



영진전문대학교

글로벌시스템융합과 - GSC

PHP – Tag, Comments and Variable

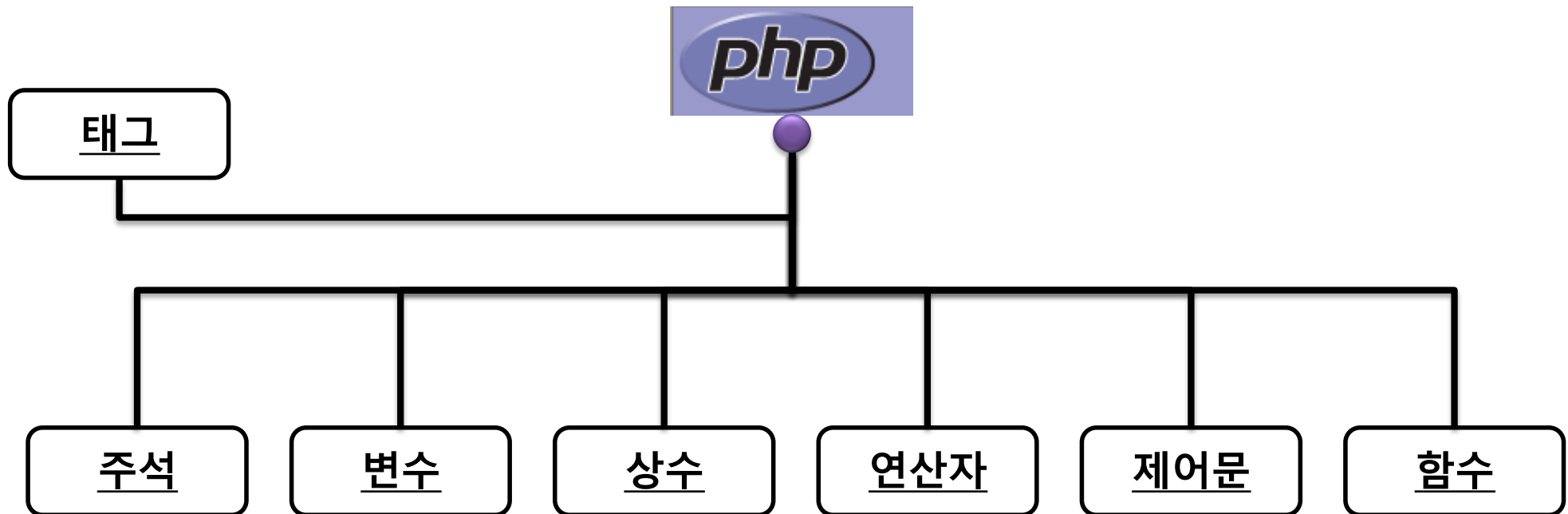
정영철 교수

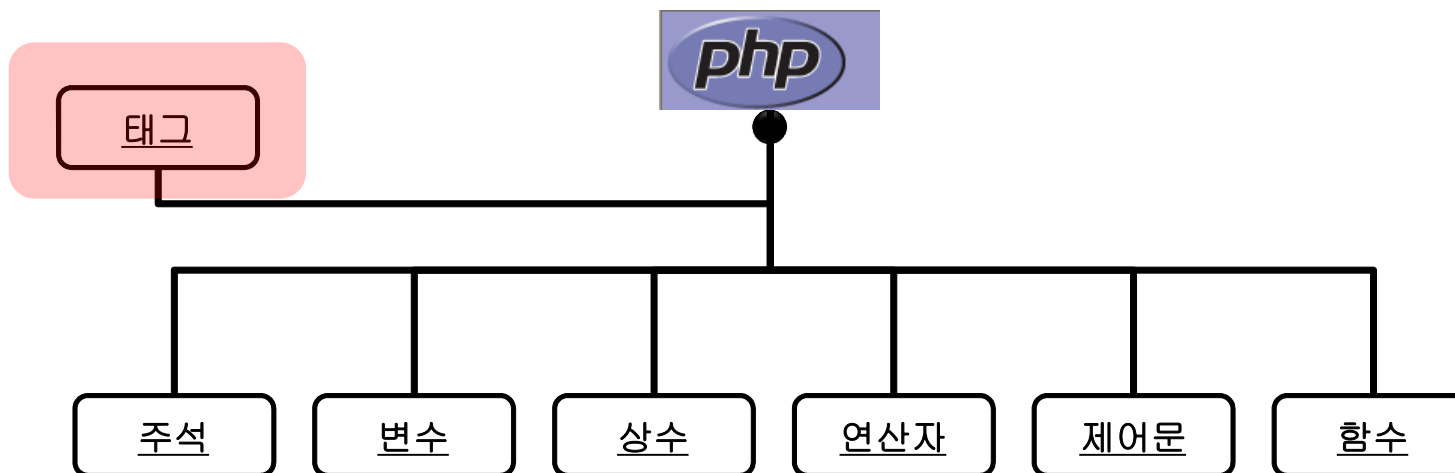
글로벌시스템융합과



A.I WITH BETTER LIFE

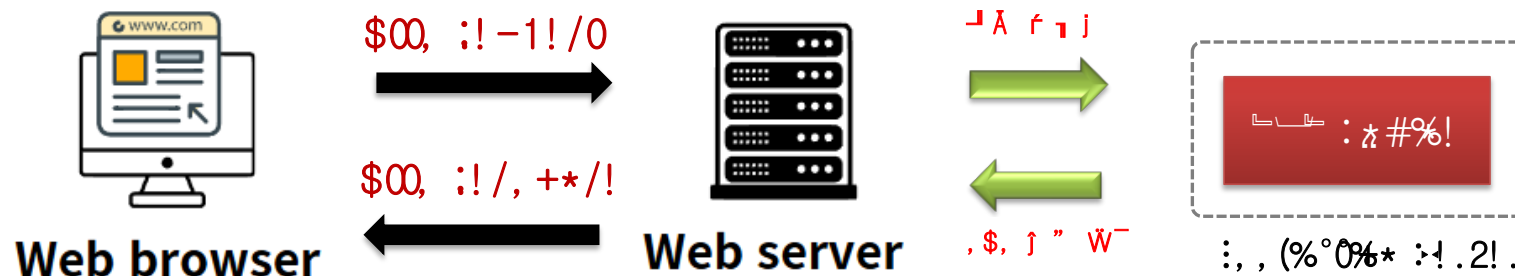
PHP 프로그래밍 언어의 구성요소 : 구조적 언어 관점





What is PHP and TAG?

