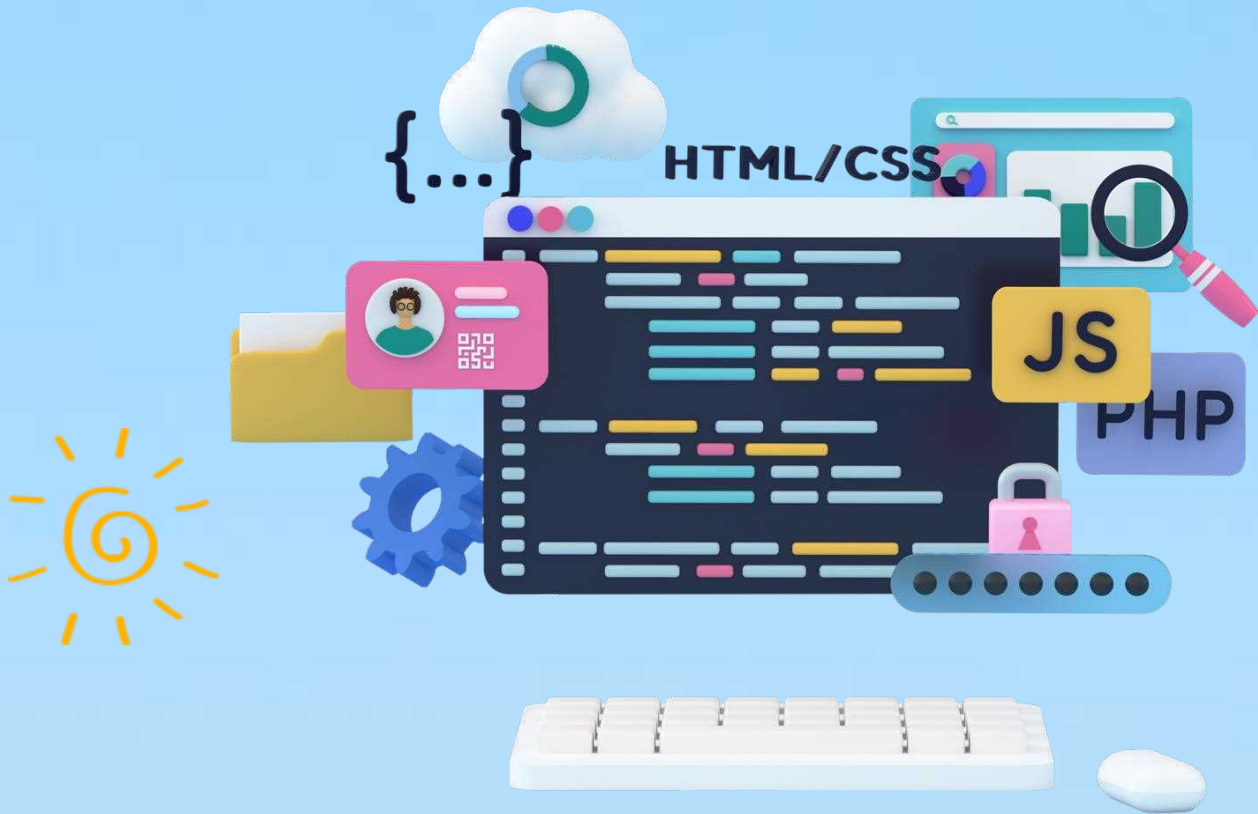


آموزش php



آموزش php



<https://iracode.com/php/>



۰۹۱۲۴۱۰۴۵۳۵

در این آموزش سعی کرده ایم که به زبانی ساده به آموزش زبان برنامه نویسی پی اچ پی پردازیم.
نصب و راه اندازی

```
<html>
<head>
<title>Online PHP Script Execution</title>
</head>
<body>
<?php
echo "<h1>Hello, PHP!</h1>";
?>
</body>
</html>
```

به منظور توسعه و اجرای صفحات وب (PHP) ، سه جزء مهم لازم است تا بر روی سیستم کامپیوتر شما نصب شود .

وب سرور:

به صورت مجازی با همه نرم افزار های وب سرور کار می کند ، دربردارنده سرور اطلاعات اینترنت مایکروسافت (IIS) است ، اما بیشتر به عنوان سرور آپاچی استفاده می شود .

