Week 4
PHP Static Statement

The PHP Static Statement

The beauty of the PHP static statement is that it allows a function to "remember" the value of a local variable for the next time the function is called.

The PHP static statement gives a PHP function a "memory". A basic PHP function does not have the ability to remember the value of a local variable that has been created or changed within the function. Once the function is finished, all knowledge of the value of the function's variable disappears. To get around this problem we can use the PHP static statement to tell the function to remember the value of a local variable from one call of a function to the next call. Let's take a look at the basic code.

```
function count_calls( )
{
static $no_calls = 0;
$no_calls++;
echo "This function has been called $no_calls
times.";
}
```

function function_name( )

```
function count_calls( )
```

The function name identifies this function and the instructions for the function are placed between the following { and }.

static $variable = value;

```
static $no_calls = 0;
```

This is the static statement which is used to keep a running total of (and remember) the number of times this function has been called. The first time the function is called, the static statement will initialize the $no_calls variable to the initial value of 0. Then as the function continues, the value of the variable is increased by 1. The second and subsequent times the function is called, the increased value of the variable is remembered from the last time. Without the static statement the value of the variable will be zero each time this function is called.

$variable++;

```
$no_calls++;
```

This expression increases the current value of the $no_calls variable by 1. The beauty of the static statement is that the function can now "remember" the increased value of the $no_calls variable for the next time the function is called.

echo statement

```
echo "This function has been called $no_calls
times.";
```

This echo statement will print the text and the new (increased) value of the $no_calls variable to the web browser.

This is all there is to our function. The next time this function is called, the value of the $no_calls variable is remembered from the last execution of the function and then increased by one again.

(c) 2008, 2009 Etudes, Inc. All rights reserved.
ETUDES - 2.3.x - Sakai 2.3.x - Server "athena69"