

13

CHAPTER

데이터베이스 연동을 위한 PHP 프로그래밍



C.ontents

- 01** PHP 개발 환경 구축
- 02** 스트립트 언어와 HTML 태그
- 03** PHP 기본 문법
- 04** HTML과 PHP의 관계

1-1 PHP 프로그래밍의 개요

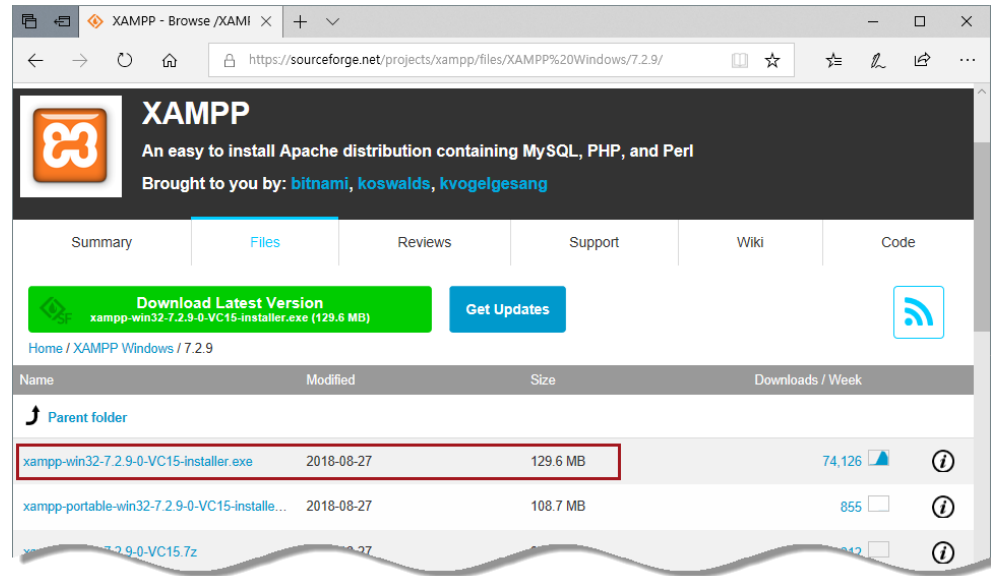
- 웹 사이트 구축
 - Apache, MySQL, PHP는 줄여서 APM이라고 함
 - Apache는 웹 서버 기능을, MySQL은 데이터베이스 기능을, PHP는 웹 프로그래밍 기능을 제공
 - 많은 웹 사이트는 이 세 가지를 조합한 환경으로 웹 서비스를 제공
- MySQL 설치 동영상(실습 13-1)
<https://www.youtube.com/watch?v=vAy4079h4yU&t=18s>

[실습 13-1] XAMPP 7.2.9 버전 설치하기

교재 449~457p 참고

1 XAMPP 설치하기

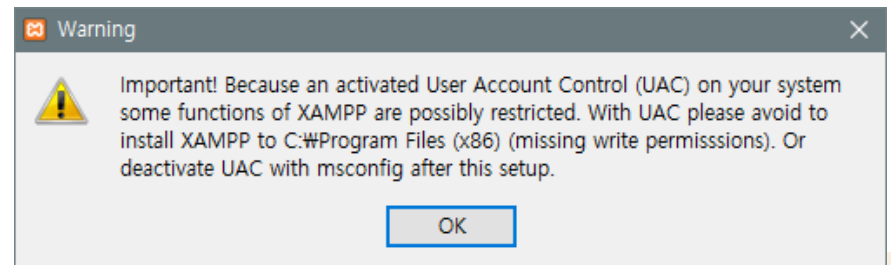
1-1 XAMPP 7.2.9 버전(xampp-win32-7.2.9-0-VC15-installer.exe, 129.6 MB) 다운로드