به رایگان آپاچی را این لینک دانلود کنید : <http://httpd.apache.org/download.cgi> : دیتابیس :

به صورت مجازی با همه نرم افزار های دیتابیس ، دربردارنده (Oracle) و (Sybase) کار می کند ، اما بیشتر از عنوان (MySQL database) استفاده می شود .

به رایگان (MySQL) را از این لینک دانلود کنید : <http://www.mysql.com/downloads/>

parser : به منظور روند دستورالعمل های اسکریپت ، یک parser باید نصب باشد تا (HTML) تولید شده بتواند به مرورگر وب فرستاده شود . این آموزش شما را راهنمایی می کند ، که چطور (PHP) parser را روی کامپیوتر خود نصب کنید .

نصب کردن php parser

قبل از اینکه شما شروع کنید ، مهم است که مطمئن شوید که شما محیط مناسب را بر روی دستگاه خود راه اندازی کرده اید ، تا برنامه های وب را با استفاده از (php) توسعه دهید .

فایل (php) روبرو را ذخیره کرده و فولدر (Apache) در (htdocs) بریزید .

```
<?php
phpinfo();
?>
```

<http://127.0.0.1/phpinfo.php>

آدرس زیر را در مرورگر خود وارد کنید :

صفحه ای حاوی اطلاعات نصب (php) به شما نشان داده می شود ، که به این معناست که شما (php) و وب سرور را به درستی نصب کرده اید ؛ در غیر این صورت ، شما باید روش های داده شده را دنبال کنید تا (php) را روی کامپیوتر خود نصب کنید .
این قسمت شما را راهنمایی می کند تا (php) را روی ۴ پلتفرم زیر پیکربندی و نصب کنید :

PHP Installation on Linux or Unix with Apache

PHP Installation on Mac OS X with Apache

PHP Installation on Windows NT/2000/XP with IIS

PHP Installation on Windows NT/2000/XP with Apache

پیکربندی آپاچی

اگر شما از آپاچی به عنوان وب سرور استفاده می کنید ، خوب این قسمت شما را راهنمایی می کند ، تا فایل های پیکربندی آپاچی را ویرایش کنید .

اینجا را بررسی کنید: PHP Configuration in Apache Server:

پیکربندی فایل php.ini

فایل پیکربندی php.ini ، این آخرین و فوری ترین راه برای تحت تأثیر قرار دادن عملکرد (php) است .

اینجا را بررسی کنید: PHP.INI File Configuration:

پیکربندی IIS Windows

برای پیکربندی IIS بر روی دستگاه خودتان ، شما می توانید به راهنمای همراه با (IIS) مراجعه کنید .

متد های منسوخ شده

در php4 ، متد هایی هستند که تعریف شده اند و هم اسم هستند ؛ حالا منسوخ شده و در آینده حذف می شوند . اگر یک سازنده php4 ، تنها سازنده تعریف شده در یک کلاس بود ، E_DEPRECATED در php7 منتشر می شود .

کلاس های اجرای یک متد __construct () بدون تاثیر هستند .
به عنوان مثال:

```
<?php
class A {
    function A() {
        print('Style Constructor');
    }
}
?>
```

خروجی :

Deprecated: Methods with the same name as their class will not be constructors

in a future version of PHP; A has a deprecated constructor in...

متد های (Static calls to non-static) :

متد های Static calls to non-static منسوخ شده اند و ممکن است که در آینده حذف بشوند .

```
<?php
class A {
    function b() {
        print('Non-static call');
    }
}
A::b();
?>
```

خروجی :

Deprecated: Non-static method A::b() should not be called statically in...

Non-static call

گزینه password_hash () :

گزینه salt برای عملکرد password_hash () ، منسوخ شده است ؛ خوب پس توسعه دهنده ها salt های نامن خود را تولید نکنند ؛ عملکرد ، خود یک salt رمزنگاری شده امن را تولید می کند ، وقتی که هیچ salt ای توسط توسعه دهنده ، ارائه نشده باشد ؛ بدین ترتیب salt های سفارشی ، دیگر لازم نیستند .

