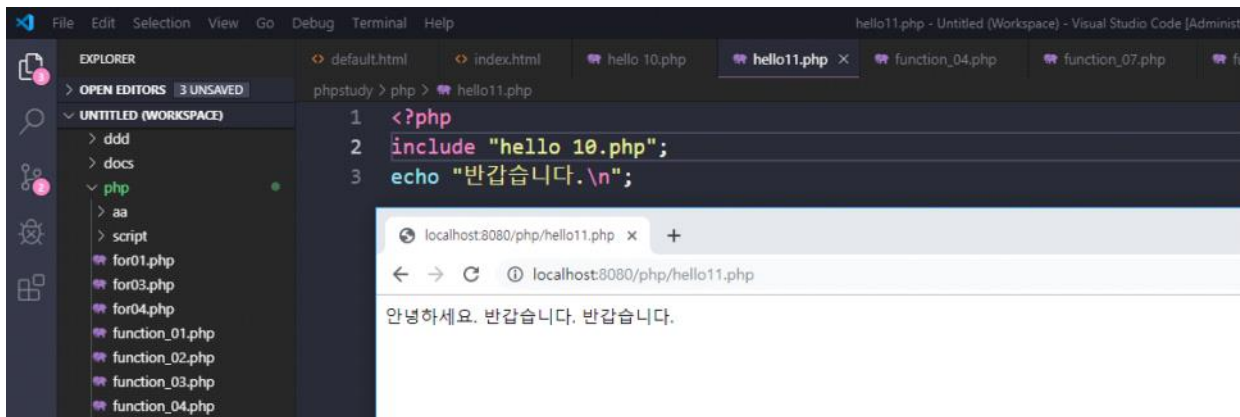


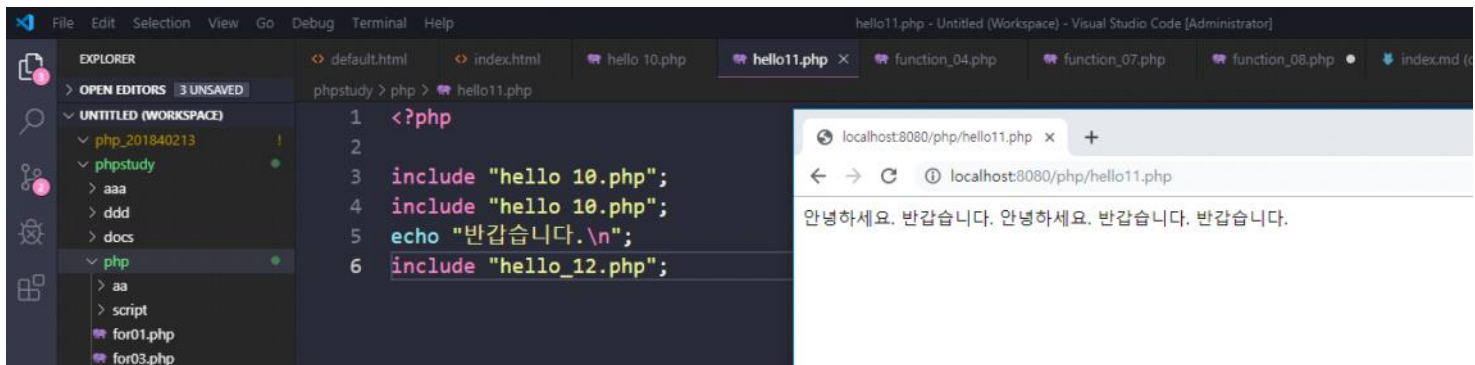
## 5일차

2019년 9월 25일 수요일 오전 10:20



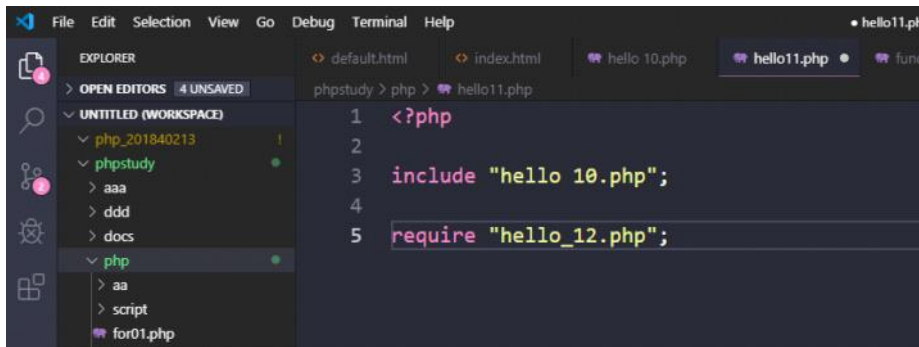
```
user@MOBILEJ0509-27 MINGW64 /d/phpstudy (master)
$ php -S localhost:8080 -t ./php
```

