

PHP 5.5 new features

2014.7.9 PHP 사용자 모임
이현석

늘어났다고 생각할 때가
정말 높은 거다

2013년 6월 php 5.5.0 릴리즈

- 해당 사항 없음 ↗

php 5.5에 포함된 핵심 기능

http://php.net/releases/5_5_0.php

- generators 와 coroutines 추가.
- foreach 문장에 non-scalar 키를 지원
- foreach 문장에 list() 를 쓸 수 있게 됨.
- finally 키워드 추가.
- 간단한 비밀번호 해싱 함수 추가.
- 상수 배열이나 문자열의 역참조 추가.
- ::class 키워드로 클래스의 전체 이름 확인.
- 변수 외의 임의의 표현식을 empty() 표현식에 사용 가능.
- Zend OPcache 기본 포함.
- The GD library 2.1로 업그레이드.

php 5.5에 포함된 핵심 기능

http://php.net/releases/5_5_0.php

- generators 와 coroutines 추가.
- foreach 문장에 non-scalar 키를 지원
- foreach 문장에 list() 를 쓸 수 있게 됨.
- finally 키워드 추가.
- 간단한 비밀번호 해싱 함수 추가.
- 상수 배열이나 문자열의 역참조 추가.
- ::class 키워드로 클래스의 전체 이름 확인.
- 변수 외의 임의의 표현식을 empty() 표현식에 사용 가능.
- Zend OPcache 기본 포함.
- The GD library 2.1로 업그레이드.

Generators provide an easy way to implement simple iterators without the overhead or complexity of implementing a class that implements the Iterator interface.

Generators 는 좀 짱나는 Iterator 인터페이스를 구현
하는 클래스 구현
없이 간단한 이터레이터를 구현
하는 쉬운 방법을 제공한다.



어떤 함수든지 yield문을 포함하고 있다면 Generator 함수다.

왜 쓰나요?

foreach 를 사용할 때 배열을 안 쓸 수 있다.

= 메모리 사용 ∇ + 시간 소모 ∇

큰 범위를 순회할 경우가 아니면 쓸 필요 없는 듯?

Generators에 대해 한글로 잘 정리된 문서
<https://crizin.net/display/dev/Generators>

php 5.5에 포함된 핵심 기능

http://php.net/releases/5_5_0.php

- generators 와 coroutines 추가.
- **foreach 문장에 non-scalar 키를 지원**
- foreach 문장에 list() 를 쓸 수 있게 됨.
- finally 키워드 추가.
- 간단한 비밀번호 해싱 함수 추가.
- 상수 배열이나 문자열의 역참조 추가.
- ::class 키워드로 클래스의 전체 이름 확인.
- 변수 외의 임의의 표현식을 empty() 표현식에 사용 가능.
- Zend OPcache 기본 포함.
- The GD library 2.1로 업그레이드.

non-scalar?

스칼라 데이터란 하나의 데이터에 한 개의 값만이 올 수 있는 값이다.

참고 : 스칼라 변수와 데이터

boolean

integer

float(소수, double)

string

PHP 5.5 이전에는 이랬다고 함

```
$it = new MultipleIterator;  
$it->attachIterator($it1);  
$it->attachIterator($it2);
```

```
// This is NOT possible  
foreach ($it as $keys => $values) {  
    // ...  
}
```

```
// Instead you have to use this  
foreach ($it as $values) {  
    $keys = $it->keys();  
  
    // ...  
}
```

php 5.5에 포함된 핵심 기능

http://php.net/releases/5_5_0.php

- generators 와 coroutines 추가.
- foreach 문장에 non-scalar 키를 지원
- **foreach 문장에 list() 를 쓸 수 있게 됨.**
- finally 키워드 추가.
- 간단한 비밀번호 해싱 함수 추가.
- 상수 배열이나 문자열의 역참조 추가.
- ::class 키워드로 클래스의 전체 이름 확인.
- 변수 외의 임의의 표현식을 empty() 표현식에 사용 가능.
- Zend OPcache 기본 포함.
- The GD library 2.1로 업그레이드.

list()