انتخاب زمینه SSLcapture_session_meta

انتخاب زمینه capture_session_meta ، منسوخ شده است . حالا ، SSL metadata از طریق عملکرد stream_get_meta_data () استفاده می شود .

ورژنهای php

آموزش های نسخه ۷ PHP

PHP نسخه ۷ ، بیشترین انتظارها را داشته و منتشر کننده ویژگی های عمده ی برنامه نویسی PHP است . نسخه ۷ PHP ، در ۳ دسامبر ۲۰۱۵ منتشر یافته است . این آموزش ، ویژگی های جدید در نسخه ۷ و استفاده ی آن ها رابه شیوه ای آسان و راحت به شما یاد میدهد .

حضار

این آموزش برای توسعه دهندگان PHP آماده شده است . بعد از کامل کردن این آموزش شما به یک سطح متوسط از علم PHP می رسید که می توانید به سطح بعدی برسید .

پیش نیازها

ما فرض می کنیم که شما همه چیز را در رابطه با نسخه های قبلی PHP میدانید و الان شما می توانید ویژگی های جدید نسخه ۷ PHP یاد بگیرید .

```
<html>
<head>
<title> PHP-7 </title>
</head>
<body>
<?php
echo "<h1>Hello, PHP-7!</h1>";
?>
</body>
</html>
```

کلاس های anonymous

حالا کلاس های بی نام می توانند با استفاده از کلاس های جدید ، تعریف شوند . کلاس های بی نام می توانند در مکانی که پر از تعاریف کلاس است ، استفاده شوند .

به عنوان مثال :

```
<?php
interface Logger {
    public function log(string $msg);
}
class Application {
    private $logger;
    public function getLogger(): Logger {
        return $this->logger;
    }
    public function setLogger(Logger $logger) {
        $this->logger = $logger;
    }
}
```

```
$app = new Application;
$app->setLogger(new class implements Logger {
    public function log(string $msg) {
        print($msg);
    }
});
$app->getLogger()->log("My first Log Message");
?>
```

خروجی :

My first Log Message

Null Coalesce

در نسخه ۷ php ، یک قابلیت جدید به نام ادغام کننده null تولید شده است ؛ که ادغام کننده سه تایی را در پیوستگی با عملکرد isset () جایگزین می کند .
ادغام کننده null ، عملوند اول اگر وجود داشته باشد و پوچ نباشد ، خود را (Return) می کند ؛ ؛ در غیر این صورت ، عملوند دوم خود را (Return) می کند .

مثال :

```
<?php
// fetch the value of $_GET['user'] and returns 'not passed'
// if username is not passed
$username = $_GET['username'] ?? 'not passed';
print($username);
print("<br/>");

// Equivalent code using ternary operator
$username = isset($_GET['username']) ? $_GET['username'] : 'not
passed';
print($username);
print("<br/>");
// Chaining ?? operation
$username = $_GET['username'] ?? $_POST['username'] ?? 'not passed';
print($username);
?>
```


not passed
not passed
not passed

خروجی :

intdiv

(php7) یک تابع جدید به نام intdiv را دارا می باشد ، که تقسیم اعداد صحیحی که در عملوند است ، را انجام می دهد و خارج قسمت را به عدد صحیح بر می گرداند .

به عنوان مثال:

```
<?php
$value = intdiv(10,3);
var_dump($value);
print(" ");
print($value);
?>
```

خروجی :

```
int(3)
  ۳
```

IntlChar

در نسخه ۷ php ، کلاس جدیدی به نام IntlChar اضافه شده که کاری می کند تا ICU های اضافی را جستجو می کند . این کلاس ، یک عدد از متد های ایستا و ثابت تعریف می کند ، تا بتواند در دستکاری کاراکتر های UNICODE ، استفاده شود . شما باید افزونه Intl را قبلا نصب داشته باشید ، تا بتوانید از این کلاس استفاده کنید .

