미리보는 PHP5.6...

전창완 (me@wani.kr)

한상현
PHP 5.6 브랜치가 만들어졌네요.

https://github.com/php/php-src/tree/PHP-5.6

php-src
github.com
php-src - The PHP Interpreter

좋아요 취소 · 댓글 달기 · 공유하기 · 1월 14일 오후 1:16 서울 근처

회원님, 정양파님 외 4명이 좋아합니다.

그때 부터였던 것 같아요..

PUG

PHP 5.6.0alpha1 released » 2014-01-23

The PHP development team announces the immediate availability of PHP 5.6.0alpha1. This release marks the beginning of the PHP 5.6.0 release cycle. All users of PHP are encouraged to test this version carefully, and report any bugs in the bug tracking system.

그리고.. alpha1의 release..

근데 beta는 알아도 alpha가 뭐죠..?

PUG

알파 버전

위키백과, 우리 모두의 백과사전.

> 이 문서의 내용은 출처가 분명하지 않습니다.
> 지금 바로 이 문서를 편집하여, 참고하신 문헌이나 신뢰할 수 있는 출처를 주석 등으로 표기해 주세요. 검증되지 않은 내용은 삭제될 수도 있습니다. 내용에 대한 의견은 토론 문서에서 나누어 주세요.

알파 버전은 개발 초기에 있어 성능이나 사용성 등을 평가하기 위한 테스터나 개발자를 위한 버전이다.

테스터로서 알파 버전을 테스트할 때에는 최악의 경우 시스템이 파괴되는 것도 각오해 둘 필요가 있다.

되는 것이므로 일반인은 사용해서는 안된다. 테스터로서 알파 버전을 테스트할 때에는 최악의 경우 시스템이 파괴되는 것도 각오해 둘 필요가 있다.

알파 버전에서 나온 문제점이나 버그를 개선하여 유저에 시험 사용을 받기 위한 버전이 베타 버전이다.

목숨걸고 준비했습니다..

목차

- New Features
  - 상수 계산 지원
  - “use function”, “use const” 구문 추가
  - “gost_crypto” 해시함수 추가
  - …Super Powerful Expression!!
- Deprecated
  - None static method의 Static 접근이 Strict에서 Deprecated로 조절
- …Etc

가장~
마지막에!

New Features

- 상수 계산 지원

```php
<?php
const ONE = 1;
const TWO = ONE * 2;

const SETENCE = "This is The Sentence, TWO is ".TWO;

echo ONE, "<br />", SETENCE;
```

1
This is The Sentence, TWO is 2

New Features

- “use function”, “use const” 구문 추가

```php
<?php
namespace Foo\Bar {
    const MY_CONST = 30;
    function foo() {
        return __FUNCTION__;
    }
}


namespace {
    use const \Foo\Bar\MY_CONST;
    use function \Foo\Bar\foo;

    echo MY_CONST, "<br />", foo();

}
```

```
30
Foo\Bar\foo
```

```php
<?php
namespace Foo\Bar {
    const MY_CONST = 30;
    function foo() {
        return __FUNCTION__;
    }
}


namespace {
    use const \Foo\Bar\MY_CONST as MY_CONSTT;
    use function \Foo\Bar\foo as fooo;

    echo MY_CONSTT, "<br />", fooo();

}
```

New Features

- “gost_crypto” 해시함수 추가

```php
<?php
echo hash("gost-crypto", "hello");

```

92ea6ddbaf40020df3651f278fd7151217a24aa8d22ebd2519cfd4d89e6450ea

Deprecated

- None static method의 Static 접근이 Strict에서 Deprecated로 조절

```php
<?php
class A {
    function f() { echo get_class($this); }
}

class B {
    function f() { A::f(); }
}

(new B)->f();
```

~~Strict Standards: Non-static method A::f() should not be called statically, assuming $this from incompatible context in /Users/Shared/Sites/sample/06static_call_deprecated.php on line 7~~
~~B~~

Deprecated: Non-static method A::f() should not be called statically, assuming $this from incompatible context in /usr/share/nginx/www/sample/06static_call_deprecated.php on line 7
B

