Is it time ?

2013-08-21

Alfonsas Cirtautas @acirtautas **#KaunasPHP** v.5

PHP 5.5

- Status
- New
- Removed
- Deprecated
- IDE
- Hosting

Status

5.5.0 – 2013-06-20

5.5.1 – 2013-07-18

5.5.2 – 2013-08-16

New

- generators (yield)
- finally (try, catch, ...)
- new password hashing api
- foreach + lists
- empty + expressions
- array/string literal deferencing
- early class name resolution
- OPcache build in (ex Zend Optimizer+)
- etc. (GD, CURL, array_column, boolval, ...)

http://www.php.net/manual/en/migration55.new-features.php

Generators

```
function rangeGenerator($start, $limit, $step) {
    $i = 1;
    for ($n = $start; $n <= $limit; $n += $step) {
        yield $i => $n;
        $i++;
    }
}


$range = rangeGenerator(1,100,2);
foreach ($range as $i => $n) {
    echo $i,'=>',$n,' ';
}
```

```
function printGenerator() {
    $i = 1;
    while(true) {
        $n = yield;
        echo $i,'=>',$n,' ';
        $i++;
    }
}

$print = printGenerator();
for ($n = 1; $n <= 100; $n+=2) {
    $print->send($n);
}
```

- Generators provides an easy and boilerplate-free way to create iterators.
- Functions which remain state between runs
- Local variables are persistent

Finally...

- New keyword to try..catch
- Block is always executed!

```
try {
    doGood();
} catch (Exception $e) {
    expectWorst($e);
} finally {
    keepCalmAndCleanUpAfter();
}
```

Password hashing API

Easy wrapper for crypt()

- **password_hash** — Creates a password hash
- **password_verify** — Verifies that a password matches a hash
- **password_get_info** — Returns information about the given hash
- **password_needs_rehash** — Checks if the given hash matches the given options

```
$hash1 = password_hash(“Password1");
$hash2 = password_hash(“Password1", PASSWORD_BCRYPT,  [ 'cost' => 12]);
```

Foreach + lists

- **Foreach()** now supports lists

```
$array = [ [1, 2], [3, 4] ];
foreach ($array as list($a, $b)) {
     echo "A: $a; B: $b\n";
}
```

Empty + expressions

- **Empty()** now works with expressions

```
function nowWorks() {
    return 0;
}

echo empty( nowWorks());
```

array/string literal deferencing

- “Constant dereferencing” means that array operations can be directly applied to string and array literals

```
echo [“Kaunas”, “PHP”, “v5”][1];


echo “KaunasPHP v.5”[12];
```

array/string literal deferencing

- new ClassName**::class** syntax provides the fully qualified class name as a string.

```
use MyVendor\SomeComponent\TargetNs as T;


if (class_exists(T\Foo::class, true))
```

Opcache build in.

- Zend Optimizer+ open sourced !!!
- Build in to PHP 5.5, always in sync
- Fastest opcode cache for PHP
- Will work together with XDebug

etc.

- Additions to **GD lib**
  - Imagecrop
  - Imagecropauto
  - Imagecreatefromwebp
  - Imagewebp
- **Additions to cURL**
  - **FTP**
  - **SMTP !!!**
  - **SSL**
- Still no native unicode support, php 6 ?

- **DateTimeImmutable** class
- **array_column()** function
- **boolval()** function

Removed

- PHP logo GUIDs

php
php
php
Powered By
Zend Engine 2

- Windows XP / 2003 support dropped
- Case insensitivity no longer locale specific
- self, parent and static are now always case insensitive
- pack() and unpack() changes to make them more compatible with Perl
- Internal execution changes

http://www.php.net/manual/en/migration55.incompatible.php

Deprecated

- MySQL extension (use MySQLi or PDO_MySQL)
- Preg_replace() /e modifier (use preg_replace_callback)

http://www.php.net/manual/en/migration55.deprecated.php

IDE

- - Zend Studio 10.1 / Zend Server 6.1.0 (???)
- - PHPStorm 6 (+v7 | EAP +)
- - NetBeans 7.3 (+v7.4 | Beta ?)

Hosting

- localhost ?

Is it time ?

“If you are just getting started with PHP make sure to start with the current stable release of PHP 5.5. PHP has made great strides adding powerful new features over the last few years. Don’t let the minor version number difference between 5.2 and 5.5 fool you, it represents *major* improvements. “

phptherightway.com