به عنوان مثال :

```
<?php
printf'%x', IntlChar::CODEPOINT_MAX;
print IntlChar::charName'@';
printIntlChar::ispunct'!';
?>
```

خروجی :

```
۱۰ffff
COMMERCIAL AT
true
```

php7 extension های حذف شده
extension های پایین از (php7) حذف شده اند :

ereg

mssql

mysql

sybase_ct

(SAPI) های پایین ، از (php7) حذف شده اند :

aolserver

apache

apache_hooks

apache2filter

caudium

continuity

isapi

milter

nsapi

phttpd

pi3web

roxen

thttpd

tux

webjames

Expectation

expectation ها در assert های قدیمی هم پشتیبانی می شوند .

expectation برای zero-cost assertions در کد تولید ، اجازه نشان دادن expectation

سفارشی را وقتی که assertion ناموفق باشد را می دهد .

وقتی پارامتر اول یک اصطلاح باشد ، assert ، زبان ساختنی ای ، است که برای این که به یک

string یا boolean تبدیل شود ، تست می شود .

پیکربندی دستورات برای assert :

دستوردهنده : zend.assertation - مقدار : ۱ - حالت توسعه دهنده ، تولید و اجرای کد

دستوردهنده : zend.assertation - مقدار : ۰ - تولید کد

دستوردهنده : zend.assertation – مقدار : ۱ – حالت ساخت ، کد را تولید نمی کند
 دستوردهنده : assert.exception – مقدار : ۱ – زمانی نشان داده می شود که assertion
 ناموفق باشد ، یا با فرستادن اشیائی که به عنوان Exception ، ارائه شده اند یا با فرستادن
 شیء AssertionError ، اگر Exception ارائه نشده بود .
 دستوردهنده : assert.exception – مقدار : ۰ – به عنوان چیزی که بالا توضیح داده شد ،
 تولید یا استفاده می شود . اما به جای فرستادن آن ، فقط یک هشدار بر پایه همان شیء تولید
 می کند .

(سازگاری با رفتار php5)

پارامترها :

assertion – در php5 ، این باید یک رشته باشد ، تا ارزیابی شود یا یک boolean تا تست
 شود . در php7 ، این می تواند هر اصطلاحی باشد که یک مقدار را برمی گرداند ؛ که اجرا می
 شود و در نتیجه ، برای اینکه استفاده شود ، نشان می دهد که assertion موفق بوده یا خیر .
 description یک توضیح انتخابی است که اگر موفق نبود ، با پیام همراه می شود .
 exception در نسخه php ۷ ، دومین پارامتر است که می تواند یک شیء Throwable به جای
 رشته توصیفی باشد ؛ که در این صورت این یک شیء است که اگر assertion ناموفق بود و
 پیکربندی assert.exeption فعال بود ، می تواند فرستاده شود .

مقدارهای بازگشتی:

اگر assertion ناموفق بود ، False ؛ در غیر این صورت True .

به عنوان مثال:

```
<?php
ini_set('assert.exception', 1);
class CustomError extends AssertionError {}
assertfalse, new CustomError('Custom Error Message!');
?>
```

خروجی : Fatal error: Uncaught CustomError: Custom Error Message! in...

CSPRNG

در نسخه ۷ (php) ، دو عملکرد دنبال شده ، تولید شده اند تا به صورت امن ، اعداد صحیح و
 رشته ها را در یک راه پلتفرم متقابل رمز نگاری کند .

random_bytes () – رمز نگاری امن pseudo-random bytes را تولید می کند .

random_int () – رمز نگاری امن pseudo-random integers را تولید می کند .

random_bytes()

یک طول دلخواه از رشته ها که از bytes های تصادفی ، درست شده ، که برای رمزنگاری مناسب است ، را تولید می کند ؛ مثل زمانی که کلید ها و وکتور را مقداردهی می کنید .
نحو اجرا :

```
string random_bytes ( int $length )
```

پارامترها :

(length) – طول رشته های تصادفی که باید به byte برگردند .

مقدار های برگشتی :

یک رشته که عدد درخواست شده از byte های تصادفی امن رمزنگاری شده است را برمی گرداند .

Exception ها و Error ها :

اگر یک سورس مناسب ، از تصادفی بودن پیدا نشد ، یک Exception نشان داده می شود .

اگر پارامتر Invalid داده شود ، یک (TypeError) نشان داده می شود .

به عنوان مثال :

```
<?php
$bytes = random_bytes(5);
print(bin2hex($bytes));
?>
۵۴cc305593
```

خروجی :

random_int()

اعداد صحیح رمزنگاری شده که برای استفاده در زمان مناسبی به کار می روند .

```
int random_int ( int $min , int $max )
```

نحوه اجرا :

پارامترها :

(min) – کمترین مقداری که برمی گردد ، که باید (php_int_min) یا بیشتر باشد .