Etc…

- phpdbg(SAPI module)의 기본 지원
- 2기가 넘는 파일 업로드 가능
- openssl_x509_fingerprint() 추가

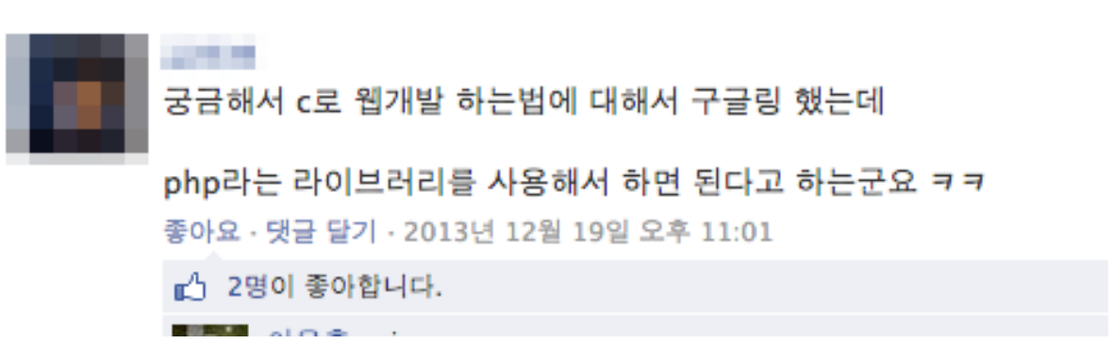
PUG

…Super Powerful Expression!!

• • •

- C에서도 있었다!

  printf(char*, …);



역시 C최고의 라이브러리 답군..!

PUG

```c
#include <stdarg.h>
#include <stdio.h>

void foo ( int num, ... ) {

    va_list args;

    va_start ( args, num );

    for ( int i = 0; i < num; i++ ) {

        printf( "%d's Value is '%c'\n", i+1, va_arg( args, int ));

    }

    va_end ( args );

}

int main() {

    foo(5, 'a','b','c','d','e');

    return 0;

}
```

C

```php
<?php

function foo() {
    $args = func_get_args();

    for ($i = 0; $len = count($args), $i < $len; $i++) {

        printf( "%d's Value is '%s'\n", $i+1, $args[$i]);

    }
}


foo('a','b','c','d','e');

```

과거의 PHP

```php
<?php

function foo(...$args) {

    for ($i = 0; $len = count($args), $i < $len; $i++) {

        printf( "%d's Value is '%s'\n", $i+1, $args[$i]);

    }
}


foo('a','b','c','d','e');

```

PHP 5.6

func_get_args()를 완벽하게 대체가능!

• • •

- 끝이 아니다!
- Python에서가져왔나..?

```
1
2   def foo( x, y, z ) :
3       print x
4       print y
5       print z
6
7
8
9   foo( 1,2,3 )
10
11  x = [1,2,3]
12
13  foo( *x );
14
```

```php
<?php

function foo( $x, $y, $z ) {
    echo $x;
    echo $y;
    echo $z;
}

foo(1,2,3);

$x = [1,2,3];

call_user_func_array('foo', $x);
```

과거의 PHP

```php
<?php

function foo( $x, $y, $z ) {
    echo $x;
    echo $y;
    echo $z;
}

foo(1,2,3);

$x = [1,2,3];

foo( ...$x );
```

PHP 5.6

call_user_func_array()를 완벽하게 대체가능!

성능비교

```
wani@ubuntu:/usr/share/nginx/www/sample$ php 0
Step1. func_get_args() vs ...
Running Time : 0.01286700s

Running Time : 0.00949300s

Step2. call_user_func_array() vs ...
Running Time : 0.01660500s

Running Time : 0.01022300s
```

```php
<?php
include "functions.php";

function average1() {
    $args = func_get_args();

    $total = 0;
    $count = 0;
    foreach ( $args as $arg ) {
        $total += $arg;
        $count++;
    }
    return $total/$count;
}

function average2(...$args) {
    $total = 0;
    $count = 0;
    foreach ( $args as $arg ) {
        $total += $arg;
        $count++;
    }
    return $total/$count;
}

$arr = [1,2,3,4,5];

echo "Step1. func_get_args() vs ... \n";

echo benchmark(function() {

    average1(1,2,3,4,5);

}, 10000), "\n";

echo benchmark(function() {

    average2(1,2,3,4,5);

}, 10000), "\n";

echo "Step2. call_user_func_array() vs ... \n";
echo benchmark(function() use($arr) {

    call_user_func_array('average2', $arr);

}, 10000), "\n";

echo benchmark(function() use($arr) {

    average2(...$arr);

}, 10000), "\n";
```

보기좋은 떡이 성능도 좋네!

PUG

• • •

정리하자면..

func_get_args()

call_user_func_array()

...

끝...