1-2 로고 확인



1-3 [Warning] 창이 나오면 <OK> 클릭

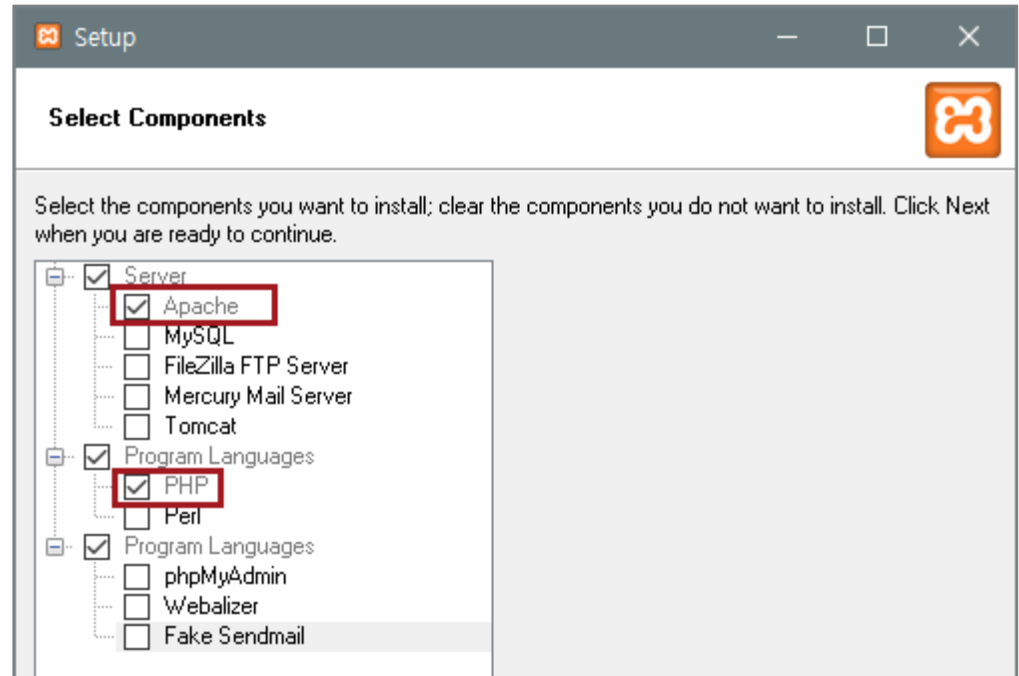


[실습 13-1] XAMPP 7.2.9 버전 설치하기

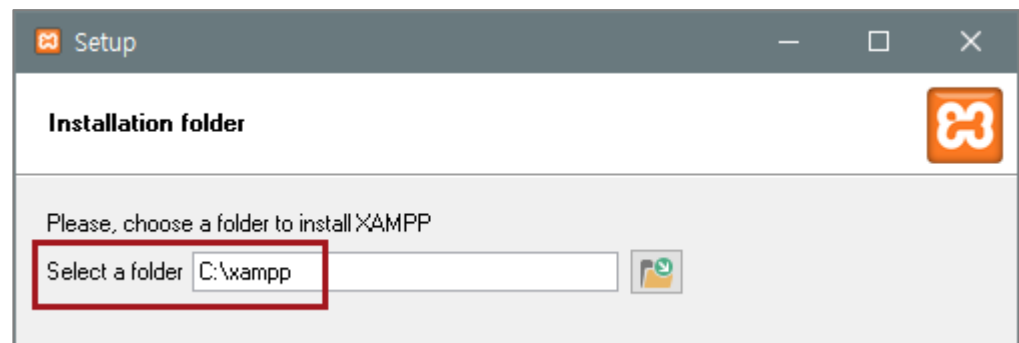
교재 449~457p 참고

1-4 [Setup - XAMPP]에서 <Next> 클릭

1-5 설치할 제품 선택



1-6 그대로 두고 <Next> 클릭



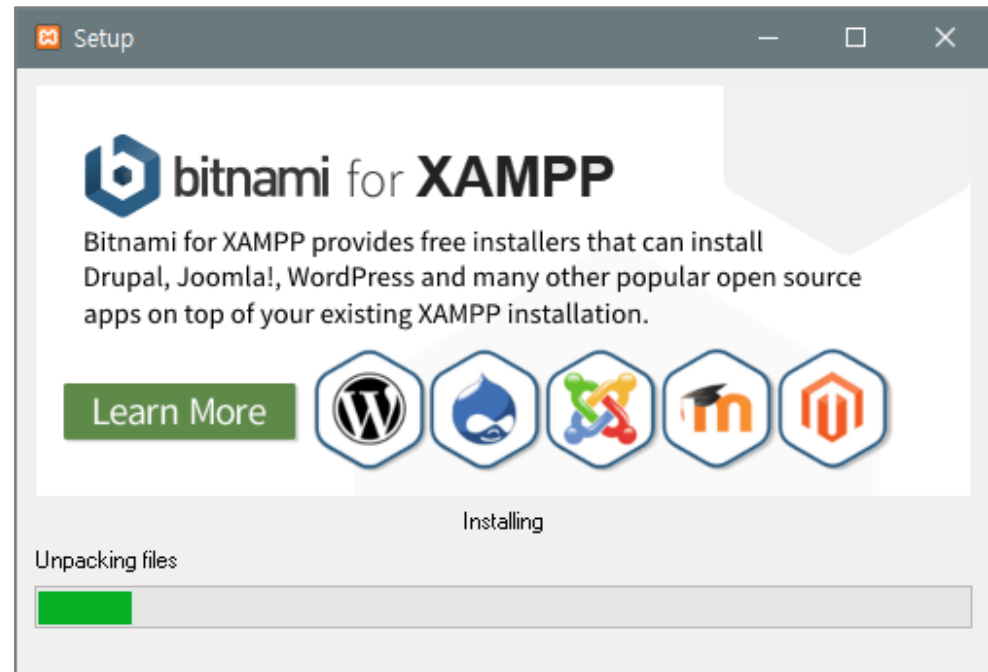
[실습 13-1] XAMPP 7.2.9 버전 설치하기

교재 449~457p 참고

1-7 ' Learn more about Bitnami for XAMPP'의 체크를 해제



1-8 설치 진행



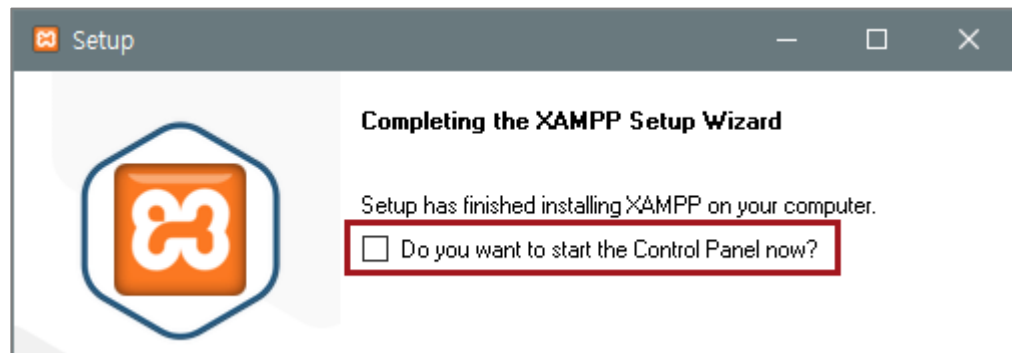
[실습 13-1] XAMPP 7.2.9 버전 설치하기

교재 449~457p 참고

1-9 모두 체크 표시를 하고 <엑세스 허용>
을 클릭



1-10 ' Do you want to start the Control
Panel now?'의 체크 해제



[실습 13-1] XAMPP 7.2.9 버전 설치하기

교재 449~457p 참고

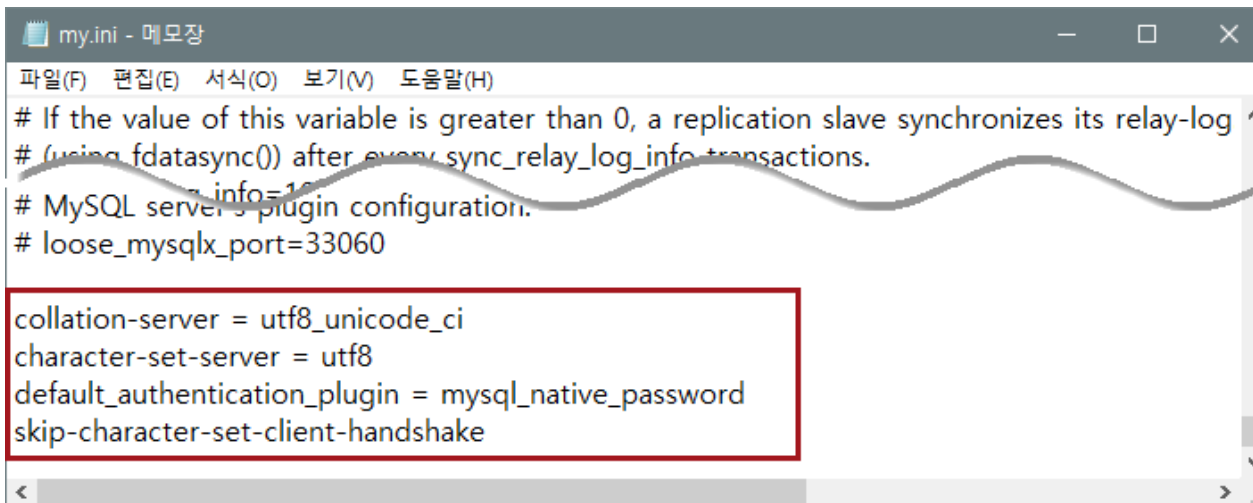
2 MySQL 서버의 설정 변경하기

2-1 메모장에서 my.ini 파일을 엽

```
cmd
CD %PROGRAMDATA%
CD MySQL
CD "MySQL Server 8.0"
NOTEPAD my.ini
```

2-2 다음 4줄 추가

```
collation-server = utf8_unicode_ci
character-set-server = utf8
default_authentication_plugin = mysql_native_password
skip-character-set-client-handshake
```



