



الجامعة الوطنية

أهم الفروقات ما بين PHP 8 والإصدارات السابقة

من حيث

- ❖ التحسينات.
- ❖ الميزات الجديدة التي لم تكن موجودة في الإصدارات السابقة.
- ❖ بالإضافة إلى تحسينات في الأداء والأمان.
- ❖ إلغاء بعض الميزات.
- ❖ دعم البيانات القوية وتحسينات في الكائنات.
- ❖ تحسينات أخرى في PHP 8.
- ❖ الترقية من الإصدارات السابقة من PHP 7.x إلى PHP 8.

سنستعرض أهم الفروقات والميزات الجديدة بين PHP 8 والإصدارات التي سبقتها (مثل PHP 7.4 و PHP 7.x).

١. تحسينات في الأداء:

- JIT (Just In Time) Compiler :
 - واحدة من أهم الإضافات في PHP 8 هي مترجم JIT . هذا التحديث يُحسن أداء PHP بشكل كبير في بعض الحالات، خصوصًا في العمليات الحسابية المعقدة. ولكن يجب ملاحظة أن تحسينات الأداء قد تكون ملحوظة أكثر في التطبيقات ذات الأحمال الثقيلة والتي تتطلب عمليات رياضية معقدة. بالنسبة لمعظم التطبيقات العادية، التحسين قد لا يكون ملحوظًا بشكل كبير.
 - PHP 7.4 كان يحتوي على بعض التحسينات أيضًا في الأداء، ولكن ليس بمستوى JIT في PHP 8.

٢. ميزات جديدة في PHP 8:

- Annotations أو Attributes :
 - تم تقديم Attributes في PHP 8 ، وهي ميزة تتيح للمطورين إضافة بيانات وصفية (Metadata) إلى الكود باستخدام الـ Annotations بشكل أكثر رسمية وقوة.
 - قبل PHP 8 ، كان يتم استخدام تعليقات DocBlock أو مكتبات خارجية مثل Doctrine Annotations لتحقيق نفس الهدف، لكن Attributes توفر طريقة أكثر كفاءة وأداءً.

• Union Types :

- يدعم PHP 8 الآن Union Types ، مما يعني أنه يمكن للمتغير أن يكون من نوعين أو أكثر. على سبيل المثال:

```
function foo (int|string $value): void {  
    // $value يمكن أن يكون إما عدد صحيح أو سلسلة  
}
```

• Named Arguments :

- يمكن للمطورين الآن تمرير المعاملات باستخدام أسماء المتغيرات بدلاً من الاعتماد على الترتيب فقط.

```
function test ($name, $age) {  
    echo "$name is $age years old";  
}  
  
// يتيح لك تحديد المعاملات باستخدام الاسم  
test (age: 25, name: 'John');  
function test ($name, $age) {  
    echo "$name is $age years old";  
}
```

- **: Match Expression**

- قدمت PHP 8 تعبير `match` وهو بديل أكثر قوة ومرونة لـ `switch`:

```
$result = match($status) {  
    'success' => 'Operation succeeded',  
    'failure' => 'Operation failed',  
    default => 'Unknown status',  
};
```

- الفائدة الرئيسية من `match` هي أنها أكثر أمانًا حيث تقوم بمقارنة القيم دون الحاجة إلى النوع المطبوع، وتدعم القيم الافتراضية (`default`) بشكل أفضل.

- **: Constructor Property Promotion**

- هذه الميزة تسهل كتابة الكود عندما يتم تمرير الخصائص إلى المُنشئ (`constructor`). قبل PHP 8، كان عليك كتابة الكود التالي:


```
class Person {  
  
    private string $name;  
  
    private int $age;  
  
    public function __construct (string $name, int $age) {  
  
        $this->name = $name;  
  
        $this->age = $age;  
  
    }  
}
```

- في PHP 8 ، يمكنك اختصار الكود إلى:

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

• Weak Maps :

- تم إضافة Weak Maps في PHP 8 ، وهي أداة يمكنك من تخزين العناصر في خريطة ولكن دون التأثير على جمع القمامة (garbage collection). هذا يعني أن الكائنات في الخريطة سيتم جمعها تلقائيًا عندما لا يكون لها مراجع.