- PHP의 개발 목적
 - 동적인 Web application 개발
 - HTML 문서에 삽입 가능한 임베디드 언어로 설계됨
 - 사용이 쉽고 배포가 간편하여 웹 프로그래밍에 널리 사용됨
- PHP 코드 작성 형태
 - 단독 .php 파일로 작성 가능
 - HTML 문서 내에 PHP 코드 삽입 가능
 - HTML과 PHP 코드를 구분하기 위한 구분자(Tag) 필요
- PHP 태그(Tag)의 개념
 - PHP 코드를 시작하고 종료하기 위한 구문적 표시P
 - **PHP 엔진은 태그로 감싼 코드만 실행**



PHP Tag 종류

순 번	시작태그	끝 태그	구분	비고
1	<? php	?>	일반적 방법	권장
2	<?	?>	Short tag	옵션 활성화 시 사용 가능
3	<? =	?>	echo 출력용	

Changelog

Version	Description
7.0.0	The ASP tags <%, %>, <%=, and the script tag <script language="php"> are removed from PHP.
5.4.0	The tag <? = is always available regardless of the short_open_tag ini setting.

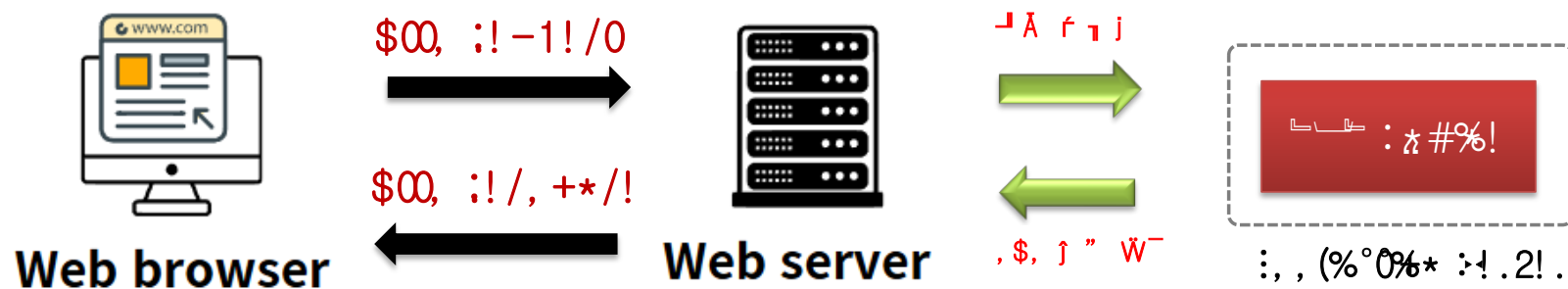
PHP Tag 예제

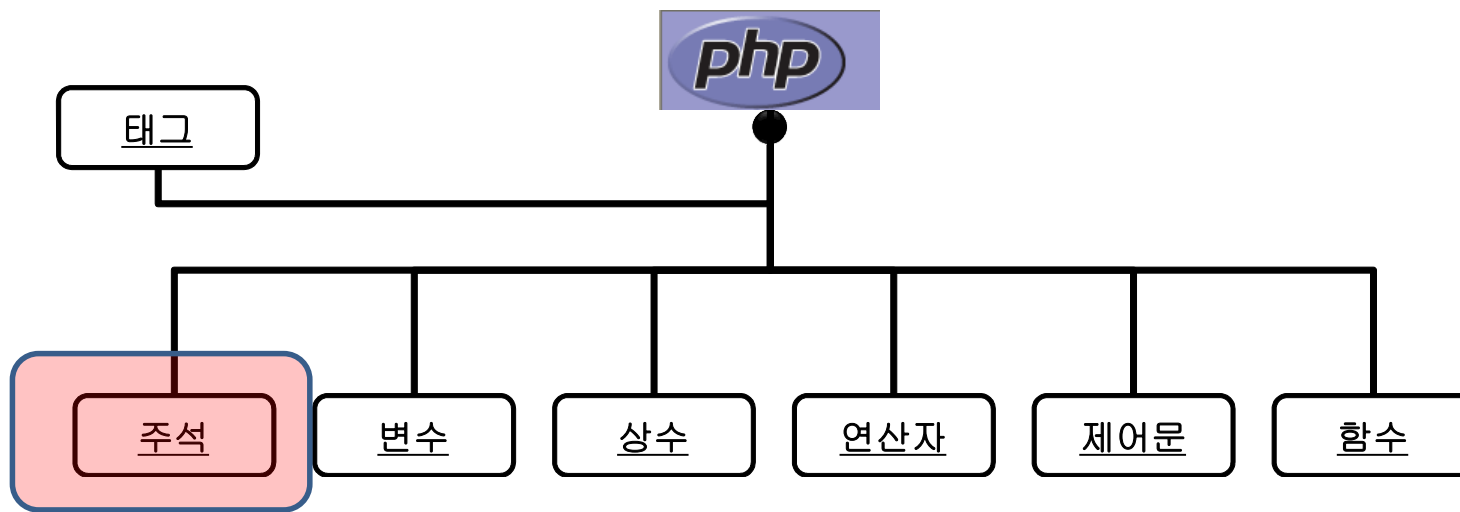
```
9  <?php // <-- Full tag : 권장사항
10     echo "hello world 1";
11  ?>
12  이 문장은 Skip 됩니다!!
13  왜냐면 PHP 엔진은 전체 소스코드 해석 시 PHP Tag가 선언된
14  문장들만 선택하기 때문입니다.
15  <? // <-- Short tag : PHP 환경 설정파일에서 On되어 있어야 사용 가능
16      //          권장 X
17      echo "hello world 2";
18
19      $outputType = "hello world 3" // <- 변수 output
20  ?>
21  <?= // <- Echo tag : <?php echo와 같다
22  $outputType;
23  ?>
```

hello world 1이 문장은 Skip 됩니다!! 왜냐면 PHP 엔진은 전체 소스코드 해석 시 PHP Tag가 선언된 문장들만 선택하기 때문입니다. hello world 2hello world 3

PHP Tag 고찰

- PHP에서 Tag를 왜 사용해야 되는가?
- HTML 문서 내 PHP 문장(Sentence)이 위치해야 될 곳은?