[실습 13-1] XAMPP 7.2.9 버전 설치하기

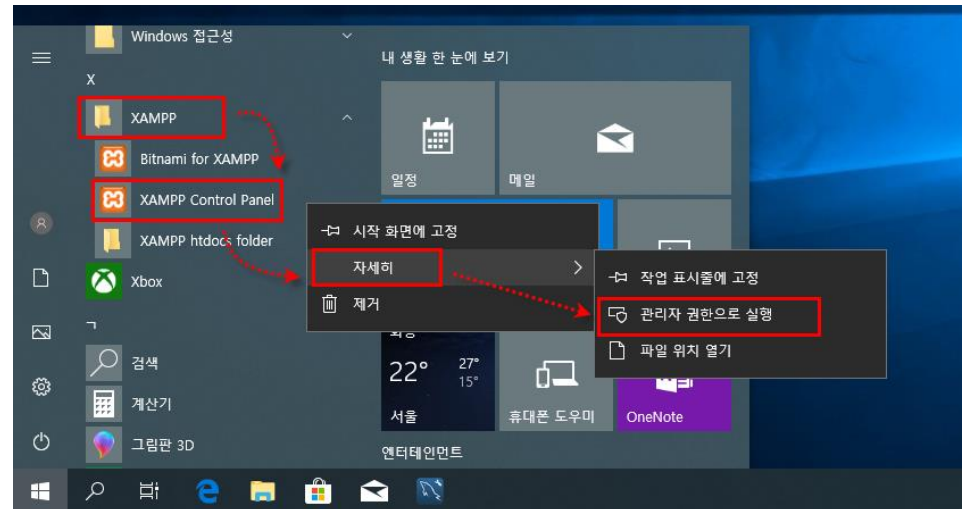
교재 449~457p 참고

2-3 재부팅

2-4 [보안 경고] 창이 나오면 <예> 클릭

3 XAMPP 환경 설정하기

3-1 [XAMPP Control Panel]을 관리자 권한으로 실행
으로 실행

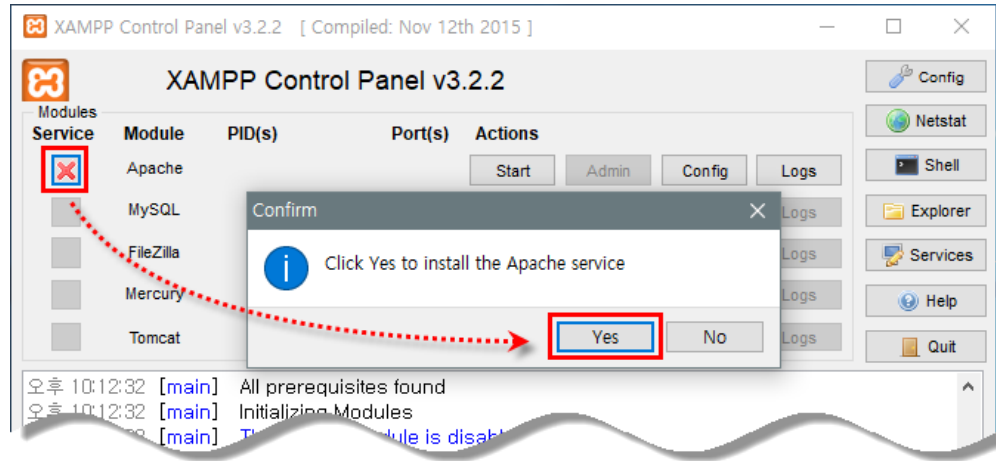


3-2 영어 그림을 선택하고 <Save> 클릭

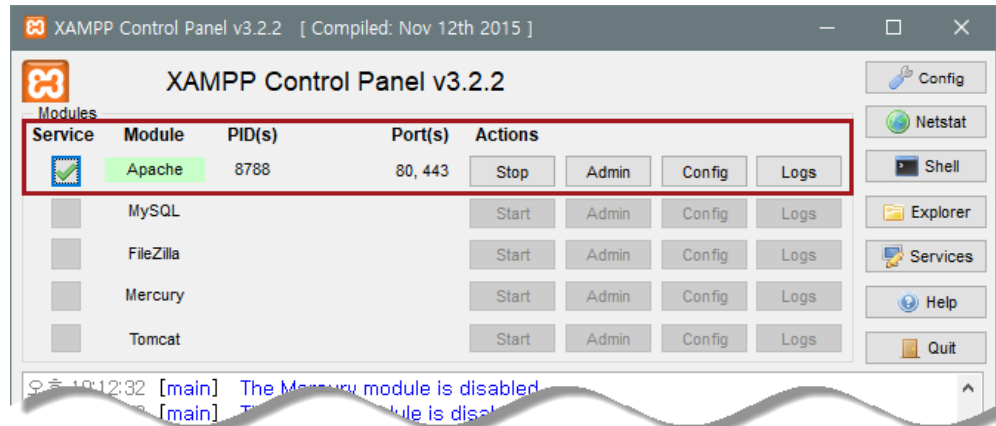
[실습 13-1] XAMPP 7.2.9 버전 설치하기

교재 449~457p 참고

3-3 Windows의 [서비스]에 Apache 서비스 등록



3-4 Apache 서비스 시작



3-5 [XAMPP Control Panel v3.2.2] 창 닫기

[실습 13-1] XAMPP 7.2.9 버전 설치하기

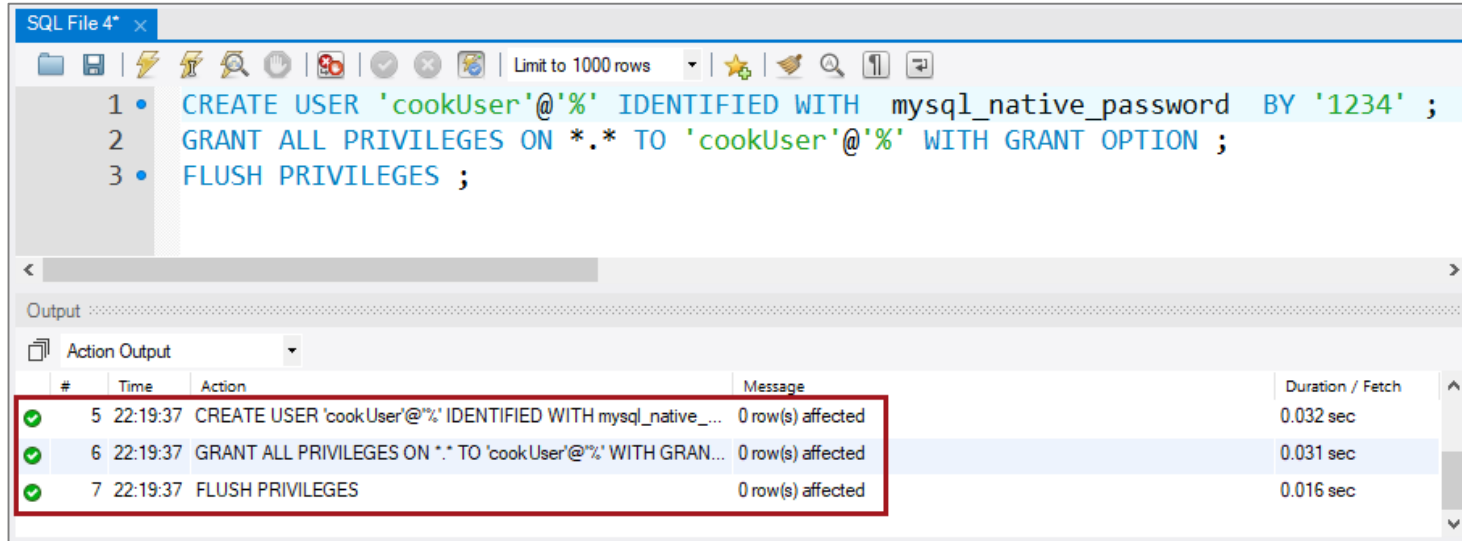
교재 449~457p 참고

4 PHP로 접속할 전용 사용자 생성하기

4-1 Workbench를 실행하여 root로 접속

4-2 용자 cookUser(비밀번호: 1234) 생성

```
CREATE USER 'cookUser'@'%' IDENTIFIED WITH mysql_native_password BY '1234';  
GRANT ALL PRIVILEGES ON * . * TO 'cookUser'@'%' WITH GRANT OPTION;  
FLUSH PRIVILEGES;
```



4-3 Workbench 종료

[실습 13-1] XAMPP 7.2.9 버전 설치하기

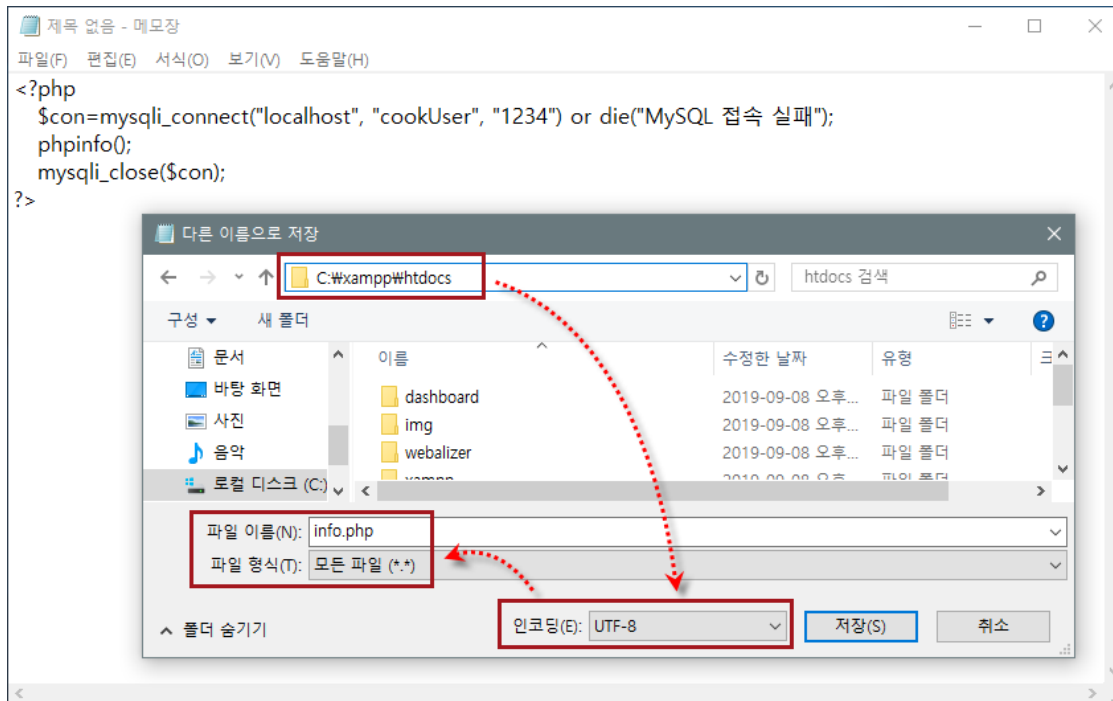
교재 449~457p 참고

5 사이트 작동 확인하기

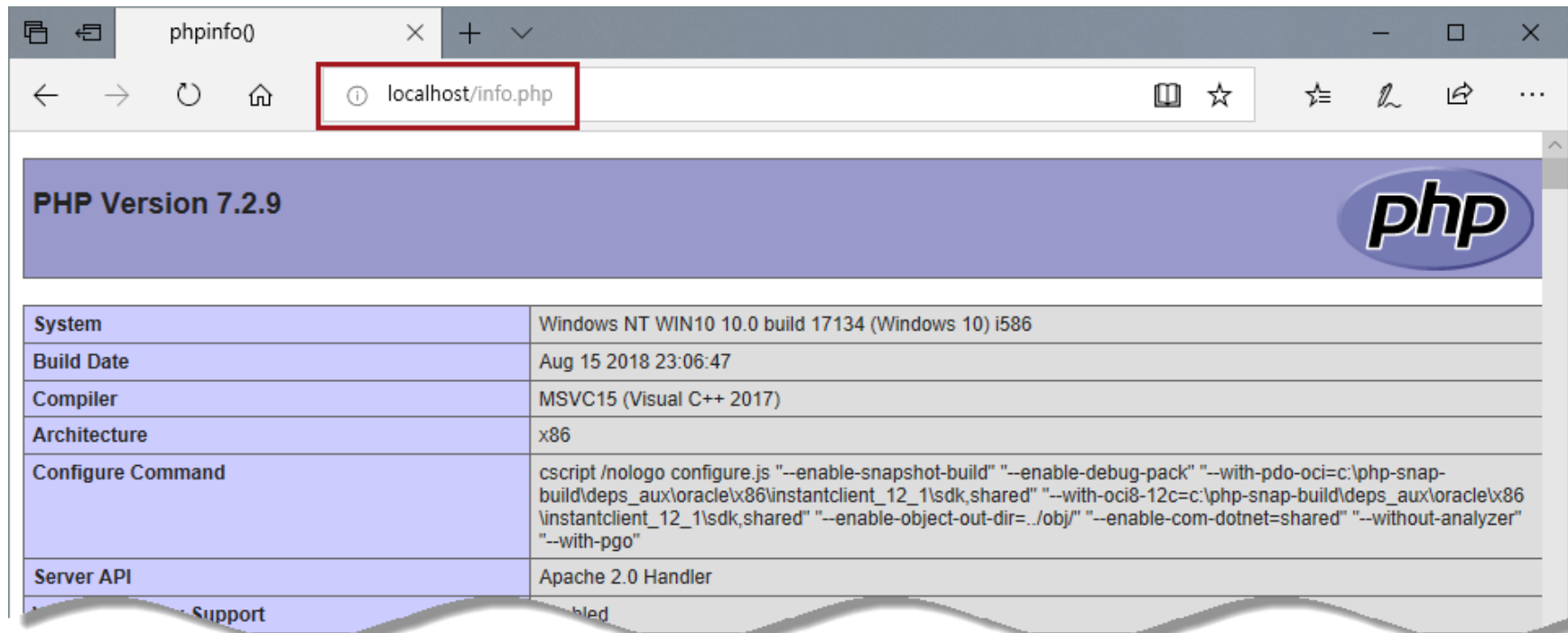
5-1 다음과 같이 코딩

```
<?php
$con=mysqli_connect("localhost", "cookUser", "1234") or die("MySQL 접속 실패");
phpinfo();
mysqli_close($con);
?>
```

5-2 info.php 파일 저장



5-3 Apache, PHP, MySQL 확인



The screenshot shows a web browser window with the address bar displaying `localhost/info.php`. The page content includes the PHP logo and the text "PHP Version 7.2.9". Below this is a table with system configuration details.