배열인것 처럼 변수를 할당한다

```
$info = array('coffee', 'brown', 'caffeine');
```

```
// 모든 변수를 목록화한다
```

```
list($drink, $color, $power) = $info;
```

```
echo "$drink is $color and $power makes it special.\n";
```

```
// 그들중 일부를 목록화한다
```

```
list($drink, , $power) = $info;
```

```
echo "$drink has $power.\n";
```



```
$users = array(  
    array('Foo', 'Bar'),  
    array('Baz', 'Qux'),  
);
```

// Before

```
foreach ($users as $user) {  
    list($firstName, $lastName) = $user;  
    echo "First name: $firstName, last name: $lastName. ";  
}
```

// After

```
foreach ($users as list($firstName, $lastName)) {  
    echo "First name: $firstName, last name: $lastName. ";  
}
```

php 5.5에 포함된 핵심 기능

http://php.net/releases/5_5_0.php

- generators 와 coroutines 추가.
- foreach 문장에 non-scalar 키를 지원
- foreach 문장에 list() 를 쓸 수 있게 됨.
- **finally 키워드 추가.**
- 간단한 비밀번호 해싱 함수 추가.
- 상수 배열이나 문자열의 역참조 추가.
- ::class 키워드로 클래스의 전체 이름 확인.
- 변수 외의 임의의 표현식을 empty() 표현식에 사용 가능.
- Zend OPcache 기본 포함.
- The GD library 2.1로 업그레이드.

예외를 다루기 위한 문법이 try, catch 밖에 없었지만
이제 Java 처럼 finally도 사용할 수 있다

Finally 블록은 항상 실행됩니다.

```
<?php
$db = mysqli_connect();
try {
    call_some_function($db);
} catch (Exception $e) {
    mysqli_close($db);
    throw $e;
}
mysqli_close($db);
```

```
<?php
$db = mysqli_connect();
try {
    call_some_function($db);
} finally {
    mysqli_close($db);
}
```

php 5.5에 포함된 핵심 기능

http://php.net/releases/5_5_0.php

- generators 와 coroutines 추가.
- foreach 문장에 non-scalar 키를 지원
- foreach 문장에 list() 를 쓸 수 있게 됨.
- finally 키워드 추가.
- **간단한 패스워드 해싱 함수 추가.**
- 상수 배열이나 문자열의 역참조 추가.
- ::class 키워드로 클래스의 전체 이름 확인.
- 변수 외의 임의의 표현식을 empty() 표현식에 사용 가능.
- Zend OPcache 기본 포함.
- The GD library 2.1로 업그레이드.

md5 나 sha512 만으로는 충분치 않다.

더 뻥센게 필요하지만
대부분의 php 개발자들은 능력이 딸리거나,
노력이 필요하다고 생각하지 않아서 간단하게 만들었다고 함.

password_get_info()
해시된 값의 정보 추출

password_hash()
해시를 실행

password_needs_rehash()
제대로 해시되었는지 검사

password_verify()
입력값과 해시값 비교

```
$password = "foo";  
$hash = password_hash($password,  
PASSWORD_DEFAULT);  
// 패스워드의 해시 값을 저장한다
```

```
if (password_verify($password, $hash)) {  
    // 패스워드가 맞음  
}  
else {  
    // 패스워드가 틀림  
}
```

php 5.5에 포함된 핵심 기능

http://php.net/releases/5_5_0.php

- generators 와 coroutines 추가.
- foreach 문장에 non-scalar 키를 지원
- foreach 문장에 list() 를 쓸 수 있게 됨.
- finally 키워드 추가.
- 간단한 비밀번호 해싱 함수 추가.
- **상수 배열이나 문자열의 역참조 추가.**
- ::class 키워드로 클래스의 전체 이름 확인.
- 변수 외의 임의의 표현식을 empty() 표현식에 사용 가능.
- Zend OPcache 기본 포함.
- The GD library 2.1로 업그레이드.

