

## مالجديد في الاصدار الثامن من الـ PHP



PHP 8.0 هو تحديث رئيسي للغة PHP.

يحتوي على العديد من الميزات والتحسينات الجديدة بما في ذلك named arguments union types ، والسماط، constructor property promotion، و عبار match ، وعامل nullsafe ، و JIT ، والتحسينات في نظام النوع ، ومعالجة الأخطاء ، والاتساق.

أداء أفضل ، بناء جملة أفضل ، أمان محسّن.

### الفهرس:

1. Named arguments
2. Attributes
3. Constructor property promotion
4. Union types
5. Match expression
6. Nullsafe operator
7. Saner string to number comparisons
8. Consistent type errors for internal functions
9. Just-In-Time compilation

## Named arguments

1. حدد المعلمات المطلوبة فقط ، مع تخطي المعلمات الاختيارية.
2. المدخلات مستقلة عن النظام وموثقة ذاتياً.

مثال:

```
<?php
    htmlspecialchars($string, double_encode: false);
?>
```

## Attributes

بدلاً من تعليقات PHPDoc التوضيحية ، يمكنك الآن استخدام البيانات الوصفية المنظمة مع البنية الأصلية لـ PHP.

مثال:

```
<?php
class PostsController
{
    #[Route("/api/posts/{id}", methods: ["GET"])]
    public function get($id) { /* ... */ }
}
?>
```

## Constructor property promotion

رمز معياري boilerplate أقل لتحديد الخصائص وتهيئتها.

مثال:

```
<?php
class Point {
    public function __construct(
        public float $x = 0.0,
        public float $y = 0.0,
        public float $z = 0.0,
    ) {}
}
?>
```

## Union types

بدلاً من تعليقات PHPDoc التوضيحية لمجموعة من الأنواع ، يمكنك استخدام تعريفات نوع الاتحاد الأصلي Union types التي يتم التحقق من صحتها في وقت التشغيل.

مثال:

```
<?php
class Number {
    public function __construct(
        private int|float $number
    ) {}
}

new Number('NaN'); // TypeError
?>
```

## Match expression

خاصية المطابقة الجديدة match مشابهة للتبديل switch وتحتوي على الميزات التالية:

1. دالة match هي عبارة، مما يعني أنه يمكن تخزين نتائجها في متغير أو إرجاعها.
2. تدعم فروع المطابقة فقط التعبيرات أحادية السطر ولا تحتاج إلى عبارة break.
3. تقوم "المطابقة" بإجراء مقارنات صلومة.

مثال:

```
<?php
echo match (8.0) {
    '8.0' => "Oh no!",
    8.0 => "This is what I expected",
};
//> This is what I expected
?>
```

## Nullsafe operator

بدلاً من شروط الاختيار الفارغة null check ، يمكنك الآن استخدام سلسلة من المكالمات مع عامل nullsafe الجديد.

عندما يفشل تقييم عنصر واحد في السلسلة ، يتم إحباط تنفيذ السلسلة بأكملها ويتم تقييم السلسلة بأكملها إلى null.

مثال:

```
<?php
$country = $session?->user?->getAddress()?->country;
?>
```

## Saner string to number comparisons

عند المقارنة بسلسلة رقمية ، تستخدم PHP 8 مقارنة رقمية. خلاف ذلك ، فإنه يحول الرقم إلى سلسلة ويستخدم مقارنة سلسلة.

مثال:

```
<?php
$check = 0 == 'foobar' // false
?>
```

## Consistent type errors for internal function

ترمي معظم الوظائف الداخلية الآن استثناء خطأ إذا فشل التحقق من صحة المعلمات.

مثال:

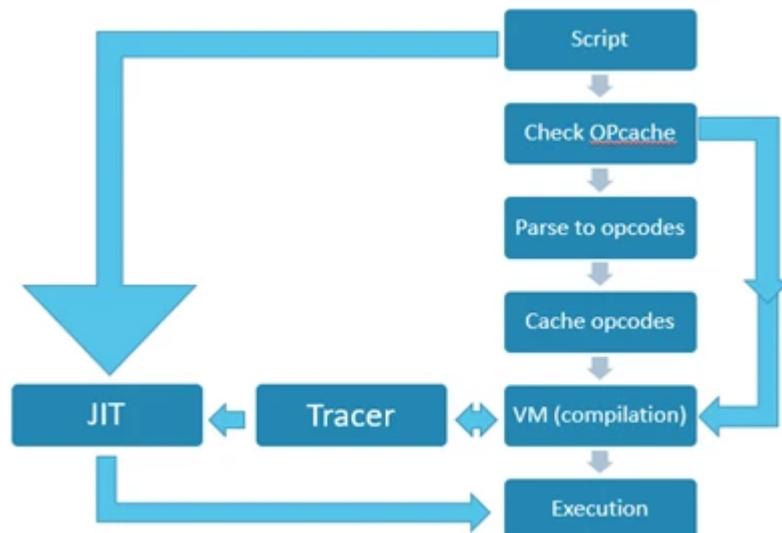
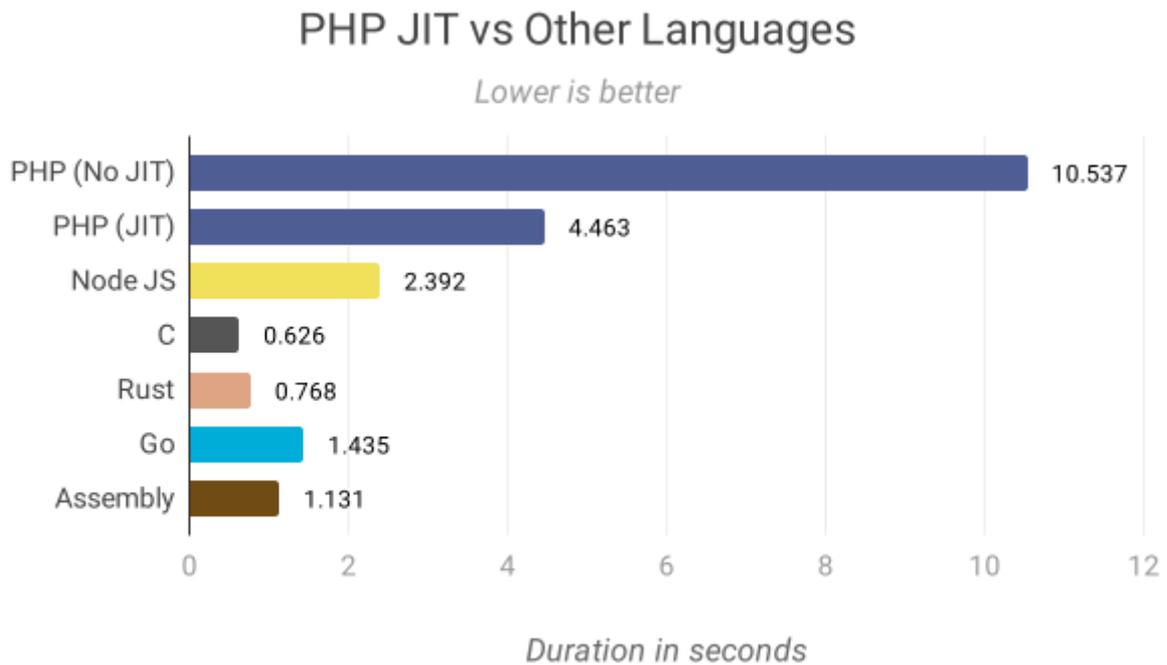
```
<?php
strlen([]); // TypeError: strlen(): Argument #1
($str) must be of type string, array given

array_chunk([], -1); // ValueError: array_chunk():
Argument #2 ($length) must be greater than 0
?>
```

# Just-In-Time compilation

يقدم PHP 8 محركي ترجمة JIT.

يُظهر Tracing JIT ، أكثر المحركات الواعدة ، أداءً أفضل بثلاث مرات تقريبًا على المعايير التركيبية وتحسين 1.5 إلى مرتين في بعض التطبيقات المحددة طويلة المدى.



## تغيرات أخرى

تحسينات لـ Type System ومعالجة الأخطاء

1. Stricter type checks for arithmetic/bitwise operators
2. Abstract trait method validation
3. Correct signatures of magic methods
4. Reclassified engine warnings
5. Fatal error for incompatible method signatures
6. The @ operator no longer silences fatal errors.
7. Inheritance with private methods
8. Mixed type
9. Static return type
10. Types for internal functions Email thread
11. Opaque objects instead of resources for Curl, Gd, Sockets, OpenSSL, XMLWriter, and XML extensions

## تعديلات وتحسينات أخرى في بناء الجملة

1. Allow a trailing comma in parameter lists RFC and closure use lists
2. Non-capturing catches
3. Variable Syntax Tweaks
4. Treat namespaced names as single token
5. Throw is now an expression
6. Allow ::class on objects

## فئات وواجهات ووظائف جديدة

1. Weak Map class
2. Stringable interface
3. str\_contains(), str\_starts\_with(), str\_ends\_with()
4. fdiv()
5. get\_debug\_type()
6. get\_resource\_id()
7. token\_get\_all() object implementation
8. New DOM Traversal and Manipulation APIs