**الفرق بين نسخة PHP 8 وما قبلها من النسخ**

في **PHP 8** ، أدخلت تحسينات كبيرة في الأداء والعديد من الميزات الجديدة التي تجعل كتابة الكود أكثر كفاءة ووضوحًا.

الفرق بين **PHP 8** والإصدارات السابقة (مثل PHP 7) يكمن في عدة جوانب تشمل الأداء، الميزات الجديدة، وتسهيل كتابة الكود.

مقارنة بين النسخة 8 وما قبلها من النسخ :

**.1 الأداء (Performance)**

* **PHP 7**
* قدمت PHP 7 تحسينات كبيرة في الأداء مقارنة بـ PHP 5 بفضل محرك **Zend Engine 3.0**.
* تشغيل الكود أسرع بمرتين مقارنة بـ PHP 5.
* **PHP 8**
* أضافت PHP 8 **محرك JIT (Just In Time Compilation)**، الذي يحسّن الأداء في العمليات الحسابية والبرمجيات الثقيلة.
* **JIT** يقوم بتحويل الكود إلى لغة الآلة أثناء التشغيل بدلاً من تفسيره.

**2. Union Types (الأنواع المشتركة)**

* **قبل PHP 8**
* لا يمكنك تحديد أكثر من نوع واحد للمتغير أو المعامل. كان يتم استخدام التعليقات التوضيحية فقط.

// PHP 7

function displayNumber($number) {

if (is\_int($number) || is\_float($number)) {

echo $number;

}

}

* **PHP 8**
* يمكنك تحديد نوع مشترك (Union Type) بسهولة.

function displayNumber(int|float $number): void {

echo $number;

}

**3. Named Arguments (المعاملات المسماة)**

* **قبل PHP 8**
* يجب تمرير القيم للدوال بترتيب محدد.

// PHP 7

function greet($name, $greeting = "Hello") {

echo "$greeting, $name!";

}

greet("Ali", "Hi"); // النتيجة: Hi, Ali!

* **PHP 8**
* يمكنك تمرير القيم بالاسم دون الالتزام بالترتيب.

greet(name: "Ali", greeting: "Hi");

greet(greeting: "Hi", name: "Ali");

**4. Nullsafe Operator**

* **قبل PHP 8**
* كنت تحتاج للتحقق من أن كل قيمة ليست **null**.

// PHP 7

$country = null;

if ($session && $session->user && $session->user->getAddress()) {

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

}

* **PHP 8**
* استخدام **nullsafe operator (?->)** يجعل الكود أبسط.

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

**5. Match Expression**

* **قبل PHP 8**
* كان يتم استخدام **switch** لتحديد القيم.

// PHP 7

switch ($status) {

case 200:

$message = "OK";

break;

case 404:

$message = "Not Found";

break;

default:

$message = "Unknown";

}

* **PHP 8**
* **match** هو بديل أكثر اختصارًا وفعالية.

$message = match ($status) {

200 => "OK",

404 => "Not Found",

default => "Unknown",

};

**6. Constructor Property Promotion**

* **قبل PHP 8**
* تحتاج لتعريف الخصائص وتمريرها في البناء.

// PHP 7

class User {

public $name;

public $age;

public function\_construct($name, $age){

$this->name = $name;

$this->age = $age;}}

* **PHP 8**
* يمكنك تعريف وتمرير الخصائص مباشرة في البناء.

class User {

public function \_\_construct(public string $name, public int $age) {}}

**.7 الدوال النصية الجديدة**

* **PHP 8**

أضافت دوال نصية جديدة:

* **str\_contains()**: تتحقق إذا كان النص يحتوي على جزء معين.
* **str\_starts\_with()**: تتحقق إذا كان النص يبدأ بجزء معين.
* **str\_ends\_with()**: تتحقق إذا كان النص ينتهي بجزء معين.

echo str\_contains("Hello World", "World"); // Yes

echo str\_starts\_with("Hello World", "Hello"); // Yes

echo str\_ends\_with("Hello World", "World"); // Yes

**.8 السمات (Attributes)**

* **قبل PHP 8**
* يتم استخدام التعليقات التوضيحية للإشارة إلى الميتا بيانات.

// PHP 7

/\*\*

\* @Route("/user") \*/

function getUser() {}

* **PHP 8**
* أُضيفت **Attributes** كبديل حديث.

#[Route("/user")]

function getUser() {}

**9 . تحسين إدارة الذاكرة باستخدام WeakMap**

* **WeakMap** تتيح تتبع الكائنات دون إبقائها في الذاكرة بعد حذفها.

$map = new WeakMap();

$obj = new stdClass();

$map[$obj] = "value";

unset($obj); // سيتم حذف الكائن من WeakMap تلقائيًا.

**10 .تحسينات في Type System**

* دعم النوع **mixed**: يمكن أن يكون المتغير أي نوع بيانات.
* دعم النوع **static** كنوع إرجاع.

**11 . دعم الفواصل في القوائم (Trailing Comma)**

* أصبح من الممكن إضافة فاصلة أخيرة في المعاملات والمصفوفات.

$array = [

"one",

"two",

"three",

];

هديل خالد عبدالله الحزوره

علوم حاسوب