Q 결과는?

```
echo array(1, 2, 3)[0];  
echo "foobar"[2];  
echo [1,3,4][2];
```

A

```
echo array(1, 2, 3)[0];  
echo "foobar"[2];  
echo [1,3,4][2];
```

php 5.5에 포함된 핵심 기능

http://php.net/releases/5_5_0.php

- generators 와 coroutines 추가.
- foreach 문장에 non-scalar 키를 지원
- foreach 문장에 list() 를 쓸 수 있게 됨.
- finally 키워드 추가.
- 간단한 비밀번호 해싱 함수 추가.
- 상수 배열이나 문자열의 역참조 추가.
- **::class 키워드로 클래스의 전체 이름 확인.**
- 변수 외의 임의의 표현식을 empty() 표현식에 사용 가능.
- Zend OPcache 기본 포함.
- The GD library 2.1로 업그레이드.


```
namespace MyNameSpace;  
class TestClass {}
```

```
echo TestClass::class; //=> MyNameSpace\TestClass
```

php 5.5에 포함된 핵심 기능

http://php.net/releases/5_5_0.php

- generators 와 coroutines 추가.
- foreach 문장에 non-scalar 키를 지원
- foreach 문장에 list() 를 쓸 수 있게 됨.
- finally 키워드 추가.
- 간단한 비밀번호 해싱 함수 추가.
- 상수 배열이나 문자열의 역참조 추가.
- ::class 키워드로 클래스의 전체 이름 확인.
- **변수 외의 임의의 표현식을 empty() 표현식에 사용 가능.**
- Zend OPcache 기본 포함.
- The GD library 2.1로 업그레이드.

아래 값들을 모두 false 로 인식함

- "" (an empty string)
- 0 (0 as an integer)
- 0.0 (0 as a float)
- "0" (0 as a string)
- NULL
- FALSE
- array() (an empty array)
- \$var; (a variable declared, but without a value)