System	Windows NT WIN10 10.0 build 17134 (Windows 10) i586
Build Date	Aug 15 2018 23:06:47
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x86
Configure Command	<code>cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=c:\php-snap-build\deps_aux\oracle\x86\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-snap-build\deps_aux\oracle\x86\instantclient_12_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"</code>
Server API	Apache 2.0 Handler
Support	abled

2-1 서버 스크립트와 클라이언트 스크립트

■ 클라이언트 스크립트

- 소스코드는 클라이언트의 웹 브라우저에서 해석됨
- 웹 서버는 클라이언트가 요청한 소스코드를 변경하지 않고 고스란히 전송하는 역할만 함
- 대표적인 클라이언트 스크립트 언어 : 자바스크립트

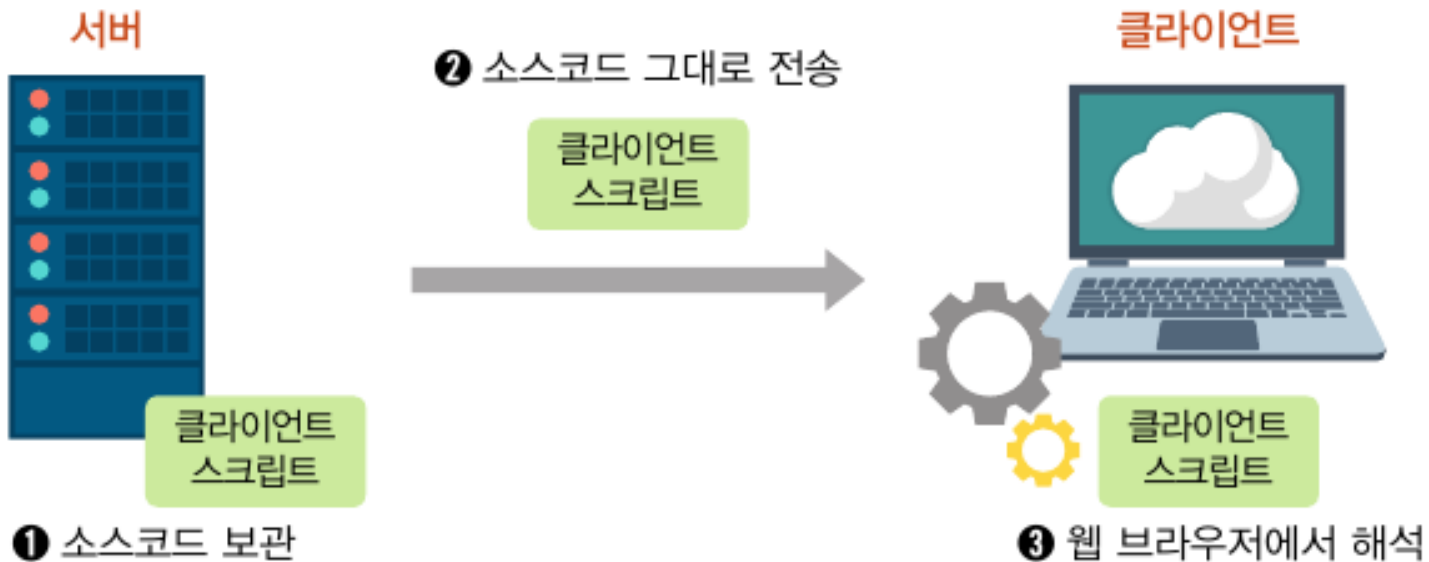


그림 13-18 클라이언트 스크립트의 개념

2-1 서버 스크립트와 클라이언트 스크립트

■ 서버 스크립트

- 클라이언트가 요청할 때마다 새로 해석한 소스코드를 클라이언트에 전송
- 소스코드를 작성할 때 그 내용이 실시간으로 변경되도록 프로그래밍해야 함
- 실시간으로 변경되는 웹 사이트를 동적인 웹 사이트(dynamic web site)라고 함
- 대표적인 서버 스크립트 언어 : PHP, JSP, ASP.net 등



그림 13-19 서버 스크립트의 개념

2-2 HTML 태그

■ HTML 파일의 특징

- HTML 파일의 확장자는 * .htm 또는 * .html
- HTML 파일은 텍스트 파일이므로 메모장 등의 편집기에서 작성하면 됨(단, 웹 브라우저에서 한글이 깨져 보일 수 있기 때문에 인코딩 방식을 UTF-8로 설정)
- HTML 파일은 <HTML> 태그로 시작해서 </HTML> 태그로 종료
- HTML은 대문자와 소문자를 구분하지 않음

■ HTML 파일의 기본 형식

```
<HTML>
<HEAD>
  화면에 표시되지 않는 정보(타이틀, 인코딩 정보 등)
</HEAD>

<BODY>
  화면에 보이는 본체(주로 태그 표현)
</BODY>
</HTML>
```


2-2 HTML 태그

■ <META>

- 웹 페이지의 각종 정보를 설정
- 검색 엔진에 문서의 내용을 요약
- 웹 페이지에서 사용하는 언어의 인코딩을 설정
- <HEAD> 영역에 작성

```
<META http-equiv="content-type" content="text/html; charset=utf-8">
```

■

- 줄 바꿈
- 다음과 같이 작성하면 '안녕하세요?'와 'MySQL 학습 중입니다.'가 2줄로 출력

```
안녕하세요? <BR> MySQL 학습 중입니다.
```

■ <U>...</U>, ..., <I>...</I>

- 글자에 밑줄, 굵은 글씨, 이탤릭체 속성 적용

```
<U>이건 밑줄</U> <BR> <B>이건 굵게</B> <BR> <I>이건 이탤릭</I>
```

■ ...

- 글자의 크기나 색상을 지정
- 다음과 같이 작성하면 글자의 색상은 빨간색, 폰트는 궁서, 크기는 10이 적용됨

```
<FONT color='red' size='10' face='궁서'>폰트 변경했어요.</FONT>
```

2-2 HTML 태그

■ <HR>

- 수평선을 그음
- 다음과 같이 작성하면 폭이 10픽셀인 선이 출력

```
<HR size='10'>
```

■ <A>...

- 클릭하면 다른 페이지가 연결되는 링크를 설정하는데 href 속성에 연결된 홈페이지를 지정
- target 속성을 별도로 지정하지 않으면 현재 페이지에 링크된 페이지가 열림
- 다음과 같이 작성한 링크를 클릭하면 새 페이지에서 www.hanbit.co.kr이 열림

```
<A href='http://www.hanbit.co.kr' target='_blank'>한빛 홈페이지 연결</A>
```

■

- 그림 파일을 화면에 출력
- 다음과 같이 작성하면 mouse.png 파일이 화면에 100×100 픽셀 크기로 출력
- width와 height를 생략하면 원래 그림 크기로 출력

```
<IMG src='mouse.png' width=100 height=100>
```

2-2 HTML 태그

- <TABLE>...</TABLE>, <TR>...</TR>, <TH>...</TH>, <TD>...</TD>
 - 표를 만들
 - <TABLE>...</TABLE> 태그로 표를 만들고 <TR>...</TR> 태그로 행을 만들
 - 행 안의 열은 <TH>...</TH> 태그 또는 <TD>...</TD> 태그로 만들
 - <TH> 태그는 제목 열을 만들어 글자를 진하게 출력
 - <TD> 태그는 일반 열을 만들
 - 다음과 같이 작성하면 제목에 아이디와 이름이 있는 2행 2열의 표가 출력

```
<TABLE border=1>
<TR>
    <TH>아이디</TH>
    <TH>이름</TH>
</TR>
<TR>
    <TD>KHD</TD>
    <TD>강호동</TD>
</TR>
<TR>
    <TD>YJS</TD>
    <TD>유재석</TD>
</TR>
</TABLE>
```

