



البحث

ما الفرق بين php8 والنسخ الذي قبلها من وماهي الإضافات التي تم اضافتها على php8

المهندسة/ عبير يوسف الحامري الدكتور/ إبراهيم الشامي

لغة php8 تم إصدارها رسمياً في ٢٦ نوفمبر ٢٠٢٠ وجاءت مع تحسينات كبيرة ومميزات جديدة مقارنة بالإصدارات السابقة فيما يلي شرح للفرق بين php8 والإصدارات السابقة بالإضافة الى المميزات والخصائص التي تمت اضافتها:

الفرق بين php8 والإصدارات السابقة: -

١- تحسين الأداء

- PHP8 يعتمد على JIT Compiler (Just-In-Time Compiler)، وهو تحسين جذري في طريقة تنفيذ الكود. حيث يعمل JIT على ترجمة الكود مباشرة إلى لغة الآلة أثناء التشغيل، مما يؤدي إلى تحسين الأداء بشكل كبير، خاصة في التطبيقات التي تعتمد على العمليات الحسابية المعقدة.

٢- رفع الكفاءة

- تم تحسين محرك Zend Engine (محرك PHP) ليصبح أكثر كفاءة في التعامل مع الكود،

مما يجعل PHP 8 أسرع وأكثر استقراراً.

٣- التوافقية مع الإصدارات القديمة

- PHp8 أزال العديد من الميزات القديمة (Deprecated Features) التي كانت موجودة في

الإصدارات السابقة، مما يعني أن بعض الأكواد القديمة قد تحتاج إلى تعديلات للعمل مع PHP 8.

أبرز الإضافات الجديدة في PHP 8: -

1- JIT Compiler (Just-In-Time Compiler)

- يقوم JIT بتحسين الأداء عن طريق تجميع الكود أثناء التشغيل بدلاً من تفسيره فقط. يتم استخدامه لتحسين الأداء بشكل خاص في التطبيقات التي تحتاج إلى حسابات ثقيلة أو معقدة.

2- Union Types (أنواع البيانات المتعددة)

- تسمح بتمرير أكثر من نوع بيانات للمتغير أو الدالة. على سبيل المثال:

```
function foo (int|float $value): int|float {  
    return $value * 2  
}
```

في المثال أعلاه، يمكن للدالة أن تقبل إما `int` أو `float` كمدخلات.

3- Named Arguments (المعاملات المسماة)

- يمكن تحديد المعاملات بالاسم عند استدعاء الدالة، مما يسهل قراءة الكود وتجنب الأخطاء عند وجود قيم افتراضية.

```
function greet ($name, $greeting = "Hello")  
{  
  echo "$greeting, $name!";  
  greet (name: "John", greeting: "Hi");  
}
```

4- Attributes(السمات)

- أداة جديدة لإضافة بيانات وصفية (Metadata) إلى الكود مثل الكلاسات والوظائف. يمكن استخدامها لتبسيط العمل مع المكتبات والأطر.

[Example Attribute] #

```
{ class My Class  
  // محتوى الكلاس  
}
```

5- Match Expression(switch - بديل)

- تحسين لهيكل switch التقليدي، أكثر بساطة وفعالية.

```
status = match($code) {  
  200=>'OK',  
  404=>'Not Found',  
}
```

```
500=>'Internal Server Error',  
default => 'Unknown Status',  
};
```

- يتميز بعدم الحاجة إلى كتابة break لكل حالة.

6- Null safe Operator(<?>)

- تبسيط التعامل مع القيم التي يمكن أن تكون null بدلاً من كتابة شروط متعددة، يمكنك استخدام هذا المشغل.

```
$country=
```

```
$user? ->get Address ()? ->get Country ();
```

إذا كانت أي من القيم null لن يتم تنفيذ الكود وسيُرجع null بدلاً من حدوث خطأ.

7-ConstructorPropertyPromotion

- تبسيط تعريف الخصائص وإنشائها مباشرة في مُنشئ الكلاس.

```

class User {
public function __construct (
private string $name,
private int $age
){}
}

```

بدلاً من كتابة الخصائص ثم تعيينها يدوياً، يتم تعريفها مباشرة في المنشئ.

8- Weak Maps (الخرائط الضعيفة)

- أداة لتحسين إدارة الذاكرة. تسمح باستخدام الكائنات كمفاتيح بدون منع حذفها من الذاكرة عند عدم الحاجة إليها.

```

$map = new Weak Map()
$object = new stdClass ();
$map[$object] = 'value';

```

9- (Error Handling) مُحسّنات الأخطاء

- تم تحسين طريقة التعامل مع الأخطاء، مثل رسائل الأخطاء الواضحة عند استدعاء دوال غير موجودة أو التعامل مع كائنات غير صالحة.

10- Stringable Interface

- واجهة جديدة تُستخدم مع الكائنات التي يمكن تحويلها إلى سلسلة نصية.

```
class My Class{  
public function __to String ():  
    string {  
return "This is my class ";  
}  
}  
$ object = new My Class ();  
echo $object;// يطبع :This is my class
```

١١. مميزات إضافية:

دالة جديدة للتقسيم الآمن، تدعم القسمة على () fdiv-
الصفء.

للتحقق مما إذا كانت سلسلة نصية -str_contains
تحتوي على نص معين.

- str_starts_with()/str_ends_with(): للتحقق
إذا كانت سلسلة نصية تبدأ أو تنتهي بنص معين.

```
str_contains('Hello World', 'World');//  
true
```

```
str_starts_with('Hello World', 'Hello');//  
true
```

```
str_ends_with('Hello World', 'World'); //  
true
```

مقارنة الأداء بين PHP 7 و PHP 8: -

- مع إضافة JIT وتحسينات المحرك الأساسي، فإن PHP 8 أسرع بنسبة تتراوح بين ١٠٪ إلى ٢٥٪ مقارنة بـ PHP 7.4، خاصة في العمليات المعقدة.

الخلاصة

PHP 8 تمثل خطوة كبيرة إلى الأمام في تطوير اللغة، حيث تركز على تحسين الأداء، تسهيل كتابة الكود، وتقليل الأخطاء. إذا كنت تعمل على مشروع جديد أو تريد تحسين مشروعك الحالي، فإن الترقية إلى ستوفر لك العديد من المزايا. ومع ذلك، يجب التأكد من توافق الكود الحالي مع الميزات الجديدة.