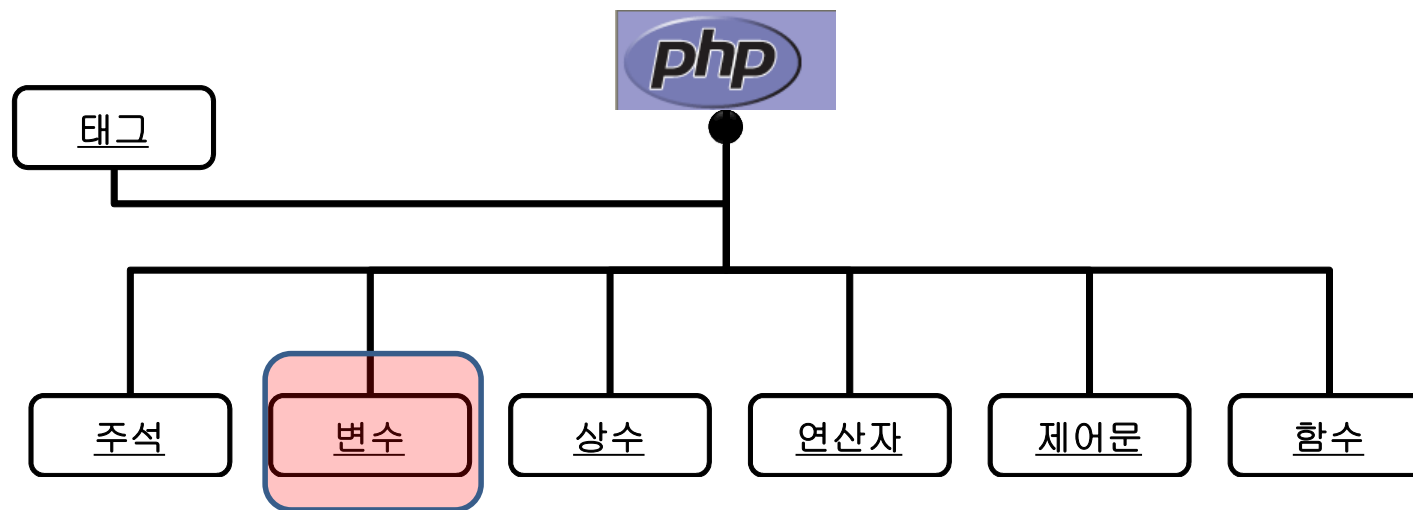


PHP에서의 주석표기 방법

- C, JAVA 언어와 동일, 단 “#” 추가!!

주석 방법	설명
//	• 한 줄 주석 처리
#	• 한 줄 주석 처리
/* */	• 여러 줄 주석 처리

```
8 <?php
9 // 싱글라인 주석 달기 - 1
10 # 싱글라인 주석 달기 - 2
11 /*
12 | 멀티라인 주석
13 |
14 */
15 ?>
```



변수 (Variable)

- Programming 언어에서 변수 학습 시 고려 사항
 - ✓ 변수 선언 방법 (Declaration)
 - ✓ 변수 명명 법 (Naming)
 - ✓ 변수 자료형 (Type)
 - ✓ 변수 범위 (Access Scope)
 - ✓ 변수 생명주기 (Lifecycle)
 - ✓ 정적 변수 (Static variable)

변수 선언 (Variable declaration) 방법

- \$ 를 이용해서 변수 선언!!

\$변수명 = 초기값;

```
<?
  $first_variable = "첫 번째 변수<BR>";
  echo $first_variable;
  .....
  $second_variable = 10;
  echo "$second_variable<br>";
  .....
  $third_variable = 0.5;
  echo "$third_variable<br>";
  .....
  $sum = $second_variable + $third_variable;
  echo "$second_variable + $third_variable = $sum";
?>
```

변수 선언 시 고려 사항

```
3  echo $value; // Warning message, undefined variable
4  // PHP에서 변수 define 의미
5  // --> 변수가 선언있고 변수 값이 null 이 아닐때 (AND 조건)
6  // --> 그 이외에는 undefine 처리
7
8  $value; // value 변수 선언, $value 값 null
9          // 변수 선언 시 초기 값이 없으면 null 값 할당
10         // 변수 선언 시에는 꼭 초기값 설정 할 것
11
12  echo isset($value) ? "true - 1" : "false - 1"; // false - 1, 변수 선언 : true and 변수 값 : null
13
14  $value = 2;
15
16  echo isset($value) ? "true - 2" : "false - 2"; // true - 2, 변수 선언 : true and 변수 값 : 2
17
18  unset($value); // 변수 값 메모리 상에서 해제
19
20  echo isset($value) ? "true - 3" : "false - 3"; // false - 3, 변수 선언 : false
```

★ 변수 관련 함수는 아래 사이트 참조

<https://www.php.net/manual/en/ref.var.php>

Dynamic type-casting (Type Juggling)

```
<?
[ $variable_status = 10; ]
$variable_status = gettype($variable_status);
echo "$variable_status<BR>";

[ $variable_status = "문자열"; ]
$variable_status = gettype($variable_status);
echo "$variable_status<BR>";

[ $variable_status = 10.5; ]
$variable_status = gettype($variable_status);
echo "$variable_status<BR>";

[ $variable_status = true; ]
$variable_status = gettype($variable_status);
echo "$variable_status<BR>";

[ unset($variable_status); ]
echo isset($variable_status) ? "Something is in" : "NULL";
?>
```