3-1 변수

■ PHP 파일의 기본 구조

```
1 <?php
2
3 ?>
```

■ PHP의 주석

```
1 <?php
2 // 한 줄 주석용
3 /*
4   여러 줄
5   주석용
6 */
7 ?>
```

■ 변수

- 다음은 변수 a에 100을 대입하는 식
- 변수 이름 앞에는 \$를 붙이고 행 끝에는 세미콜론(;)을 붙임

```
$a = 100;
```

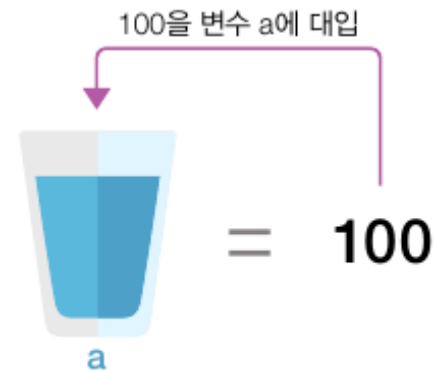


그림 13-20 변수에 값을 대입할 때 처리되는 방식

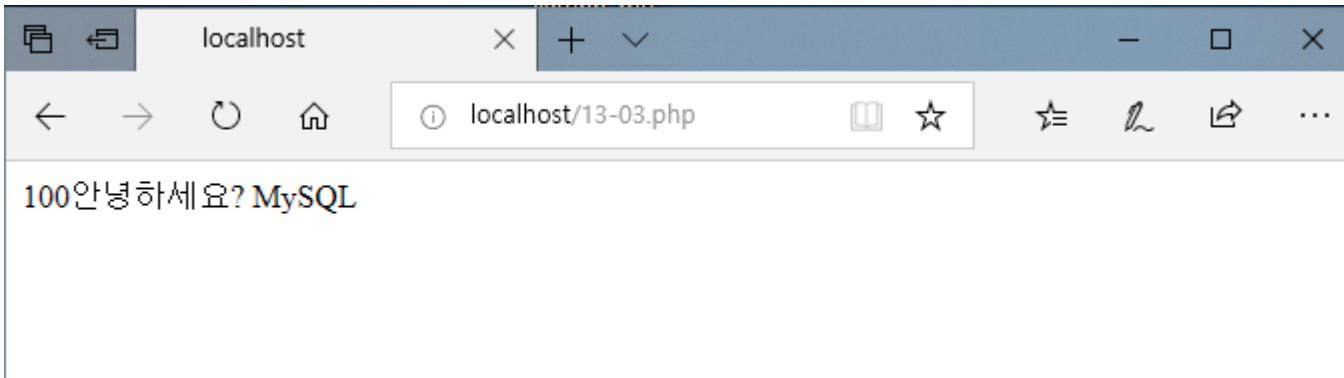
3-1 변수

■ 변수의 이름을 지을 때 규칙

- 맨 앞에 \$가 붙음
- 문자, 숫자, 밑줄(_)을 사용할 수 있지만 숫자로 시작할 수는 없음
예: \$abc(o), \$abc123(o), \$_abc(o), \$_abc123(o), \$123(x), \$123abc(x)
- 대·소문자를 구분. \$abc와 \$ABC는 다른 변수

■ 변수 출력

```
1 <?php
2  $a = 100;
3  print $a;
4
5  $b = "안녕하세요? MySQL";
6  echo $b;
7 ?>
```



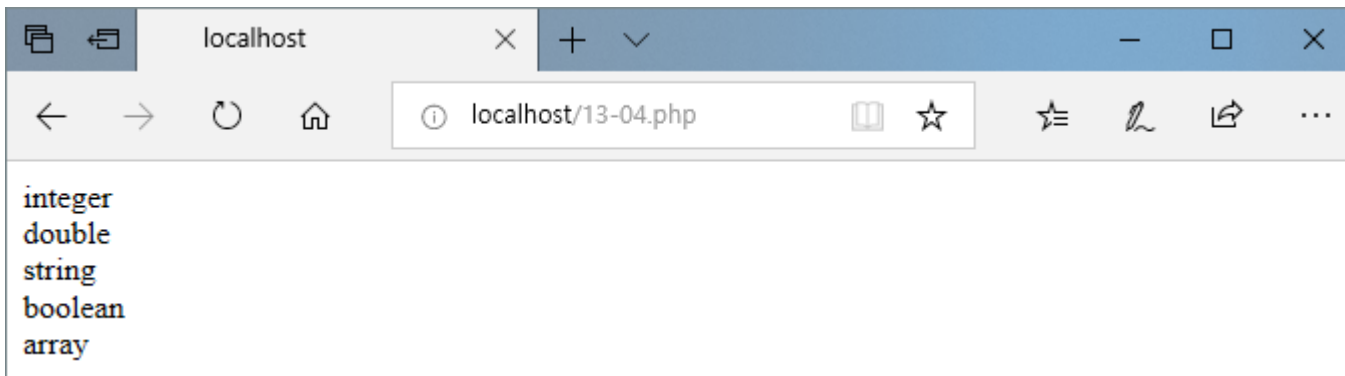
3-2 데이터 형식

■ PHP의 데이터 형식

- 정수형(int), 실수형(double), 문자열형(string), 불형(boolean), 객체 (object), 배열(array) 등이 있음
- PHP에서는 별도로 변수를 선언하지 않아도 변수에 값을 대입하는 순간 변수의 데이터 형식이 결정
- 변수를 사용하는 중 다른 데이터 형식의 값이 대입되면 새로운 값에 맞추어 변수의 데이터 형식이 바뀜

■ 변수의 데이터 형식

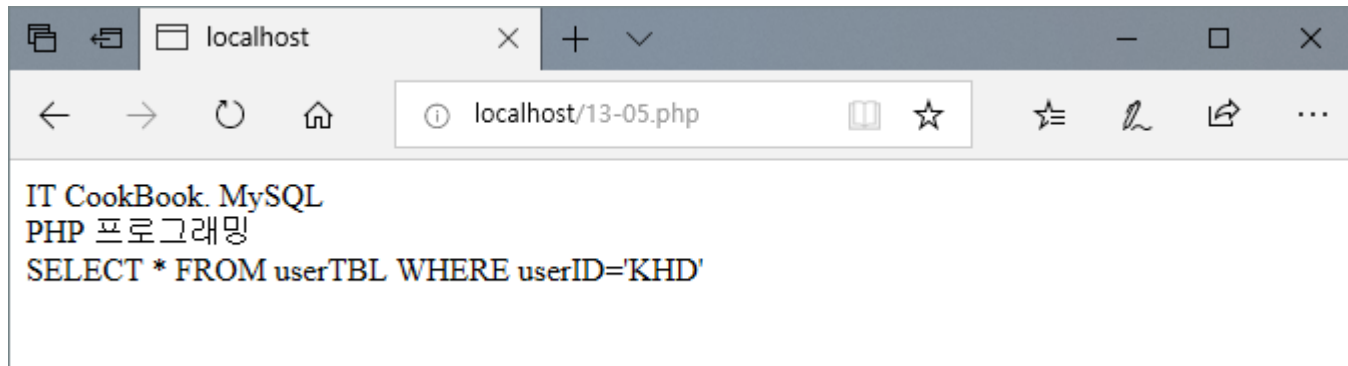
```
1 <?php
2 $a = 123; echo gettype($a), "<br>";
3 $a = 123.123; echo gettype($a), "<br>";
4 $a = "MySQL"; echo gettype($a), "<br>";
5 $a = true; echo gettype($a), "<br>";
6 $a = array(1, 2, 3); echo gettype($a), "<br>";
7 ?>
```



3-2 데이터 형식

■ 문자열을 변수에 대입

```
1 <?php
2 $str1 = "IT CookBook. MySQL <br>"; echo $str1;
3 $str2 = 'PHP 프로그래밍<br>'; echo $str2;
4 $str3 = "SELECT * FROM userTBL WHERE userID='KHD' "; echo $str3;
5 ?>
```



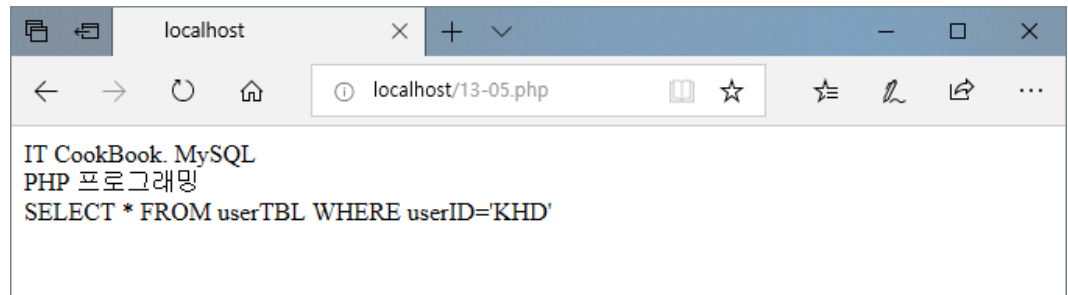
3-3 조건문

■ if ... else 문

- 조건에 따라서 분기할 때 사용
- else{ } 부분을 생략할 수 있음
- 조건식의 결과는 TRUE(참) 또는 FALSE(거짓)
- 조건식에는 비교 연산자인 ==(같다), < >(같지 않다), <(작다), >(크다), <=(작거나 같다), >=(크거나 같다) 등이 주로 사용

```
if(조건식) {  
    // 참일 때 실행  
} else {  
    // 거짓일 때 실행  
}
```

```
1 <?php  
2 $a = 100;  
3 $b = 200;  
4  
5 if($a > $b) {  
6     echo "a가 b보다 큽니다.";  
7 } else {  
8     echo "a가 b보다 작습니다.";  
9 }  
10 ?>
```

