تعتبر PHP 8 إصدارًا رئيسيًا جديدًا ميما يلي أهم الفروقات والميزات الجديدة في PHP الف@8:

الفرق بين لغة php8 والاصدارات القديمة؟

1. أداء محسّن مع JIT (Just-In-Time Compilation)

ميزة جديدة: ميزة JIT تعني أن الكود يتم ترجمته مباشرة إلى لغة الآلة وقت التشغيل، مما يحسن الأداء في بعض العمليات الحسابية المعقدة.

ملاحظة: التأثير الأكبر يظهر في التطبيقات المكثفة مثل معالجة البيانات الضخمة أو العمليات الحسابية، وليس بالضرورة في تطبيقات الويب التقليدية.

1. أنواع بيانات محسّنة (Union Types) :

الميزة: يمكنك الآن تحديد أكثر من نوع واحد للمتغير أو الدالة.

مثال:

Function foo(int|float $param) {

; Return $param

}

هذا يسمح للمتغير بقبول قيم من نوع int أو float.

1. الإضافة Nullsafe Operator :

الميزة: تسهل التعامل مع القيم التي قد تكون فارغة (null) دون الحاجة إلى كتابة شروط إضافية.

مثال:

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

إذا كانت أي من القيم null، لن يحدث خطأ وسيتم إرجاع null بدلًا من ذلك.

1. تحسينات على Match Expression:

الميزة: بديل أكثر كفاءة وأنيق لـ switch مع دعم الإرجاع الفوري.

مثال:

$result = match($input) {

1 => 'One',

2 => 'Two',

Default => 'Other',

};

1. السمات (Attributes):

الميزة: طريقة جديدة لتحديد البيانات الوصفية (Metadata) على الكود باستخدام سمات تشبه التعليقات التوضيحية.

مثال:

#[Route('/path', methods: ['GET'])]

Function myFunction() {

// ...}

1. تحسينات في النظام اللغوي

Constructor Property Promotion: اختصار لتعريف المتغيرات في الكلاس.

Class User {

Public function \_\_construct(

Public string $name,

Private int $age

) {}

}

هذا يعفي من كتابة التعريف الكامل للمتغيرات.

Named Arguments: تمرير المعاملات (Arguments) للدوال بالاسم بدل الترتيب.

myFunction(param1: 'value1', param2: 'value2');

1. تحسينات على الدوال القياسية

دوال جديدة مثل:

Str\_contains(): للتحقق مما إذا كانت سلسلة نصية تحتوي على نص معين.

Str\_starts\_with(): للتحقق إذا كانت سلسلة تبدأ بمقطع معين.

Str\_ends\_with(): للتحقق إذا كانت سلسلة تنتهي بمقطع معين.

1. تحسينات على معالجة الأخطاء

تم تحسين رسائل الخطأ لتكون أكثر وضوحًا.

إضافة استثناءات لعمليات الـ Engine بدلاً من الأخطاء التقليدية.

1. تحسينات على أنواع البيانات والقيم الافتراضية

دعم استخدام Static Return Type.

دعم أنواع البيانات في المتغيرات الافتراضية للبارامترات.

الفرق في الأداء

PHP 8 أسرع من PHP 7.x بنسبة قد تصل إلى 10-20% في بعض السيناريوهات، خصوصًا بفضل JIT.

الإضافات الجديدة مع أمثلة:

1. دعم الأنواع الصارمة (Type Declarations)

في PHP 7 وما بعدها، يمكنك تحديد نوع المدخلات والمخرجات للدوال.

Function addNumbers(int $a, int $b): int {

Return $a + $b;

}

Echo addNumbers(5, 10); // 15

1. المشغل Null Coalescing (??)

يستخدم لتوفير قيمة افتراضية إذا كانت المتغيرات غير معرفة أو فارغة.

$name = $\_GET['name'] ?? 'Guest';

Echo $name; // إذا لم يتم تحديد "name" في URL، تطبع "Guest".

1. التعبيرات السهمية (Arrow Functions)

طريقة مختصرة لتعريف الدوال المجهولة.

$numbers = [1, 2, 3];

$squared = array\_map(fn($n) => $n \* $n, $numbers);

Print\_r($squared); // [1, 4, 9]

1. أصناف Enum (في PHP 8.1)

تُستخدم لتعريف مجموعات ثابتة من القيم.

Enum Status {

Case Active;

Case Inactive;

}

$status = Status::Active;

1. السمات (Attributes)

تُستخدم لتوفير بيانات وصفية (Metadata) للكيانات البرمجية.

#[Route('/home')]

Function home() {

Echo "Welcome to Home!";

}

1. دعم Union Types (في PHP 8.0)

للسماح بتحديد أكثر من نوع للمتغير أو الإرجاع.

Function printData(int|string $data) {

Echo $data;

}

printData("Hello"); // مسموح

printData(123); // مسموح

1. تحسينات القوالب النصية (JIT – Just In Time Compilation)

تحسين أداء البرامج الثقيلة (PHP 8.0 وما بعده).

**الدكتور: ابراهيم الشامي. عمل الطالب: وليد راوح شمسان.**

**المادة;php التخصص: تقنية معلومات.**