integer

string

float

boolean

NULL

식별자 규칙

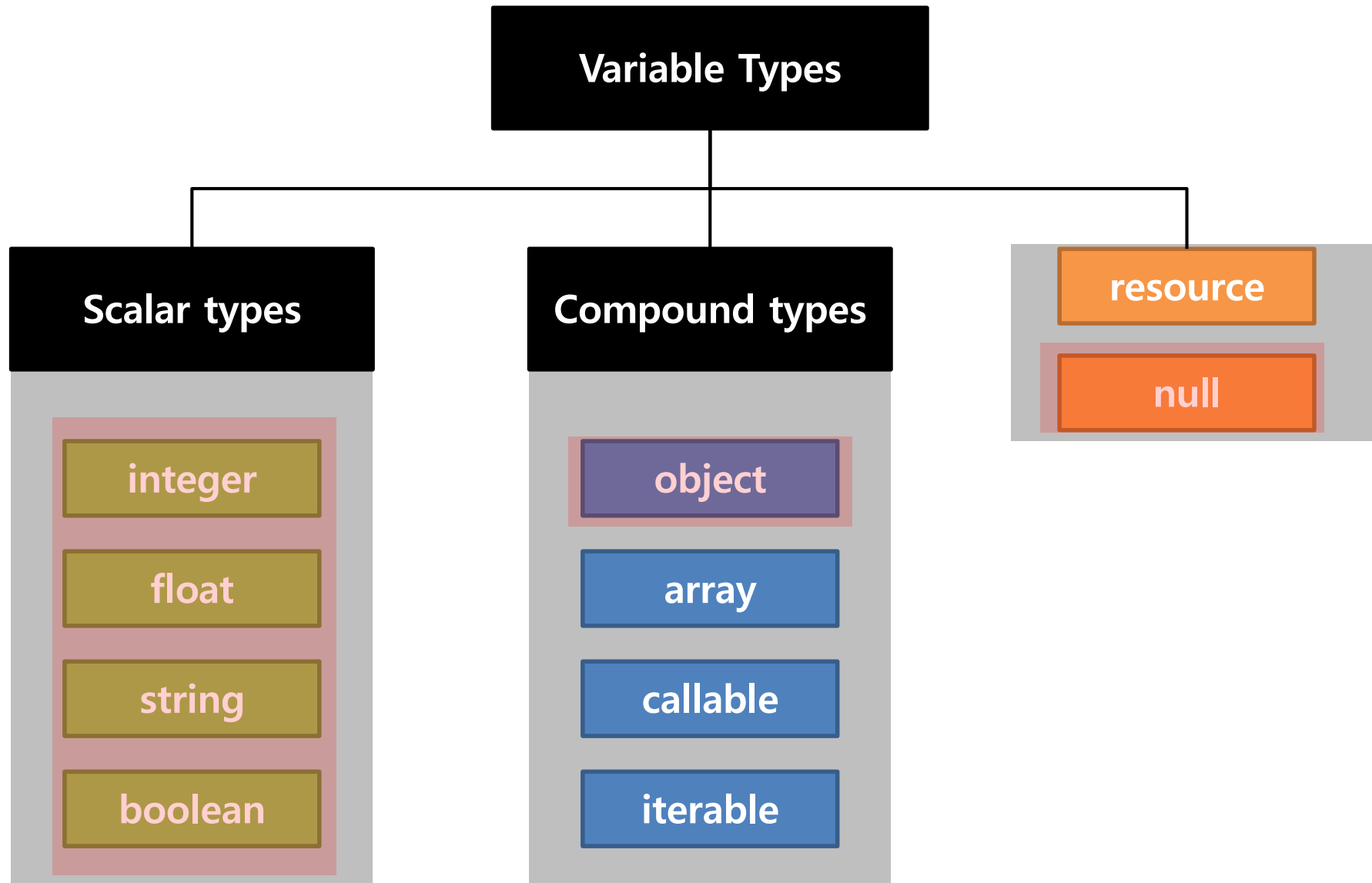
- PHP에서 식별자 규칙(**C언어와 동일**)
 - 사용가능 문자
 - A-Z, a-z, 0-9, _
 - 첫 번째 시작문자
 - A-Z, a-z, _
 - 대소문자 구분

<?

```
$apple    = "1st variable:  apple<BR>";  
$apple#   = "2st variable:  apple<BR>";  
$apple2   = "3st variable:  apple2<BR>";  
$_test    = "4st variable:  _test<BR>";  
$2test    = "5st variable:  2test<BR>";
```

?>

PHP 변수 자료형 종류 (1)



PHP 변수 자료형 종류 (2)

```
1  <?php
2  $list = array(1, 2, 3);
3
4  // Type hint : 함수 매개변수의 자료형을 명시
5  // PHP 5.0 이후로 기능 제공
6  // 함수 매개변수와 인자 값의 자료형을 비교,
7  // 일치 하지 않을 경우 Error 발생
8  // 아래 예제에서 prtSomething(array $argList)
9  // "array"는 pseudo-type
10 function prtSomething(array $argList) {
11     for($i = 0 ; $i < count($argList) ; $i++)
12         echo $argList[$i];
13 }
14
15 prtSomething($list); // 1, 2, 3
16
17 $value = 2;
18 prtSomething($value); // Error
```

Scalar 변수 자료형 종류

- PHP에서 지원하는 Scalar 변수 자료형

구 분	크 기	비 고
integer	4bytes	
float	8bytes	IEEE 754
string	길이 제한 없음	PHP 7.0 이상 and 64bit OS
	2GB	그 외
boolean		true or false
NULL		

- PHP 문서에서는 Null은 Scalar형으로 분류되지 않지만, Scalar형으로 보는게 맞다.

자료형: integer (1)

- 크기 : 4byte!!
 - 시스템 환경에 따라 다르다 (운영체제 종류, Kernel 종류)
- 리터럴 상수 정수 표현법 제공
 - 진법 표현
 - 자릿수 표현

```
<?php
$a = 1234; // decimal number
$a = 0123; // octal number (equivalent to 83 decimal)
$a = 0x1A; // hexadecimal number (equivalent to 26 decimal)
$a = 0b11111111; // binary number (equivalent to 255 decimal)
$a = 1_234_567; // decimal number (as of PHP 7.4.0)
?>
```

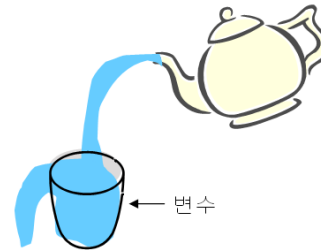
자료형: integer (2)

- What is overflow?

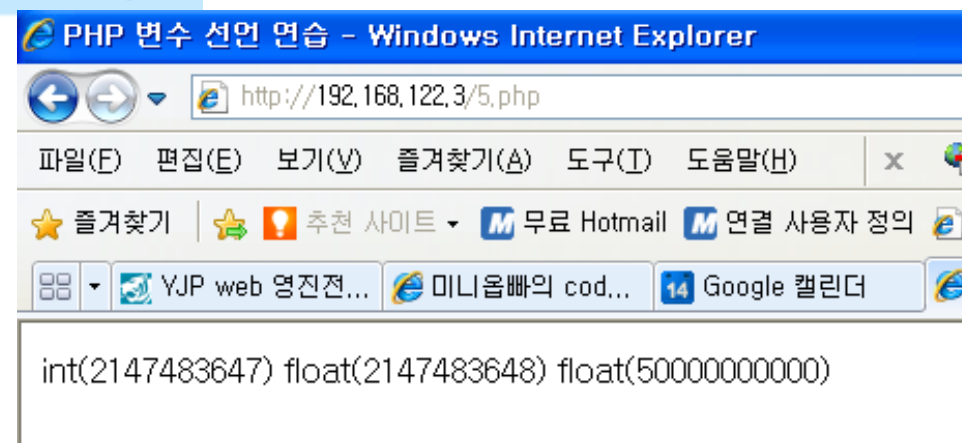
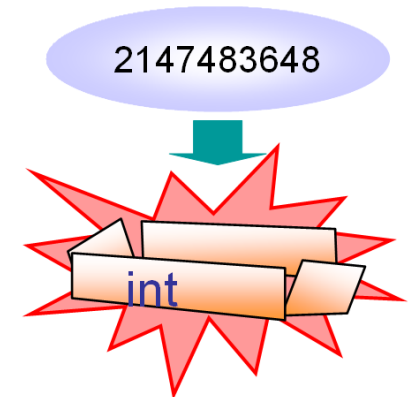
```
<?php
$large_number = 2147483647;
var_dump($large_number);

$large_number = 2147483648;
var_dump($large_number);

$million = 1000000;
$large_number = 50000 * $million;
var_dump($large_number);
?>
```