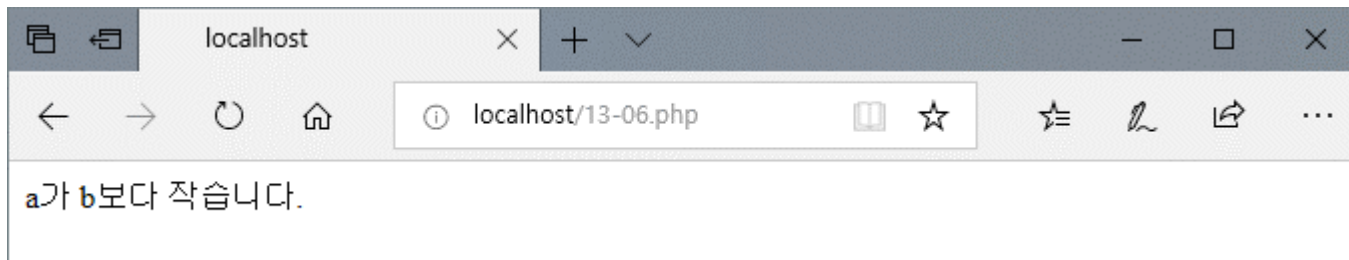


3-3 조건문

■ 여러 개의 조건 처리

- if ... elseif ... else 문 사용
- elseif{ } 부분에서 여러 개의 조건을 처리

```
1 <?php
2     $jumsu = 83;
3
4     if($jumsu >= 90) {
5         echo "A학점";
6     } elseif($jumsu >= 80) {
7         echo "B학점";
8     } elseif($jumsu >= 70) {
9         echo "C학점";
10    } elseif($jumsu >= 60) {
11        echo "D학점";
12    } else {
13        echo "F학점";
14    }
15 ?>
```



3-3 조건문

- switch ... case 문
 - 여러 개의 조건 처리
 - default 부분은 생략 해도 됨

```
switch(변수) {  
    case 값1 :  
        // 값1이면 이 부분을 처리  
        break;  
    case 값2 :  
        // 값2이면 이 부분을 처리  
        break;  
    ...  
    default:  
        // 어디에도 해당되지 않으면 이 부분을 처리  
}  

```

3-3 조건문

- [소스코드 13-7]을 switch ... case 문으로 작성

```
1 <?php
2     $jumsu = 83;
3
4     switch(intval($jumsu / 10)) {
5         case 10:
6         case 9:
7             echo "A학점"; break;
8         case 8:
9             echo "B학점"; break;
10        case 7:
11            echo "C학점"; break;
12        case 6:
13            echo "D학점"; break;
14        default:
15            echo "F학점";
16    }
17 ?>
```

3-4 반복문

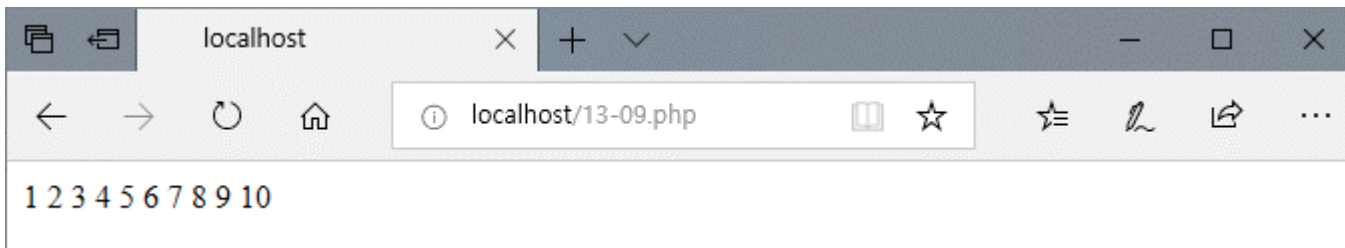
■ for 문

- 지정된 수만큼 반복

```
for(초깃값; 조건식; 증감식) {  
    // 이 부분을 반복  
}
```

■ for 문으로 1부터 10까지 출력하는 코드

```
1 <?php  
2     for($i=1; $i<=10; $i=$i+1) {  
3         echo $i, " ";  
4     }  
5 ?>
```



3-4 반복문

■ for 문의 실행 순서

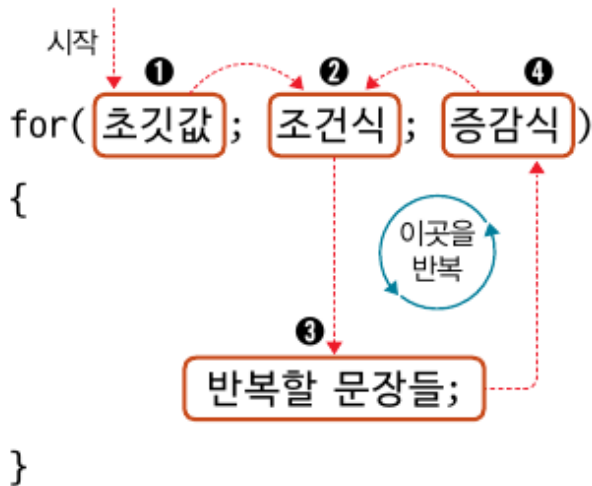
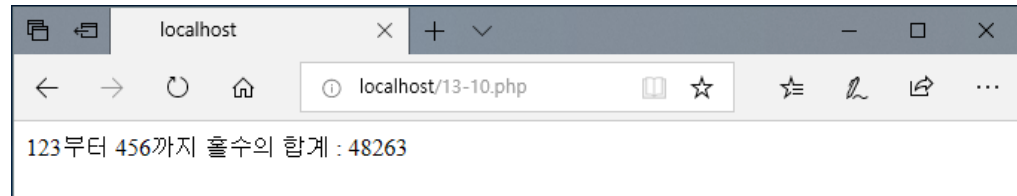


그림 13-27 for 문의 실행 순서

■ for 문으로 123부터 456까지 홀수의 합계(123+125+127+...+455)를 구하는 코드

```
1 <?php
2 $hap = 0;
3
4 for($i=123; $i<=456; $i=$i+2) {
5 $hap = $hap + $i;
6 }
7
8 echo "123부터 456까지 홀수의 합계 : ", $hap;
9 ?>
```



3-4 반복문

■ while 문

- 특정 명령을 반복할 때 사용
- 초기값과 증감식이 없고 조건식만 있음

```
while(조건식) {  
  // 이 부분을 반복  
}
```

■ while 문의 실행 순서

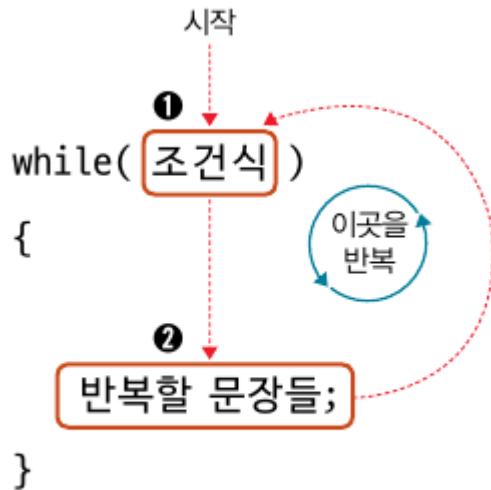


그림 13-29 while 문의 실행 순서

3-4 반복문

- while 문의 초깃값과 증감식 위치

```
초깃값;  
while(조건식) {  
    // 이 부분을 반복  
    증감식;  
}
```

- for 문으로 작성한 [소스코드 13-10]과 동일한 결과를 출력하는 코드

```
1 <?php  
2     $hap = 0;  
3  
4     $i = 123;  
5     while($i<=456) {  
6         $hap = $hap + $i;  
7         $i = $i + 2;  
8     }  
9  
10    echo "123부터 456까지 홀수의 합계 : ", $hap;  
11 ?>
```

3-5 배열

■ 배열

- 하나씩 사용하던 종이 상자(변수)를 한 줄로 붙여놓은 것과 같음
- 변수를 한 줄로 붙인 후 이름을 \$aa로 지정하여 사용
- 배열 내의 각 변수에는 \$aa[0], \$aa[1], \$aa[2], \$aa[3]과 같이 번호를 붙이는데, 이 번호를 첨자라고 하며 0부터 시작
- 변수 3개를 모아 하나의 배열을 만들었다면 첨자는 0, 1, 2

