 신청

마라톤 신청 페이지(신규가입)

아이디 :

이름 :

출생연도 :

지역 :

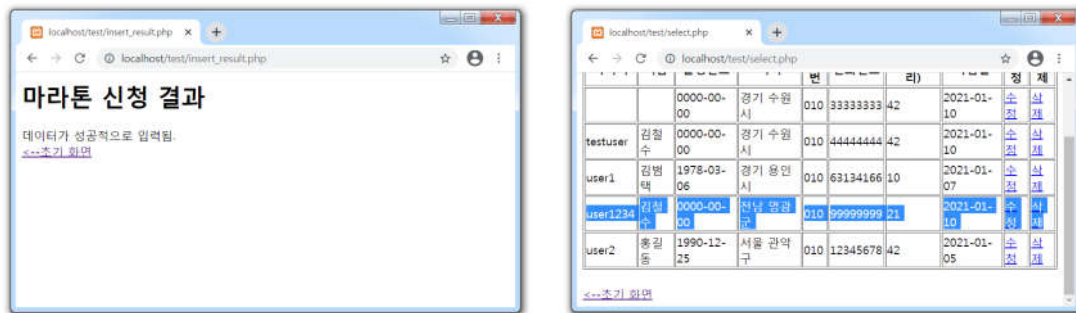
휴대폰 국번 :

휴대폰 전화번호 :

신청거리 :

간단 프로젝트 (회원가입, 마라톤 신청 결과)

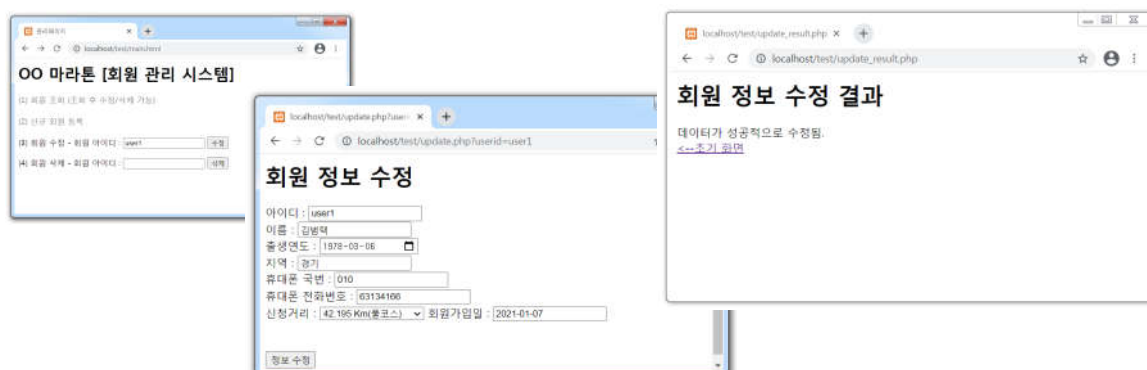
- 신규회원 등록화면을 통해 마라톤 신청 결과 페이지



109

간단 프로젝트 (회원정보 수정)

- 마라톤 신청 정보 수정페이지



110

간단 프로젝트 (회원정보 삭제)

■ 마라톤 신청 정보 삭제페이지