Php를 php폴더로 옮겨가서 실행 하는것이다. 지금의 경로는 phpstudy이기 때문에 여기서 실행하면 phpstudy에서 실행한것이 되버린다.



```
[Wed Sep 25 10:29:20 2019] PHP Warning: include(): Failed opening 'hello_12.php' for inclusion (include_path='.;C:\php\pear') in D:\phpstudy\php\hello11.php on line 6
```

없는 파일을 지정해주면 이렇게 나온다.



위의 include는 파일이 없어도 위예꺼 까지는 실행을 시켜주는데

Require를 사용하게 되면 실행이 되지 않는다

```
PHP Warning: require(hello_12.php): failed to open stream: No such file or directory in D:\phpstudy\php\hello11.php on line 5
PHP Fatal error: require(): Failed opening required 'hello_12.php' (include_path='.;C:\php\pear') in D:\phpstudy\php\hello11.php on line 5
```

현재 경로에 파일이 없어서 나오지 않는 것이다.

```
require "../hello/hello_12.php";
```

그래서 이렇게 경로를 넣어줘야 한다.

```
user@MOBILEJ0509-27 MINGW64 /d/phpstudy/php (master)
$ php hello11.php
안녕하세요.
반갑습니다.
대림입니다.대림입니다.
```

이렇게 나오게 된다.

만약에 .을 넣지 않고 그냥 한다면 루트 폴더에서 시작한다.

// 이거 시험문제 내신다고 함

```
default.html x index.html hello 10.php hello11.php function_10.php x
phpstudy > php > function_10.php
1 <?php
2
3
4
5 function daelim (){
6     echo "대림대학교";
7 }
```

```
user@MOBILEJ0509-27 MINGW64 /d/phpstudy/php (master)
$ php function_10.php
```

함수를 선언만 해주고 호출을 안했기 때문에 안나오는 것이 맞다.

```
default.html x index.html hello 10.php hello11.php
phpstudy > php > function_10.php
1 <?php
2
3
4
5 function daelim (){
6     return "대림대학교";
7 }
8 $name = daelim();
```

반환값을 받으려면 다른곳으로 저장해준뒤 echo를 해주면 된다.

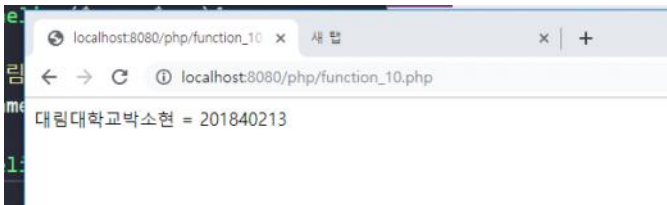
그리고

echo \$name; 해주면 된다.

```
user@MOBILEJ0509-27 MINGW64 /d/phpstudy/php (master)
$ php function_10.php
대림대학교
```

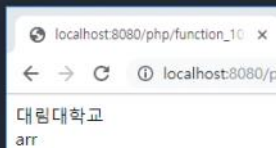
```
default.html x index.html hello 10.php hello11.php function_10.php x hello_12.php
phpstudy > php > function_10.php
1 <?php
2
3
4
5 function daelim ($name,$num){
6
7     echo "대림대학교";
8     echo $name ." = " . $num;
9 }
10 $name = daelim(박소현, 201840213);
11
```

이렇게 값을 2개를 받아서



이렇게 출력된다.

```
default.html x index.html hello 10.php hello11.php function_10.php x hello_12.php
phpstudy > php > function_10.php
1 <?php
2
3
4
5 function daelim ($name,$num){
6
7     echo "대림대학교<br>";
8     // echo $name ." = " . $num;
9     $arr = [$name , $num];
10    return arr;
11 }
12 $name = daelim(박소현, 201840213);
13 print $name;
```



```

1  <?php
2
3
4
5  function daelim ($name,$num){
6
7      echo "대림대학교<br>";
8      //      echo $name ." = ". $num;
9      $arr = [$name , $num];
10     return arr;
11 }
12 $name = daelim(박소현, 201840213);
13 print $name;
14
15