(max) – بیشترین مقداری که برمی گردد ، که باید کمتر یا مساوی (php_int_max) باشد .

مقدار های برگشتی :

یک secure random integer رمزنگاری شده در محدوده ای که شامل کمترین تا بیشترین است ، را برمی گرداند .

Exception ها و Error ها :

اگر یک سورس مناسب از تصادفی بودن ، پیدا نشد ، یک Exception نشان داده می شود .

اگر پارامتر بی اعتباری داده شود ، یک (TypeError) نشان داده می شود .

اگر (max) کمتر از (min) بود ، یک خطا نشان داده می شود .

```
<?php
print(random_int(100, 999));
print("");
print(random_int(-1000, 0));
?>
```

به عنوان مثال :

خروجی :

۶۱۴

۸۸۲-

متد Closure::call()

متد Closure::call() ، به عنوان یک راه مختصر نویسی اضافه شده تا موقتا دامنه شیء را گرفته و بسته کنیم و بعد آن را فراخوانی کنیم . این کارایی در مقایسه با (bindto) در نسخه ۵/۶ (php) خیلی سریع تر است .

مثال برای نسخه های قبل از (php 7) :

```
<?php
class A {
    private $x = 1;
}
// Define a closure Pre PHP 7 code
$value = function() {
    return $this->x;
};
// Bind a closure
$value = $value->bindTo(new A, 'A');
print($value());
?>
```

خروجی : ۱

```
<?php
class A {
    private $x = 1;
}
// PHP 7+ code, Define
$value = function() {
    return $this->x;
};
print($value->call(new A));
?>
```

خروجی : ۱

گزینه های Session

در (php7) به بالا ، عملکرد `session_start()` آپشن های زیادی را می توانید انتخاب کنید تا دستوراتی که در پیکربندی `php.ini` وجود دارد را ، ویرایش کنید . این آپشن ها از `session.lazy_write` هم پشتیبانی می کند ، که اگر به صورت پیش فرض ، داده های فایل تغییر کند ، مجبور به ویرایش فایل `session` می شوید .

آپشن دیگری که اضافه شده `read_and_close` نام دارد ، که نشان می دهد که داده های `session` باید خوانده و بعد از آن ، بدون آن که تغییر کنند ، بسته شوند . برای مثال ، `session.cache_limiter` را به صورت شخصی کرده ، تا `session` ، بعد از خوانده شدن ، با استفاده از کدی که در پایین است ، بسته شود .

```
<?php
session_start([
    'cache_limiter' => 'private',
    'read_and_close' => true,
]);
?>
```

تفاوت ها

نسخه ۷ (PHP) چیست ؟

(PHP 7) یک عمده منتشر شده از برنامه نویسی (PHP) است و همین طور تبلیغ شده تا یک انقلاب را در راه برنامه های وب به دنبال داشته باشد . `php7` ، توسعه یافته است و برای تلفن همراه به شرکت ها و در نهایت (Cloud) تحویل داده می شود . این انتشار مهمترین تغییر برای (PHP) ، بعد از انتشار نسخه ۵ (PHP) در سال ۲۰۰۴ است .

امکانات جدید

ده ها ویژگی که به (PHP ۷) اضافه شدند که قابل توجه ترین آن ها در پایین ذکر شده اند: بهبود کارایی : دارای ادغام کد (PHPNG) در (PHP 7) ، این ویژگی دو برابر سریع تر از (PHP 5) است .

مصرف کمتر حافظه : (PHP 7) بهینه شده ، از منبع کمتری بهره میگیرد .

اعلان های نوع عددی : حالا ، پارامترها و انواع برگشتی میتوانند ، اجرا شوند .

پشتیبانی (۶۴-bit) : پشتیبانی برای ماشین هایی که برپایه (۶۴-bit) ساخته شده باشند .

بهبود سلسله مراتب Exception : سلسله مراتب Exception بهبود یافته است .

خیلی از FatalError ها تبدیل به Exception شده اند : محدوده ی Exception ها بیشتر

شده است و خیلی از FatalError هایی که به Exception تبدیل می شوند ، را پوشش می

دهد .