٣. تحسينات في الأمان:

• Sanitized Input Functions :

- تم تحسين بعض الدوال الخاصة بالمدخلات مثل (`filter_var()`) لضمان تصفية المدخلات بشكل أفضل، مما يعزز الأمان بشكل عام.

• Error Handling:

- في PHP 8 ، تم تحسين التعامل مع الأخطاء واستثناءات الأخطاء، مما يجعل الكود أكثر استقرارًا وأمانًا.

٤. إلغاء بعض الميزات:

- **Deprecated Features** الميزات المهملة.
- تم إلغاء بعض الوظائف والميزات القديمة في PHP 8 مثل.
- **real type** تم إلغاء النوع **real** واستخدام **float** بدلاً منه.
- **create_function ()** تم إلغاء هذه الوظيفة لأنها كانت تحتوي على مشاكل أمنية.
- **each ()** تم إلغاء هذه الوظيفة لأنها كانت قديمة وغير فعالة.
- **mbstring.func_overload** تم إلغاء هذه الخاصية.
- **JSON_THROW_ON_ERROR** :
- تم إلغاء التعديل الذي يسمح بإرجاع أخطاء **JSON** في الإصدارات السابقة.

٥. دعم البيانات القوية وتحسينات في الكائنات:

- **Improved Type System:**
 - تم تحسين النظام النوعي في PHP 8 ليشمل المزيد من أنواع البيانات الثابتة والمرنة، مثل **Static Return Types** و **Union Types**.
- **Static Variables in Functions:**
 - تم تحسين التعامل مع المتغيرات الثابتة داخل الدوال بحيث يمكن استخدامها بطريقة أكثر مرونة.

٦. تحسينات أخرى في PHP 8 :

- **Nullsafe Operator (?->):**
 - تم إضافة **Nullsafe Operator** ، وهو يسمح بالوصول إلى خصائص أو استدعاء دوال على الكائنات التي قد تكون **null** دون إلقاء استثناء:

```
$user?->getName();
```


إذا كان `$user` هو `null` ، فسيتم إرجاع `null` بدلاً من إلقاء استثناء.

- `str_contains()`, `str_starts_with()`, `str_ends_with()`
- تم إضافة هذه الدوال لتسهيل عمليات البحث عن السلاسل النصية، وهي لا تتطلب الآن استخدام `strpos()` أو `substr()` المعقدة.
- `str_contains()` تحقق إذا كانت السلسلة تحتوي على نص معين.
- `str_starts_with()` تحقق إذا كانت السلسلة تبدأ بنص معين.
- `str_ends_with()` تحقق إذا كانت السلسلة تنتهي بنص معين.

٧. الترقية من الإصدارات السابقة من PHP 7.x إلى PHP 8:

- التوافق العكسي: يجب على المطورين الانتباه إلى أن هناك بعض التغييرات في سلوك بعض الوظائف في PHP 8 ، مما يعني أن الكود الذي كان يعمل على PHP 7 قد لا يعمل بالطريقة نفسها على PHP 8 .
- دعم الإصدارات القديمة: إذا كنت تستخدم PHP 7.4 أو أقدم، عليك تحديث بعض الأكواد لتكون متوافقة مع PHP 8 ، خاصةً تلك التي تستخدم وظائف مهمة أو تم إلغاؤها.

❖ الخلاصة:

PHP 8 قدم العديد من التحسينات في الأداء والأمان، وأضاف ميزات جديدة مثل JIT و Union Types و Attributes وغيرها من الميزات القوية. كما تم إزالة بعض الوظائف القديمة التي كانت تسبب مشاكل في الأداء أو الأمان.