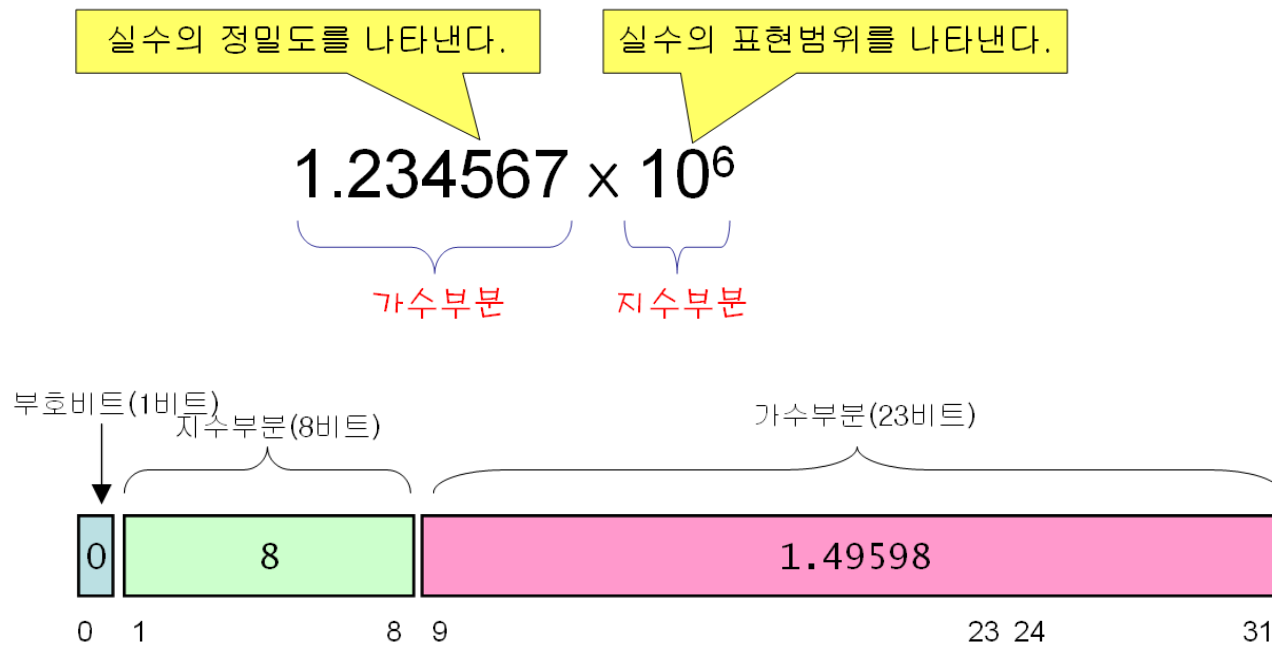


overflow



자료형: float (1)

- 크기 : 8byte
 - IEEE 754 형식을 따른다.

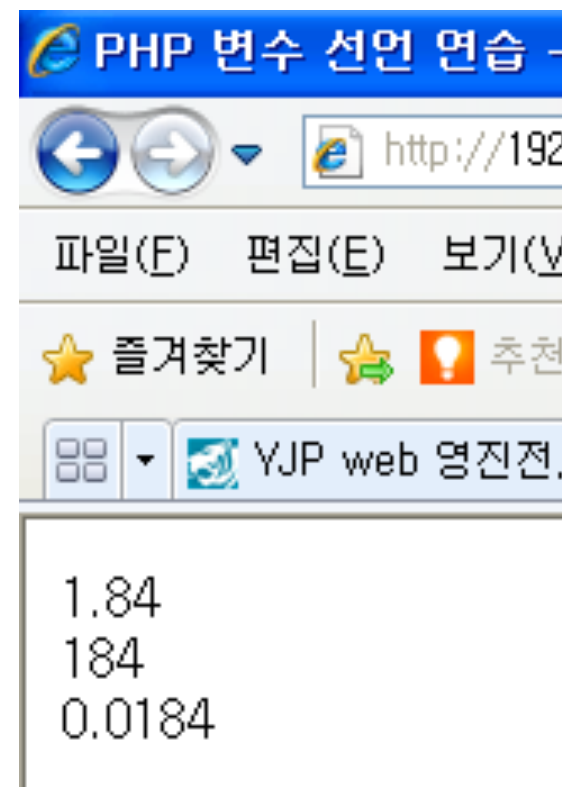


자료형: float (2)

```
<?php
$float_value_expression = 1.84;
echo $float_value_expression."<br>";

$float_value_expression = 1.84e2;
echo $float_value_expression."<br>";

$float_value_expression = 1.84e-2;
echo $float_value_expression."<br>";
?>
```



자료형: string - Single and Double quoted

- 문자열 저장
- 최대 크기
 - PHP 7.0 이상, 64bit OS 크기 제한 없음
 - 그 외 2GB 크기 제한
- 문자열 리터럴 상수 종류
 - Single quoted
 - **Double quoted**
 - Heredoc syntax
 - Nowdoc syntax

```
1  <?php
2  $name = "Youngchul Jung";
3
4  // Single quoted
5  echo '이름 : $name ₩" ₩" ';
6  echo "<br>";
7  // Double quoted
8  echo "이름 : $name ₩" ₩" ";
9  ?>
```

• 출력결과

```
이름 : $name ₩" ₩"
이름 : Youngchul Jung " "
```

자료형: string – Heredoc quoted

```
1 <?php
2 $str = <<<MYCOMMENT
3     Example of string
4     spanning multiple lines
5     using heredoc syntax.
6 MYCOMMENT;
7
8 echo $str."<br>";
9
10 $name = "Youngchul Jung";
11 echo <<<EOT
12     My name is "$name".
13     Nice to meet you!!
14 EOT;
15 ?>
```

- 식별자를 이용해 문자열 지정
- 식별자 명명법
 - 영문자, 숫자, _ 만 가능
 - 첫 글자 영문자만 가능
- 시작 식별자
 - 식별자 선언 후 다음 글자는 반드시 개행 문자(Enter)
- 종료 식별자
 - 라인 첫 글자로 위치
 - ; 후 반드시 개행 문자(Enter)

출력결과

Example of string spanning multiple lines using heredoc syntax.
My name is "Youngchul Jung". Nice to meet you!!