تولید کننده Random numbers های امن : اضافه شدن تولید کننده Random numbers های امن ، (Secure API) .

(SAPIs) منسوخ شده و پسوند ها حذف شده اند : برنامه های مختلف قدیمی و (SAPIs) و برنامه های افزودنی ، از نسخه آخر حذف شده اند .

اپراتور یکی شدن (the null coalescing operator) : اپراتور جدید یکی شدن ، اضافه شده است .

اعلامیه های عددی : پشتیبانی برای نوع بازگشتی و نوع پارامتری اضافه شده است .

کلاس های پنهان : پشتیبانی برای نوع پنهانی کلاس ، اضافه شد .

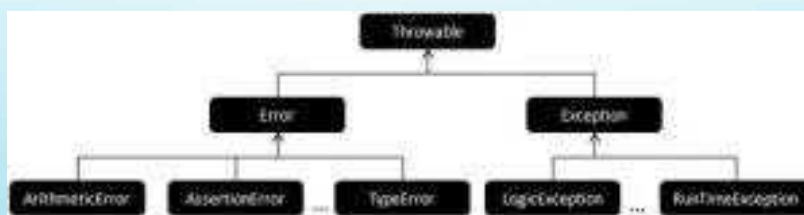
Zero cost asserts : پشتیبانی برای Zero cost asserts اضافه شد .

PHP 7 از موتور zend 3.0 استفاده می کند تا کارایی برنامه را دو برابر و ۵۰% مصرف حافظه بهتر از PHP 5.6 کند. این ویژگی اجازه میدهد تا کاربران همزمان بیشتری ، بدون لازم داشتن سخت افزار اضافی استفاده کنند . PHP 7 با توجه به حجم کار امروزی بازنویسی و طراحی شده است

گروه بندی use ها 2

در نسخه ۷ (php) ، رفع خطا و گزارش ، تغییر کرده است . به جای گزارش خطا ها به روش سنتی که در نسخه ۵ (php) استفاده می شد ، حالا ، بیشتر خطا ها با استفاده از فرستادن exception خطا ، رفع می شوند . شبیه این ErrorException ، به اولین بلوکی که تطبیق داده شد ، نشان داده می شوند . اگر هیچ تطبیقی داده نشد ، یک defaultException با set_exception_handler () صدا زده می شود . در این مورد هیچ defaultException ای ، وجود ندارد ، به همین دلیل استثنا تبدیل به FatalError شده و شبیه یک خطای سنتی رفع می شود .

از زمانی که سلسله مراتب از Exception تمديد نمی شوند ، کدی که از Exception \$e { ... } استفاده می کند ، بلاک می شود تا Exception های گرفته نشده ، رفع شوند در نسخه ۵ (php) این شکل خطا ها رفع نمی شوند . یک Error \$e { ... } بلاک می کند و یا یک رفع کننده set_exception_handler () ، مورد نیاز است تا FatalError را رفع کند .





به عنوان مثال :

```

<?php
class MathOperations {
    protected $n = 10;

    // Try to get the Division by Zero error object and display as Exception
    public function doOperation(): string {
        try {
            $value = $this->n % 0;
            return $value;
        } catch (DivisionByZeroError $e) {
            return $e->getMessage();
        }
    }
}

$mathOperationsObj = new MathOperations();
print($mathOperationsObj->doOperation());
?>

```

خروجی : Modulo by zero

Scalar type

در نسخه ۷ (php) ، یک قابلیت جدید به نام اعلان های نوع عددی (Scalar Type Declarations) ، معرفی شده است . اعلان های نوع عددی دو گزینه دارند : اجباری – حالت پیش فرض است و نیازی به مشخص کردن نیست . سخت گیرانه – باید با صراحت تمام اشاره شود . انواع داده شده برای پارامتر های عملکرد می تواند با حالت های بالا ، اجرا شود .

int

float

bool

string

interfaces

array

callable

مثال برای حالت اجباری :

```
<?php
// Coercive mode
function sum(int ...$ints) {
    return array_sum($ints);
}
print(sum(2, '3', 4.1));
?>
```

خروجی : ۹

مثال برای حالت سخت گیرانه :

```
<?php
// Strict mode
declare(strict_types=1);
function sum(int ...$ints) {
    return array_sum($ints);
}
print(sum(2, '3', 4.1));
?>
```

spaceship

در نسخه ۷ php ، یک قابلیت جدید به نام اپراتور spaceship تولید شده است . این قابلیت برای مقایسه دو اصطلاح ، آمده است . وقتی که اولین اصطلاح به ترتیب کمتر ، مساوی ، یا بیشتر از دومین حالت باشد ، این قابلیت ، مقدار های -۱ ، ۰ ، ۱ برگشت می دهد .