```

이렇게 되면 어떤 값을 출력하는지 모르기 때문에 지정을 해줘야 한다.

```

1  <?php
2
3
4
5  function daelim ($name,$num){
6
7      echo "대림대학교<br>";
8      //      echo $name ." = ". $num;
9      $arr = [$name , $num];
10     return arr;
11 }
12 $name = daelim("박소현", 201840213);
13 print_r($name);
14 echo $name[0 ] . " = " . $name[1];
15
16

```

```

1  <?php
2
3  function daelim ($name,$num) :array
4  {
5
6      echo "대림대학교<br>";
7      //      echo $name ." = ". $num;
8      $arr = [$name , $num];
9      return arr;
10 }

```

```
1 <?php
2 include_once "function_11.php";
3 include_once "function_11.php";
4 list($name, $num) = daelim("대림이",201840213);
5 // $name <--- $name[0]
6 // $num <---$name[1]
7 // print_r[name];
8
9 echo $name . "=" . $num;
10
11
```

Include 같은경우 호출한 만큼 나오게 되는데 그런 경우의 수가 나오지 못하게 한중복된 값이 한번만 나오게 해주는것이다.

```
1 <?php
2 echo"hello";
```

이렇게 해주면 바로 나오게 된다.

```
1 <?php
2 function greeting(){
3     echo"hello";
4 }
5
6
7 greeting();
```

```
1 <?php
2 function greeting(){
3     echo"hello";
4 }
5
6 function greeting(){
7     echo"hello";
8 }
9
10
11 greeting();
```

이렇게 하면 이름이 중복되기 때문에 안된다.

```
Debug Terminal Help
hello11.php x function_10.php index.php class_01.php x
phpstudy > php > class > class_01.php
1 <?php
2 class left{
3
4     function greeting(){
5         echo"hello";
6     }
7
8
9 }
```

그래서 이렇게 class 를 만들어주고 이름을 지어준다.

```
hello11.php function_10.php index.php class_01.php class_02.php x
phpstudy > php > class > class_02.php
1 <?php
2 class right{
3
4     function greeting(){
5         echo"hello";
6     }
7
8
9 }
```

```
hello11.php function_10.php index.php x class_01.php class_02.php
phpstudy > php > class > index.php
1 <?php
2 include "class_01.php";
3 include "class_02.php";
4
```

이렇게 해주면 된다.

그러면 일단 호출은 안해서 나오지는 않는다.

사용을 할려면

Index.php에 선언을 해줘야 한다.

클래스를 생성하는것을 인스턴스화라고 하는데 클래스를 인스턴스 해줘야 한다.

클래스의 결과물을 인스턴스라고 한다.

```
6 $obj1 = new left;
```

Left라는 클래스를 인스턴스화 해서 obj1이라는 객체를 생성해서 안에 넣어주는 것이다.

```
$obj1 = new left;
var_dump($obj1);
```

```
$obj1 ->greeting();
```

```
index.php x
phpstudy > php > class > index.php
1 <?php
2 include "class_01.php";
3 include "class_02.php";
4
5
6 $obj1 = new left;
7 var_dump($obj1);
8
9 $obj1 ->greeting();
```

localhost:8080/php/class x +  
localhost:8080/php/class  
object(left)#1 (0) { } hello

이렇게 하면 출력  
이 된다.

```
index.php x class_01.php
phpstudy > php > class > index.php
1 <?php
2 include "class_01.php";
3 include "class_02.php";
4
5
6 $obj1 = new left;
7
8 var_dump($obj1);
9
10 $obj1 ->greeting();
11
12 $obj2 = new right;
13 var_dump($obj2);
14 $obj2 ->greeting();
```