자료형: string – Nowdoc quoted

```
1 <?php
2 $str = <<< 'MYCOMMENT'
3 Example of string
4 spanning multiple lines
5 using heredoc syntax.
6 MYCOMMENT;
7
8 echo $str."<br>";
9
10 $name = "Youngchul Jung";
11 echo <<<'EOT'
12 My name is "$name".
13 Nice to meet you!!
14 EOT;
15 ?>
```

- Heredoc과 동일
- 단, 문자열 내 파싱 X
 - ✓ Single quoted 처리와 같음

• 출력결과

Example of string spanning multiple lines using heredoc syntax.
My name is "\$name". Nice to meet you!!

자료형: Boolean

- Boolean형-> 참(True) or 거짓(False)

```
<?
$a = true;

echo '$a is ' . ($a ? "True" : "False") . "<br>";

$b = false;

echo '$b is ' . ($b ? "True" : "False");
?>
```

\$a is True
\$b is False

Falsy

- False는 아니지만 false 값으로 간주

When converting to boolean, the following values are considered **FALSE**:

- the boolean **FALSE** itself
- the integers 0 and -0 (zero)
- the floats 0.0 and -0.0 (zero)
- the empty string, and the string "0"
- an array with zero elements
- the special type NULL (including unset variables)
- SimpleXML objects created from empty tags

변수 범위 (Variable Scope)



- 전역변수 (Global variable)
 - PHP 소스 파일 전체 범위에서 사용되는 변수
 - Hoisting 지원하지 않음. 즉 변수가 선언된 시점 이후로 접근 가능
- 지역변수 (Local variable)
 - **함수** 내 정의 된 변수
 - 함수 내에서만 접근 가능
 - Hoisting 지원하지 않음. 즉 변수가 선언된 시점 이후로 접근 가능
- 슈퍼전역변수 (Super Global variable)
 - PHP 엔진에서 제공되는 변수
 - 전역 변수 범위

전역 변수 (1)

```
1 <?php
2 // global variable, 함수 또는 클래스, 인터페이스 밖에서 선언 될 경우
3 $value_1 = 1; // 전역 변수 선언, 접근 가능 범위 : 4번 라인 부터 ~ 파일 끝까지
4 // 변수 Hoisting 기능 지원 하지 않음
5
6 { $value_2 = 2; } // { } block-level scope 지원 하지 않음
7
8 // 흐름제어문 내 선언 변수 -> 전역 변수
9 for($i = 0 ; $i < 2 ; $i++) {
10     echo $i;
11 }
12 echo $i;
13 /* <-- Java code
14 for(int i = 0 ; i < 2 ; i++) {
15     System.out.println(i);
16 }
17 System.out.println(i); // Error
18 -->> */
19 <?>
20 <?php
21 echo $value_2; // Error, { } block-level scope 지원 하지 않음
22 <?>
```

전역 변수 (2)

```
1 <?php
2 $value = "전역 변수: value";
3
4 function foo() {
5     // 함수 내에서 선언된 변수는 지역변수
6     // 함수에서 전역 변수명을 이용해 전역변수 참조 불가
7     echo $value; // Error, 지역변수 $value를 찾고 있으나, 선언 되지 않았음
8
9     // 이를 해결하기 위해 두 가지 방법 제공
10
11    // 1. global 키워드 이용
12    global $value; // 전역 변수 value, 현 함수내 사용 선언
13    echo $value; // 결과 값 : "전역 변수 : value"
14
15    // 2. $GLOBALS 슈퍼 전역 함수 이용, 추천 방법
16    echo $GLOBALS["value"]; // 결과 값 : "전역 변수 : value"
17 }
18 foo();
19 ?>
```

지역 변수

```
2 // 함수 레벨 Scope 지원
3 function foo() {
4     // 지역변수 : 함수 내 선언된 변수
5     // 즉 함수 내 변수 참조 시 함수 내 선언된 변수만 참조
6     // 지역변수 접근 범위 - 시작 : 변수 선언, 종료 : 함수 종료
7     $value_1 = "지역변수1";
8     {
9         $value_2 = "지역변수2";
10    }
11
12    echo $value_1; // 결과 값 : "지역변수1"
13    echo $value_2; // 결과 값 : "지역변수2" <- 블록레벨 Scope 지원 X
14 }
15
16 foo();
```

Super Global Variable (1)

- PHP 엔진에서 제공하는 변수
- 모든 PHP파일에서 접근 가능

- [\\$GLOBALS](#)
- [\\$_SERVER](#)
- [\\$_GET](#)
- [\\$_POST](#)
- [\\$_FILES](#)
- [\\$_COOKIE](#)
- [\\$_SESSION](#)
- [\\$_REQUEST](#)
- [\\$_ENV](#)

- 전역변수 접근
- 웹 서버 정보 관리
- HTTP Req – GET 정보 획득
- HTTP Req – POST 정보 획득
- HTTP Req – 파일 정보 획득
- 쿠키 관련 정보 관리 (예: 로그인)
- 세션 관련 정보 관리 (예: 로그인)
- HTTP Req 정보 관리
- 시스템 환경 관련 정보 획득 (예: 서버 이름)

Super Global Variable (2)