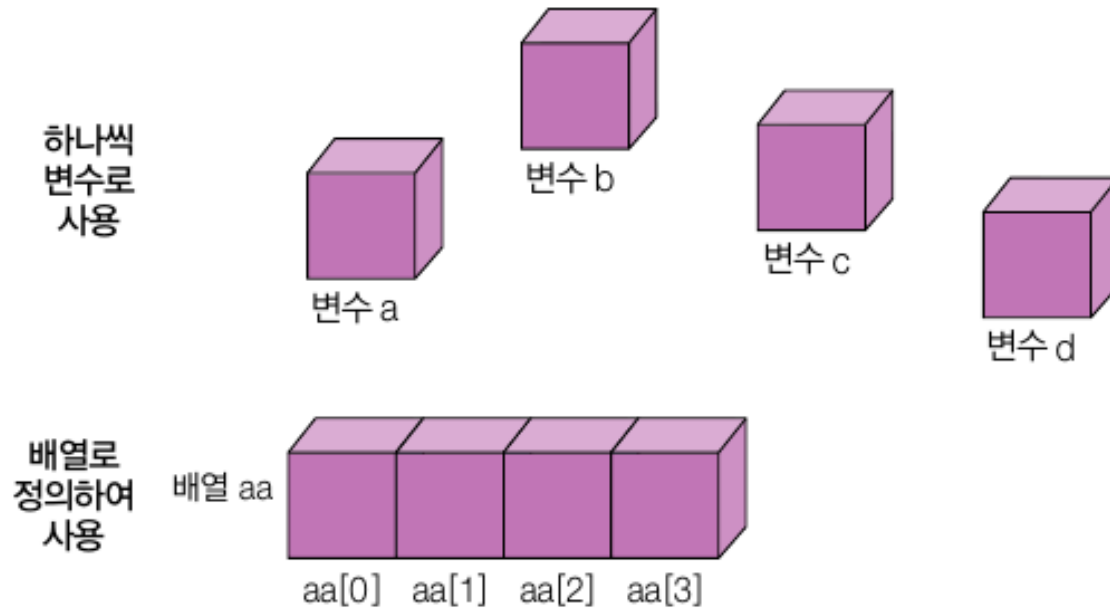


그림 13-30 배열의 개념

3-5 배열

- PHP에서 배열을 만드는 형식

- 형식 1

```
$배열명 = array(값1, 값2, 값3, ...);
```

- 형식 2

```
$배열명 = range(시작값, 끝값, 증가값);
```

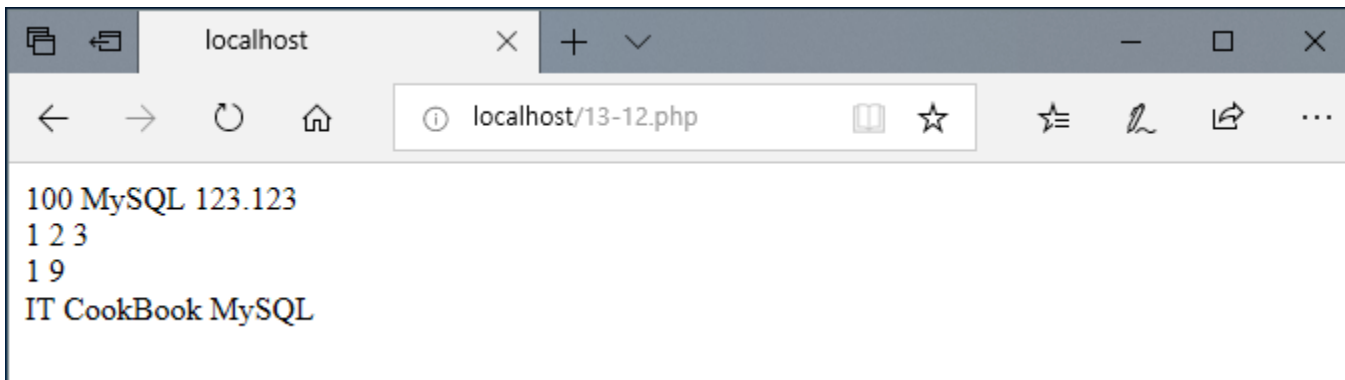
- 형식 3

```
$배열명[0] = 값1;  
$배열명[1] = 값2;  
$배열명[3] = 값3;  
..
```

3-5 배열

■ 배열을 만드는 간단한 예

```
1 <?php
2   $myArray = array(100, 'MySQL', 123.123);
3   echo $myArray[0], " ", $myArray[1], " ", $myArray[2], "<br>";
4
5   $myArray = range(1, 3);
6   echo $myArray[0], " ", $myArray[1], " ", $myArray[2], "<br>";
7
8   $myArray = range(1, 10, 2);
9   echo $myArray[0], " ", $myArray[4], "<br>";
10
11  $newArray[0] = 'IT';
12  $newArray[1] = 'CookBook';
13  $newArray[2] = 'MySQL';
14  echo $newArray[0], " ", $newArray[1], " ", $newArray[2], "<br>";
15 ?>
```



3-5 배열

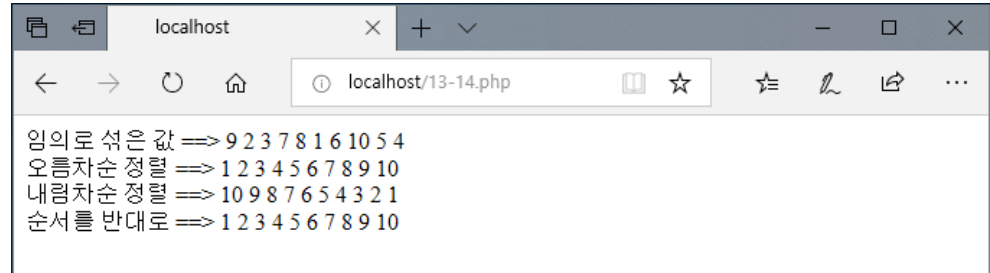
- 배열에 1부터 10까지 입력한 후 그 합계를 구하는 프로그램

```
1 <?php
2   $hap = 0;
3   $myArray = range(1, 10);
4
5   for($i=0; $i<10; $i++) {
6       $hap = $hap + $myArray[$i];
7   }
8   echo "배열의 합계 : " , $hap;
9 ?>
```

3-5 배열

■ 배열을 활용한 다양한 함수

```
1 <?php
2 $myArray = range(1, 10);
3
4 echo "임의로 섞은 값 ==> ";
5 shuffle($myArray);
6 foreach($myArray as $data)
7 echo $data, " ";
8
9 echo "<br>오름차순 정렬 ==> ";
10 sort($myArray);
11 foreach($myArray as $data)
12 echo $data, " ";
13
14 echo "<br>내림차순 정렬 ==> ";
15 rsort($myArray);
16 foreach($myArray as $data)
17 echo $data, " ";
18
19 echo "<br>순서를 반대로 ==> ";
20 $revArray = array_reverse($myArray);
21 foreach($revArray as $data)
22 echo $data, " ";
23 ?>
```



3-6 내장 함수

■ 실제 프로그래밍에서 자주 사용되는 함수 소개

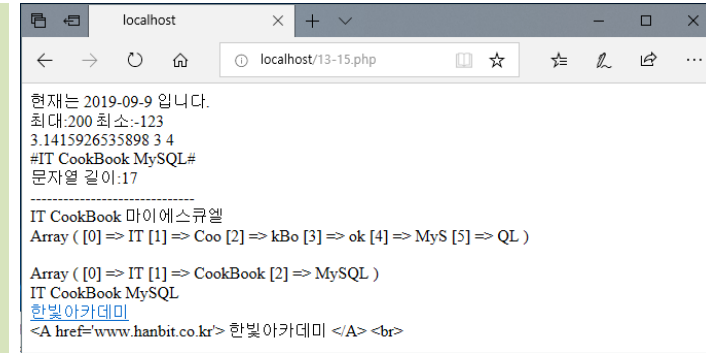
표 13-1 PHP의 주요 내장 함수

함수	사용법	설명
date()	date("Y-m-j-h")	지정한 포맷으로 날짜를 반환한다. Y는 연도, m은 월, j는 일, h는 시를 의미한다.
max(), min()	max(배열 또는 숫자), min(배열 또는 숫자)	최댓값, 최솟값을 반환한다.
pi()	pi()	파이 값을 반환한다. M_PI 상수와 동일하다.
round(), ceil()	round(숫자), ceil(숫자)	소수점 아래를 반올림하거나 올림하여 정수로 만든다.
trim()	trim(문자열)	양쪽 공백을 제거한다.
strlen()	strlen(문자열)	문자열의 길이를 구한다.
str_repeat()	str_repeat(문자열, 횟수)	문자열을 횟수만큼 반복한다.
str_replace()	str_replace(old, new, target)	target 문자열에서 old를 new로 바꾼다.
str_split()	str_split(문자열, 길이)	문자열을 길이만큼 잘라 배열로 분리한다. 길이를 생략하면 1로 간주한다.
explode()	explode(구분자, 문자열)	문자열을 구분자로 분리하여 배열로 저장한다.
implode()	implode(배열, 구분자)	배열 중간에 구분자를 넣어 하나의 문자열로 잇는다.
htmlspecialchars()	htmlspecialchars(HTML 코드)	HTML 코드를 해석하지 않고 그대로 웹 브라우저에 표시한다.