그리고 각각을 정의 해주면

```
index.php x class_01.php
phpstudy > php > class > index.php
1 <?php
2 include "class_01.php";
3 include "class_02.php";
4
5
6 $obj1 = new left;
7
8 var_dump($obj1);
9
10 $obj1 ->greeting();
11 echo "<br>";
12 $obj2 = new right;
13 var_dump($obj2);
14 $obj2 ->greeting();
```

localhost:8080/php/class x +  
localhost:8080/php/class  
object(left)#1 (0) { } left hello  
object(right)#2 (0) { } rightg hello



이렇게 나오게 된다.

```
ip function_10.php autoload.php class_01.php foo.php class_ ...
phpstudy > php > class > foo.php
1 <?php
2
3 class foo{
4     public static $aaa ="foo hahaha";
5     public function hello(){
6         echo self :: $aaa;
7     }
8 }
```

```
index.php class_01.php
phpstudy > php > class > index.php
21 // echo <br>;
22
23 // $obj1 ->daelim();
24 // $result ->daelim();
25
26 // //메소드 제안
27 // echo "--<br>";
28 // $obj1->greeting()->daelim()->foo()->bar();
29
30 echo "--<br>";
31 echo foo::$aaa = "foo hahaha";
32 echo "--<br>";
33 foo::hello();
```

```
<?php
// 선언
class foo{
    // 변수, 함수
    public $name; // 프로퍼티
    public static $daelim = "대림대학교";

    public function hello(){
        echo self::$daelim.">>";
        echo $this->name."짱이에요<br>";
        // echo self::$name."일까요?";
        return $this;
    }

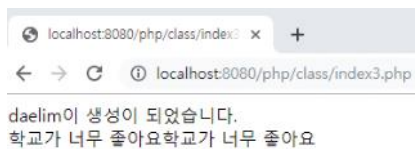
    public function greeting(){
        echo "반가워요<br>";
        return $this;
    }
}
```

```
$obj1 = new foo;
$obj1->name = "대림이"; // 객체 프로퍼티 값을 저장
$obj1->hello();

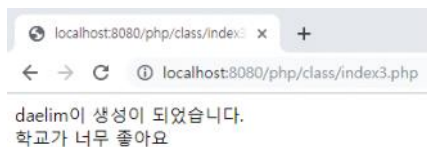
$obj2 = new foo;
$obj2->name = "대숙이";
$obj2->hello();

foo::$daelim = "우주최강";
$obj1->hello();
$obj2->hello();
```

```
ion_10.php x autoload.php class_01.php foo.php index2.php index3.php class_02.php
phpstudy > php > class > index3.php
1 <?php
2
3
4 class daelim
5 {
6     public function __construct()
7     {
8         echo __CLASS__ . "이 생성이 되었습니다.<br>";
9         $this->hello();
10    }
11    public function hello()
12    {
13        echo "학교가 너무 좋아요";
14    }
15 }
16
17 $obj1 = new daelim;
18 $obj1->hello();
```



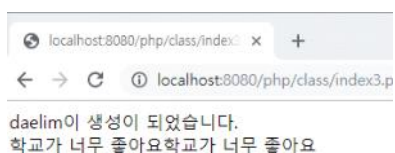
```
private function hello()
{
    echo "학교가 너무 좋아요";
}
```



접근 제한자가 private로 변경 되었기 때문에 이렇게 나오는 것이다.

Private 에서는 this로의 접근만 가능 하다.

```
ion_10.php x autoload.php class_01.php foo.php index2.php index3.php class_02.php
phpstudy > php > class > index3.php
1 <?php
2
3
4 class daelim
5 {
6     public function __construct()
7     {
8         echo __CLASS__ . "이 생성이 되었습니다.<br>";
9         $this->hello();
10    }
11    public function hello()
12    {
13        echo "학교가 너무 좋아요";
14    }
15 }
16 class food extends daelim{
17     public function menu()
18     {
19         echo "학교 학식이 너무 맛이 없어요";
20     }
21 }
22 $obj1 = new food;
23 $obj1->hello();
```





```
protected function hello()
{
    echo "학교가 너무 좋아요";
}
```

위의 public 을 protected로  
변경 해준다.

```
16 class food extends daelim{
17     public function menu()
18     {
19         echo "학교 학식이 너무 맛이 없어요";
20     }
21 }
22 $obj1 = new food;
23 $obj1->hello();
24 $obj1->menu();
```

그리고 새로 daelim class를 추가해 준다.

```
final public function hello()
{
    echo "학교가 너무 높아요";
}
```