- HTML Form을 이용한 데이터 전달

main.html 파일

localhost/test/main.html

이름과 나이를 입력하세요

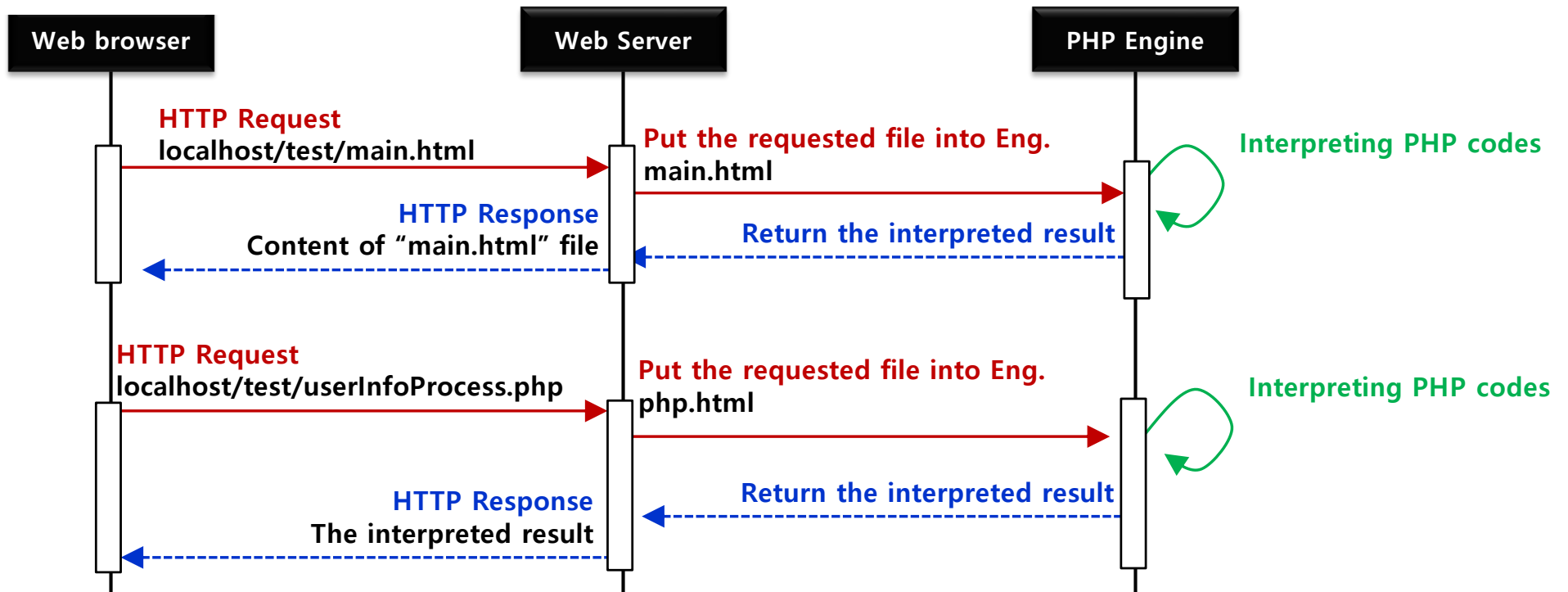
이름 :

나이 :

userInfoProcess.php 파일

localhost/test/userInfoProcess.php

정영철님 환영합니다.저도 20살 입니다.



Super Global Variable (3)

- HTML Form을 이용한 데이터 전달

▪ `) %$@ ({ } ,`

localhost/test/main.html

이름과 나이를 입력하세요

이름 : 정영철

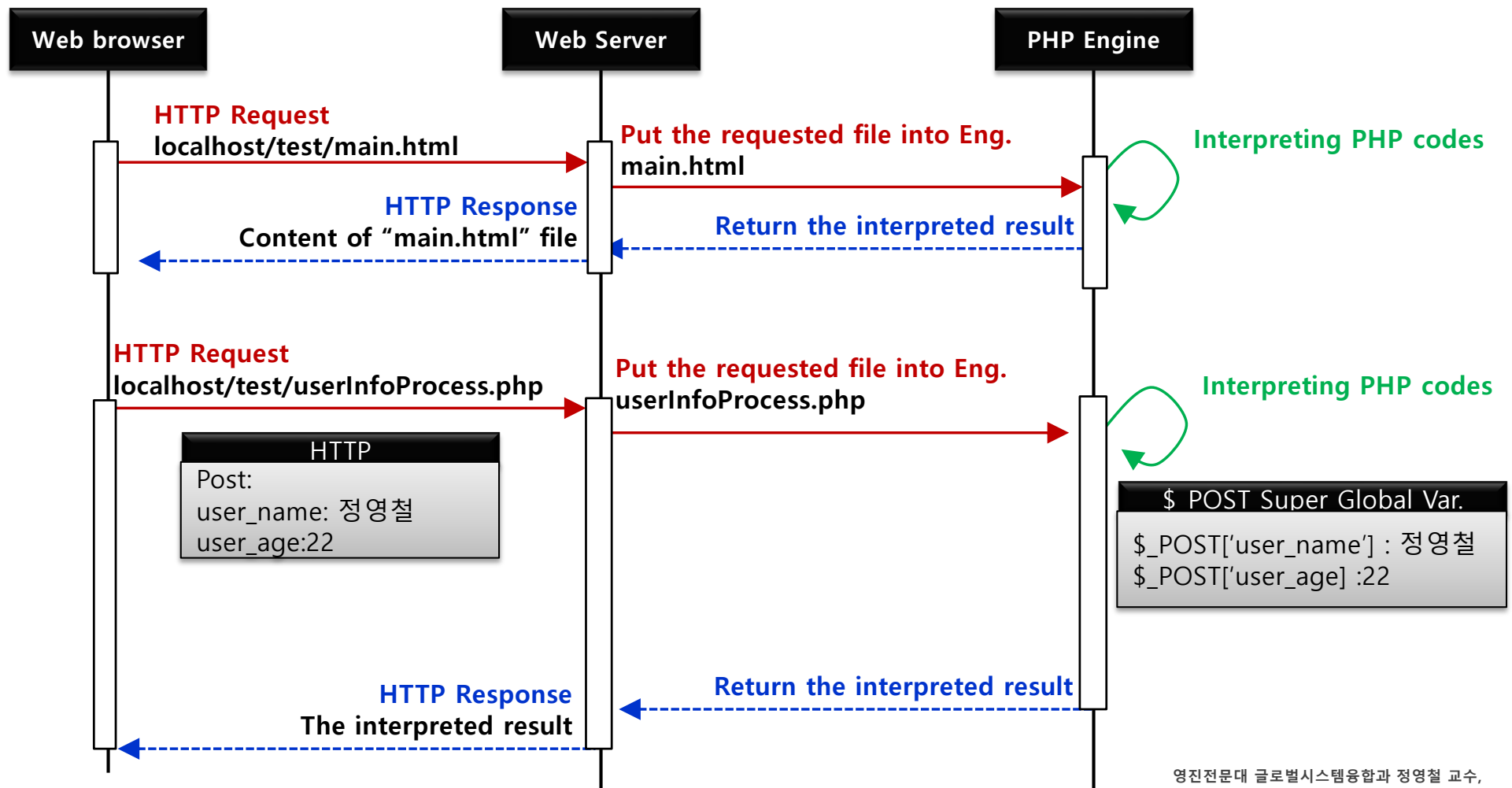
나이 : 20

제출하기

▪ `1/!~*"+^.= ?//\; $, { }`

localhost/test/userInfoProcess.php

정영철님 환영합니다.저도 20살 입니다.