به عنوان مثال :

```
<?php
//integer comparison
print( 1 <=> 1);print("<br/>");
print( 1 <=> 2);print("<br/>");
print( 2 <=> 1);print("<br/>");
print("<br/>");
//float comparison
print( 1.5 <=> 1.5);print("<br/>");
print( 1.5 <=> 2.5);print("<br/>");
print( 2.5 <=> 1.5);print("<br/>");
print("<br/>");
//string comparison
print( "a" <=> "a");print("<br/>");
print( "a" <=> "b");print("<br/>");
print( "b" <=> "a");print("<br/>");
?>
```

unserialize

نسخه ۷ php تابع جدید Filtered unserialize معرفی کرده ، تا وقتی که شیء ها را unserialize می کند در داده های نامطمئن ، امنیت بهتری را ارائه دهد . از کد های تزریقی جلوگیری کرده و توسعه دهنده را قادر می سازد ، تا کلاس هایی که می توانند unserialize شوند را ، اصطلاحاً whitelist یا اعتماد سازی کند .

به عنوان مثال :

```
<?php
class MyClass1 {
    public $obj1prop;
}
class MyClass2 {
    public $obj2prop;
}

$obj1 = new MyClass1();
$obj1->obj1prop = 1;
$obj2 = new MyClass2();
$obj2->obj2prop = 2;

$serializedObj1 = serialize($obj1);
$serializedObj2 = serialize($obj2);
```



```
// default behaviour that accepts all classes
// second argument can be omitted.
// if allowed_classes is passed as false, unserialize converts all objects into
__PHP_Incomplete_Class object
$data = unserialize($serializedObj1 , ["allowed_classes" => true]);

// converts all objects into __PHP_Incomplete_Class object except those of
MyClass1 and MyClass2
$data2 = unserialize($serializedObj2 , ["allowed_classes" => ["MyClass1",
"MyClass2"]]);

print($data->obj1prop);
print("<br/>");
print($data2->obj2prop);
?>
```

خروجی : ۱
۲

داده بازگشتی

در نسخه ۷ (php) یک قابلیت جدید به نام اعلامیه های نوع بازگشتی تولید شده است .
اعلامیه های نوع بازگشتی ، نوع مقداری که یک عملکرد باید بازگرداند را مشخص می کند .
انواع داده های زیر میتوانند نوع بازگشتی معتبر باشند .

int
float
bool
string
interfaces
array
callable

مثال برای نوع بازگشتی معتبر :

```
<?php
declare(strict_types = 1);
function returnIntValue(int $value): int {
    return $value;
}
print(returnIntValue(5));
?>
```

خروجی : ۵

مثال برای نوع بازگشتی نامعتبر :

```
<?php
declare(strict_types = 1);
function returnIntValue(int $value): int {
    return $value + 1.0;
}
print(returnIntValue(5));
?>
```

خروجی :

Fatal error: Uncaught TypeError: Return value of returnIntValue() must be of ...the type integer, float returned

کارایی

همانطور که تیم zend ، عکس های زیر در رابطه با مقایسه کارایی PHP 7 در برابر PHP 5.6 و HHVM 3.7 در برنامه های مبتنی بر PHP محبوب را نشان میدهد .