오버라이딩을 못하게 하기 위해 final 을 사용한다.

```
<?php

abstract class greeting // 추상클래스
{
    // 이것은 꼭 만들어야해
    abstract public function hello();

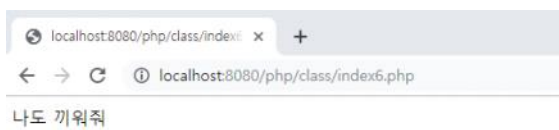
    public function pages()
    {
        echo "페이지 출력";
    }
}

class daelim extends greeting{
    public function hello(){
        echo "그냥 해";
    }
}

$obj = new daelim;
$obj->hello();
$obj->pages();
```

localhost:8080/php/class/index5  
localhost:8080/php/class/index5.php  
그냥 해페이지 출력

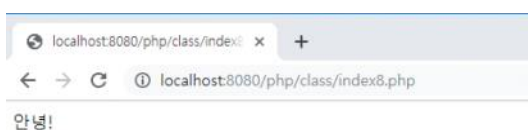
```
class_01.php • foo.php index2.php • index3.php index4.php index5.php index6.php x
phpstudy > php > class > index6.php
1 <?php
2
3 class aaa{
4
5     use abcd;
6
7 }
8 trait abcd{
9     public function join(){
10         echo "나도 끼워줘";
11     }
12 }
13 $obj = new aaa;
14 $obj->join();
15
```



Trait 혼자서 객체를 만들수 없음 그래서 class 안에 넣어줘야 함 필요한 부분을 만들어 놓고 사용  
이 가능 하다.  
Trait 는 상속이 가능하다.

네임 스페이스

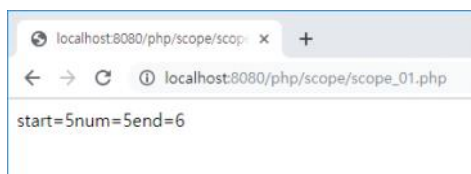
```
index5.php index6.php index7.php x index8.php x
phpstudy > php > class > index7.php
1 <?php
2
3 namespace sh\daelim;
4
5 class aaa{
6     public function hello(){
7         echo "안녕!";
8     }
9 }
phpstudy > php > class > index8.php
1 <?php
2
3 include"index7.php";
4
5
6 use \sh\daelim\aaa;
7 $obj= new aaa;
8 $obj->hello();
```



```
index8.php x
phpstudy > php > class > index8.php
1 <?php
2
3 include"index7.php";
4
5
6 use \sh\daelim as bbbb;
7 $obj= new bbbb\aaa;
8 $obj->hello();
```

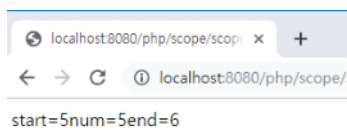
이렇게도 사용이 가능하다.

```
scope_01.php x
phpstudy > php > scope > scope_01.php
1 <?php
2
3 $a = 5;
4 echo "start=".$a;
5
6 function num(){
7     // $a = 2;
8     global $a;
9     echo "num=".$a;
10
11     $a = 6;
12 }
13
14 num();
15 echo "end=".$a;
16
```



Global 같은 경우 밖에 있는 값을 받을수 있기 때문에 5라는 값을 가져올수 있다.

```
scope_01.php x scope_02.php x
phpstudy > php > scope > scope_02.php
1 <?php
2
3 $a = 5;
4 echo "start=".$a;
5
6 function num(){
7     // $a = 2;
8     echo "num=".$GLOBALS['a'];
9
10     $GLOBALS['a'] = 6;
11 }
12
13 num();
14 echo "end=".$a;
15
```



```

scope_01.php x scope_02.php x
phpstudy > php > scope > scope_02.php
1  <?php
2
3  $a = 5;
4  echo "start=".$a;
5
6  function num(){
7      // $a = 2;
8      echo "num=".$GLOBALS['a'];
9
10     $GLOBALS['a'] = 6;
11 }
12
13 num();
14 echo "end=".$a;
15
16
17 echo $_SERVER['REQUEST_URI'];
18

```

```

localhost:8080/php/scope/scope_02.php
start=5num=5end=6/php/scope/scope_02.php

```

웹 서버와의 통신에서 전달 받는 값들을 변수로 받을 수 있다.