php 5.5에 포함된 핵심 기능

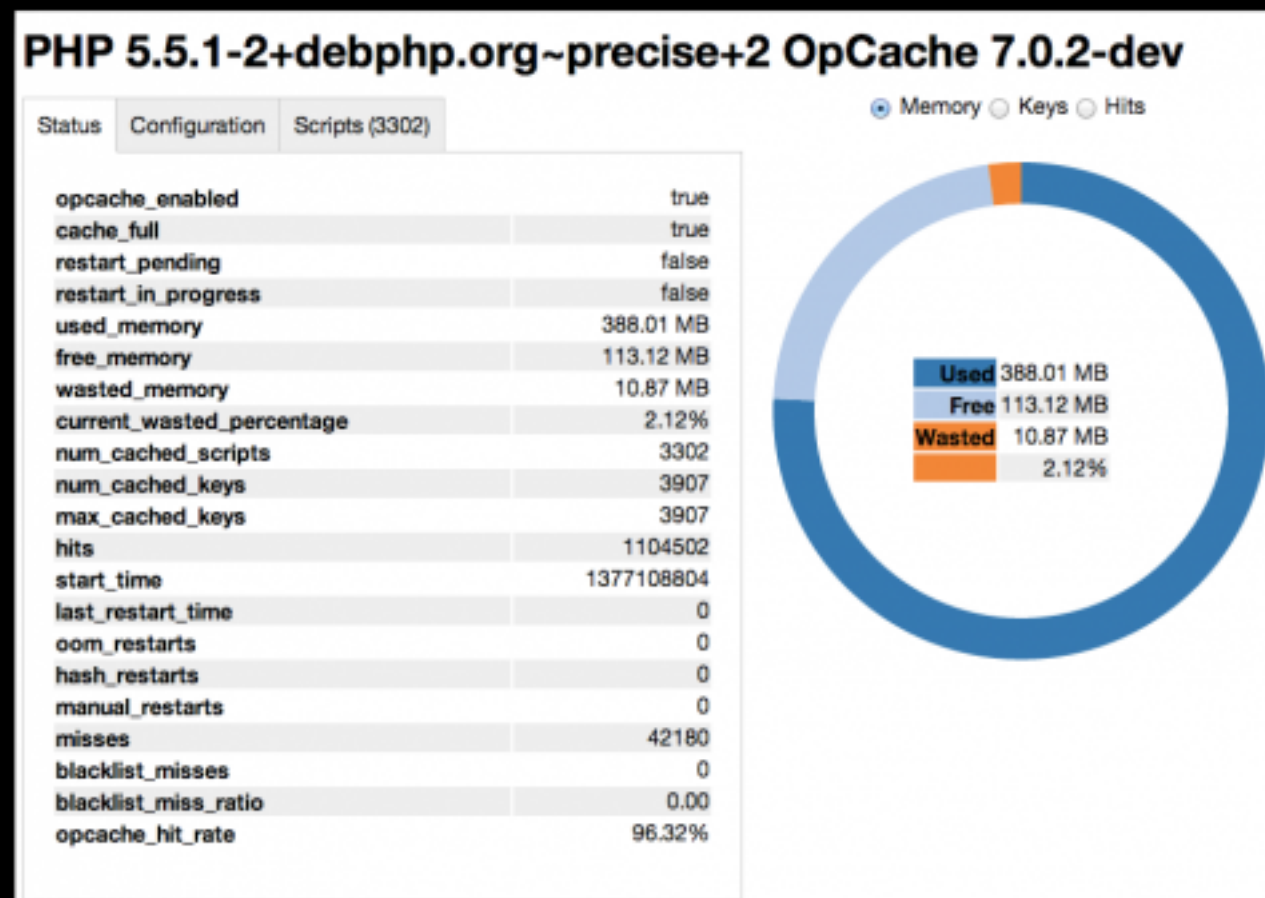
http://php.net/releases/5_5_0.php

- generators 와 coroutines 추가.
- foreach 문장에 non-scalar 키를 지원
- foreach 문장에 list() 를 쓸 수 있게 됨.
- finally 키워드 추가.
- 간단한 비밀번호 해싱 함수 추가.
- 상수 배열이나 문자열의 역참조 추가.
- ::class 키워드로 클래스의 전체 이름 확인.
- 변수 외의 임의의 표현식을 empty() 표현식에 사용 가능.
- **Zend OPcache 기본 포함.**
- The GD library 2.1로 업그레이드.

Opcache 가 APC 보다 5~20% 빠르다고 함.
그리고 Opcache 는 php에 5.5 이상에 기본 포함이기 때문에
한 달 째 느리게 대응되는 APC 를 쓸 이유가 없다고 함.

Opcache Web viewer

<https://rtcamp.com/tutorials/php/zend-opcache/>



합계

```
LeeHyunseok-ui-MacBook-Pro:pug leehyunseok$ php finally.php
PHP Parse error:  parse error, expecting `catch (T_CATCH)'' in /Users/leehyunseok/
Sites/pug/finally.php on line 11
LeeHyunseok-ui-MacBook-Pro:pug leehyunseok$ php -v
PHP 5.4.24 (cli) (built: Jan 19 2014 21:32:15)
Copyright (c) 1997-2013 The PHP Group
Zend Engine v2.4.0, Copyright (c) 1998-2013 Zend Technologies
LeeHyunseok-ui-MacBook-Pro:pug leehyunseok$
```

General

Hosts

Modules

PHP

MySQL

Dynamic DNS

Postfix

Default version: 5.5.10






This version will be used when running PHP in module mode and when creating new hosts.

Mode: ☒ Identical PHP version for all hosts (module)
☐ Individual PHP version for every host (CGI)

Cache module to speed up PHP execution: off

Debugger: ☐ Activate Xdebug[Open MacGDBp](#)

What to log: ☐ Display startup errors
☒ All other errors and warnings
☐ Errors
☐ Warnings
☐ Notices
☐ Others

Log errors: ☐ to screen☒ to log file:  Macintosh HD >  응용 프로그램 >  MAMP >  logs >  php_error.log

MAC 기본 PHP 대신 MAMP 의 PHP 쓰기

I have created a symlink at the original php location.

1. Locate your osx php version with:

```
which php
```

The result should be:

```
/opt/local/bin/php
```

2. Backup (move) your original php binary:

```
sudo mv /opt/local/bin/php /opt/local/bin/php.bak
```

3. Create the symlink:

```
sudo ln -s /Applications/MAMP/bin/php/php5.4.4/bin/php /opt/local/bin/php
```

4. Run your new php version:

```
php -v
```

Now you can use the php command with the newest version.

How to use MAMP's version of PHP instead of the default on OSX