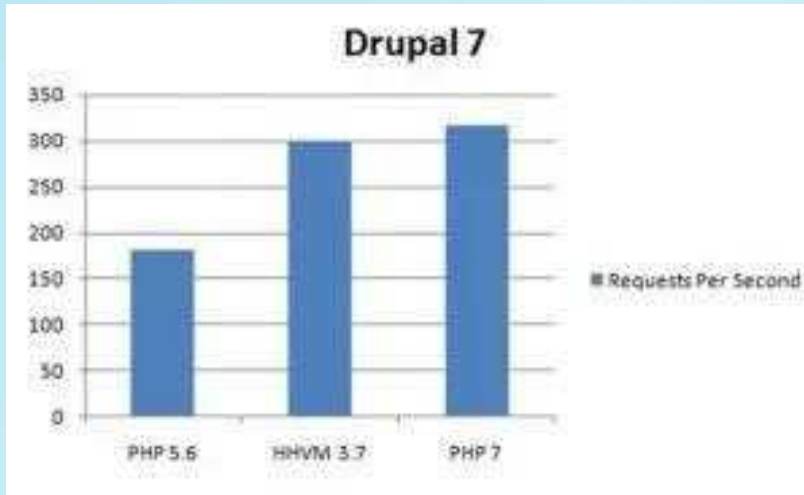
Magento 1.9

وقتی که تراکنش magento اجرا می شود ، و php7 به دلیل ۲ برابر سریع تر بودن از PHP 5.6 خودش را ثابت می کند .



Drupal 7

وقتی که تراکنش Drupal اجرا می شود ، و php7 به دلیل ۲ برابر سریع تر بودن از PHP 5.6 خودش را ثابت می کند .

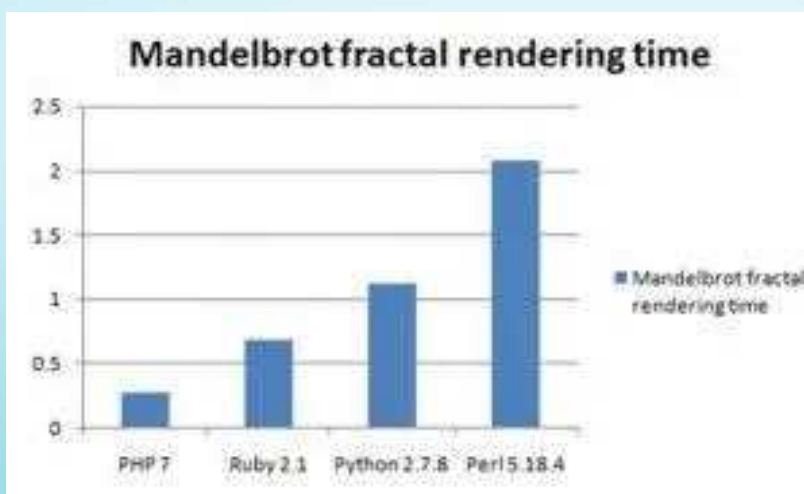


WordPress 3.6

وقتی که تراکنش WordPress اجرا می شود ، و php7 به دلیل ۲ برابر سریع تر بودن از PHP 5.6 خودش را ثابت می کند .



مقایسه چند زبان پویا



php7-کارایی

گروه بندی use ها

در هر صورت در (php7) ، یک statement می تواند در وارد کردن هر کلاس ، هر عملکرد و هر ثابت که فضای نام مثل هم دارند ، به جای استفاده از چندین statement استفاده شود .

به عنوان مثال :

آرایه های ثابت

آرایه های ثابت می توانند با استفاده از تابع define () تعریف شوند .

در php 5.6 ، آن ها فقط می توانستند با استفاده از کلیدواژه های توابع ، تعریف شوند .

به عنوان مثال :

```
<?php
//define a array using define function
define('animals', [
    'dog',
    'cat',
    'bird'
]);
print(animals[1]);
?>
```

<?php

خروجی : cat

// Before PHP 7

```
use com\tutorialspoint\ClassA;
use com\tutorialspoint\ClassB;
use com\tutorialspoint\ClassC as C;
use function com\tutorialspoint\fn_a;
use function com\tutorialspoint\fn_b;
use function com\tutorialspoint\fn_c;
use const com\tutorialspoint\ConstA;
use const com\tutorialspoint\ConstB;
use const com\tutorialspoint\ConstC;
```

// PHP 7+ code

```
use com\tutorialspoint\{ClassA, ClassB, ClassC as C};
use function com\tutorialspoint\{fn_a, fn_b, fn_c};
use const com\tutorialspoint\{ConstA, ConstB, ConstC};
```

?>