3-6 내장 함수

■ PHP 내장 함수의 사용 예

```
1 <?php
2 $today = "현재는 ".date("Y-m-j")." 입니다.";
3 echo $today, "<br>";
4
5 $ary = array(100, 50, 200, 7);
6 echo "최대:", max($ary) , " 최소:", min(-123, 50, 999), "<br>";
7
8 echo pi(), " ", round(M_PI), " ", ceil(M_PI), "<br>";
9
10 $str = " IT CookBook MySQL "; // 앞뒤에 공백 3개씩
11 $str = trim($str);
12 echo "#", $str, "#", "<br>";
13
14 echo "문자열 길이:", strlen($str), "<br>";
15
16 echo str_repeat("-", 30), "<br>";
17
18 echo str_replace( "MySQL", "마이에스큐엘", "IT CookBook MySQL"), "<br>";
19
20 $ary = str_split("IT CookBook MySQL", 3);
21 print_r($ary); echo "<br>"; // 배열 출력
22 echo "<br>";
23
24 $ary = explode(" ", "IT CookBook MySQL");
25 print_r($ary); echo "<br>"; // 배열 출력
26
27 echo implode($ary, " "), "<br>";
28
29 $myHTML = "<A href='www.hanbit.co.kr'> 한빛아카데미 </A> <br>";
30 echo $myHTML;
31 echo htmlspecialchars($myHTML);
32 ?>
```



3-6 내장 함수

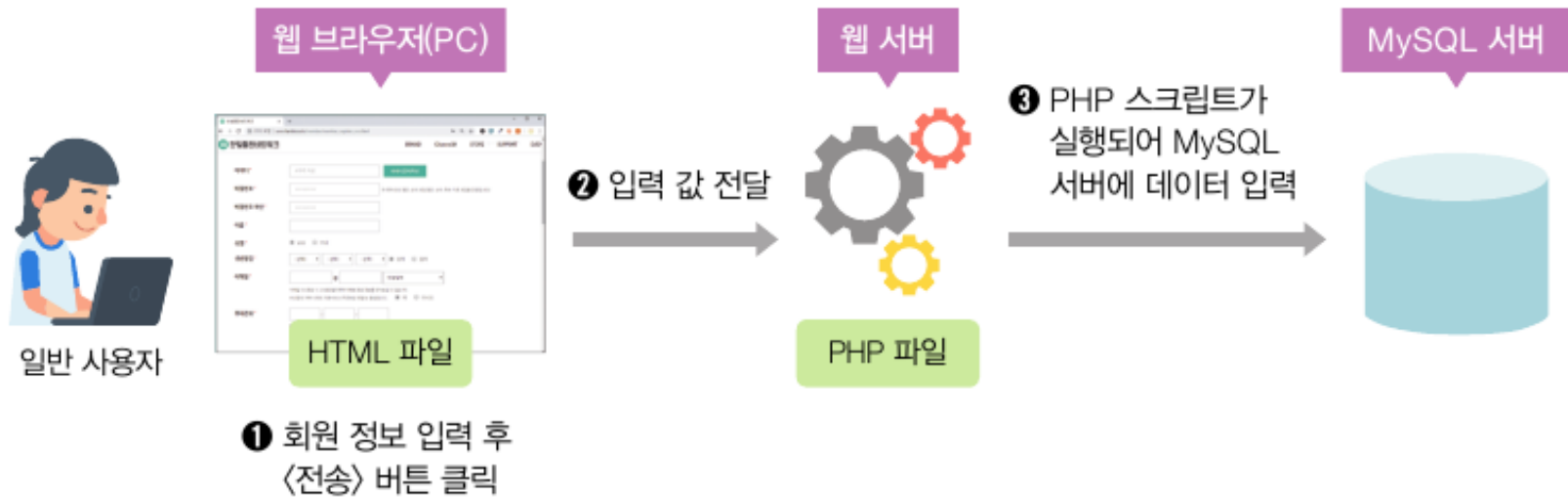
- MySQL과 관련된 내장 함수

표 13-2 PHP에서 제공하는 MySQL 주요 관련 함수

함수	설명
mysqli_connect()	MySQL 서버에 연결한다.
mysqli_connect_error()	MySQL 서버에 연결 오류가 발생하면 그 원인을 알려준다.
mysqli_close()	MySQL 서버에 연결된 것을 종료한다.
mysqli_select_db()	사용할 데이터베이스를 지정한다.
mysqli_query()	SQL 문을 서버에서 실행한다.
mysqli_error()	SQL 문이 서버에서 실패한 경우 그 원인을 알려준다.
mysqli_num_rows()	SELECT 문의 실행 결과 몇 개의 행이 출력되었는지 알려준다.
mysqli_fetch_array()	SELECT 문의 실행 결과에서 결과 행을 추출한다.

4-1 HTML과 PHP 데이터 전송

- 웹 브라우저에서 MySQL 서버에 데이터를 입력하는 과정



4-1 HTML과 PHP 데이터 전송

- 사용자의 아이디와 이름을 입력한 후 PHP 파일을 전달하는 코드

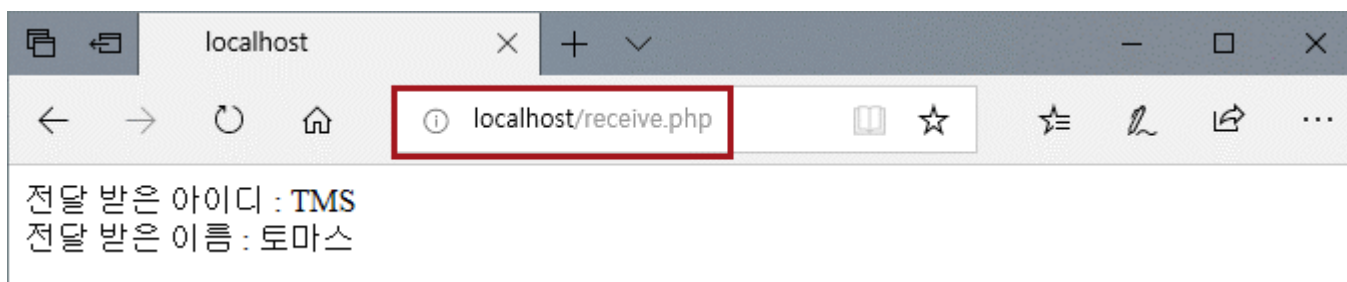
```
1 <HTML>
2 <HEAD>
3   <META http-equiv="content-type" content="text/html; charset=utf-8">
4 </HEAD>
5 <BODY>
6
7   <FORM METHOD="post" ACTION="receive.php">
8     아이디 : <INPUT TYPE="text" NAME="userID">
9     이름 : <INPUT TYPE="text" NAME="userName">
10    <BR><BR>
11    <INPUT TYPE="submit" VALUE="전송">
12  </FORM>
13
14 </BODY>
15 </HTML>
```

The screenshot shows a web browser window with the address bar displaying 'localhost/send.html'. The page content includes a form with two text input fields: '아이디 : TMS' and '이름 : 토마스'. Below these fields is a submit button labeled '전송'. A red dotted arrow originates from the submit button and points back to the input fields, indicating the flow of data submission.

4-1 HTML과 PHP 데이터 전송

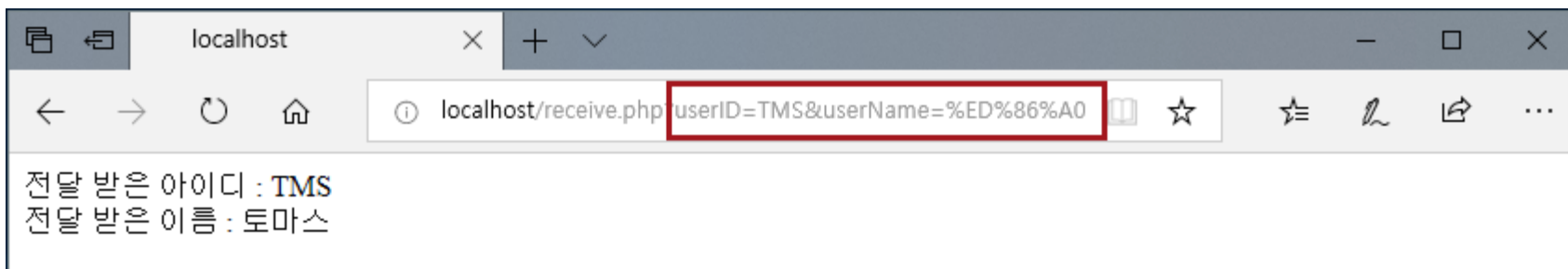
- send.html 파일의 내용을 받은 receive.php 파일의 코드

```
1 <?php
2 $userID = $_POST["userID"];
3 $userName = $_POST["userName"];
4
5 echo "전달 받은 아이디 : ", $userID, "<br>";
6 echo "전달 받은 이름 : ", $userName, "<br>";
7 ?>
```



- GET 방식으로 데이터를 주고받으려면

- [소스코드 13-16]에서 7행의 post를 get으로 바꾸고, [소스코드 13-17]에서 2, 3행의 \$_POST를 \$_GET으로 바꾸면 됨



4-1 HTML과 PHP 데이터 전송

- POST 방식과 GET 방식의 차이점
 - POST 방식은 주소 창에 'localhost/receive.php'만 있음
 - GET 방식은 주소 뒤에 '?변수=값&변수=값'이 있음
 - GET 방식은 전달되는 데이터가 주소 창에 노출되기 때문에 주의
 - 비밀번호와 같이 중요한 정보를 전달하는 경우에는 POST 방식을 사용하는 것이 바람직

4-2 HTML과 PHP 프로그래밍

- HTML 문법으로만 작성된 파일은 * .php로 저장해서 사용해도 상관없음

```
1 <HTML>
2 <HEAD>
3   <META http-equiv="content-type" content="text/html; charset=utf-8">
4 </HEAD>
5 <BODY>
6
7   <H1>이 파일은 * .PHP 입니다.</H1>
8
9 </BODY>
10 </HTML>
```

4-2 HTML과 PHP 프로그래밍

- PHP와 HTML 코드를 섞어서 사용할 수도 있음

```
1 <?php
2   $num1 = 100;
3   $num2 = 200;
4   $sum = $num1 + $num2;
5 ?>
6
7 <HTML>
8 <HEAD>
9   <META http-equiv="content-type" content="text/html; charset=utf-8">
10 </HEAD>
11 <BODY>
12
13   <H1>계산 결과는 <?php echo $sum ?> 입니다.</H1>
14
15 </BODY>
16 </HTML>
```