Super Global Variable (4)

■) %\$0) (f 1

localhost/test/main.html

이름과 나이를 입력하세요

이름 : 정영철

나이 : 20

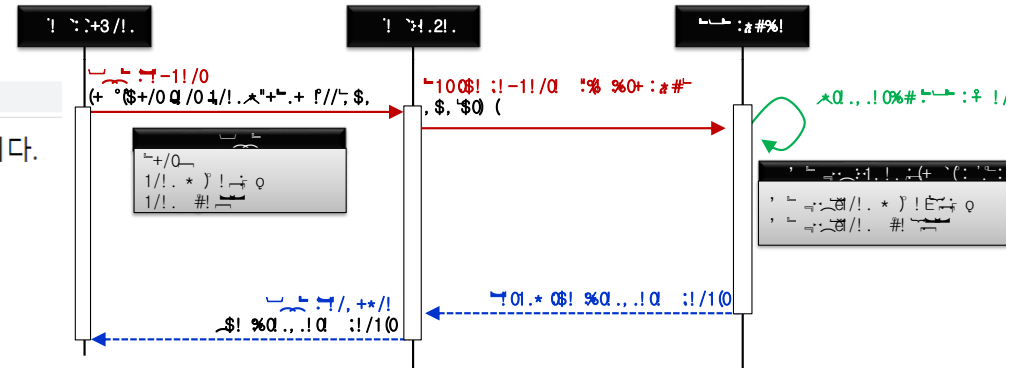
제출하기



■ 1/!..*"+~.+ f//~\$, f 1

localhost/test/userInfoProcess.php

정영철님 환영합니다.저도 20살 입니다.



main.html 파일

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6  </head>
7  <body>
8      <form action="userInfoProcess.php" method="post">
9          <fieldset style="width:300px">
10             <legend> 이름과 나이를 입력하세요</legend>
11             이름 : <input type="text" name="user_name"> <br>
12             나이 : <input type="text" name="user_age">
13             <input type="submit" value="제출하기">
14          </fieldset>
15      </form>
16  </body>
17 </html>
    
```

userInfoProcess.php 파일

```

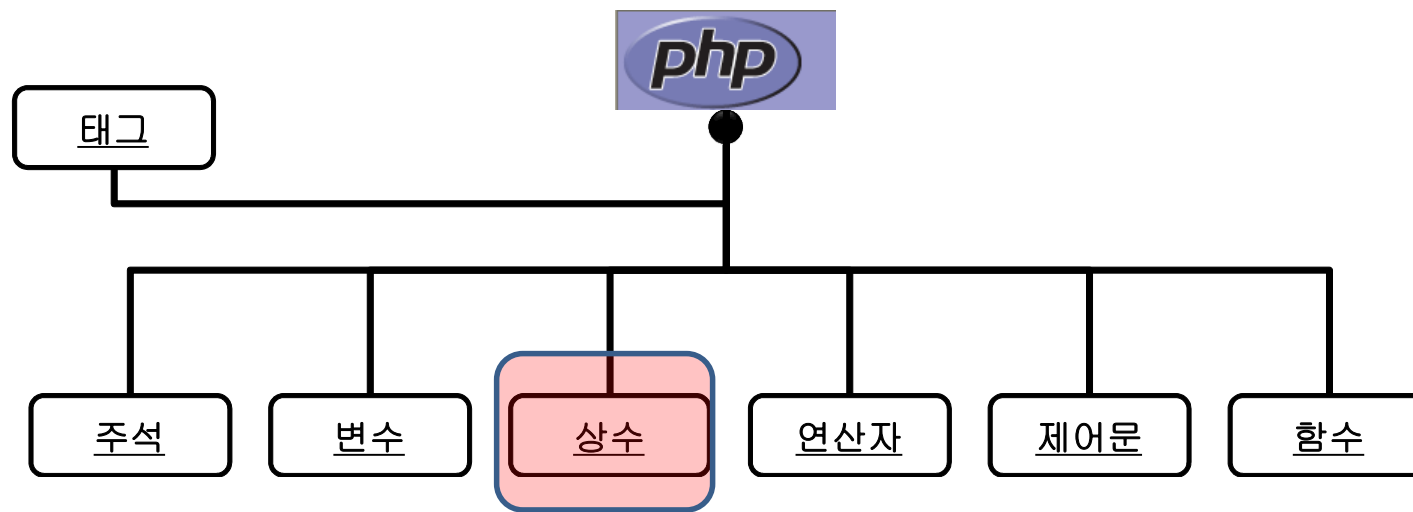
1  <?php
2  if( isset($_POST['user_name']) && isset($_POST['user_name']) ) {
3      echo "{$_POST['user_name']}님 환영합니다.";
4      echo "저도 {$_POST['user_age']}살 입니다.";
5  } else {
6      echo "잘못된 접근!!";
7  }
    
```

정적(Static) 변수

```
3  function test_1 () {
4      $value = 0;
5      for($i = 0 ; $i < 3 ; $i++)
6          $value++;
7      echo $value;
8  }
9  test_1(); // 결과 값 : 3
10 test_1(); // 결과 값 : 3
11
12 function test_2 () {
13     static $value = 0; // static 변수, 함수 내에서만 사용 가능
14                        // 함수가 종료 되어도 값은 유지
15     for($i = 0 ; $i < 3 ; $i++)
16         $value++;
17     echo $value;
18 }
19 test_2(); // 결과 값 : 3
20 test_2(); // 결과 값 : 6
21 test_2(); // 결과 값 : 9
```

Variable variables

```
1 <?php
2
3 $varName = "myName";
4
5 // 변수를 이용해 변수명을 접근 할 수 있다.
6 $$varName = "Youngchul Jung"; // $myName = "Youngchul Jung";
7
8 echo $myName;
9
10 // 결과 "Youngchul Jung"
```



상수형

- 상수란
 - 변하지 않는 값 (프로그램 시작~종료)
- PHP에서는 define 함수와 const 키워드를 이용해서 정의
- **define("상수명", "상수에 저장 될 값");**

```
1  <?
2  define("SITE_NAME", "My Website");
3  echo SITE_NAME;  // 출력: My Website
4
5  define("CONFIG", ["host" => "localhost", "port" => 3306]);
6  echo CONFIG["host"];  // 출력: localhost
7
8
9  const PI = 3.1415;
10 echo PI;  // 출력: 3.1415
11 ?>
```

Extra Slides

Q/A

감사합니다



주문식교육의 산실
영진전문대학교