



الجمهورية اليمنية
الجامعة الوطنية
كلية العلوم والهندسة



اعداد: وئام فيصل الاصبحي
تقنية معلومات.
مستوى ثالث.

إشراف الدكتور: إبراهيم الشامي.

الـ PHP : هي لغة برمجة تُستخدم بشكل رئيسي لتطوير تطبيقات الويب. لقد مرّت PHP بالعديد من التحديثات على مر السنين، وكل نسخة جديدة تقدم ميزات وتحسينات جديدة.

الفرق بين PHP 8 والنسخ التي قبلها:

تعتبر PHP 8 واحدة من أهم التحديثات التي أُضيفت في لغة البرمجة، مع تحسينات كبيرة على الأداء، التعامل مع الأخطاء، وتحسينات في الكتابة البرمجية. إليك مقارنة بين PHP 8 وبعض النسخ السابقة:

١. تحسينات في الأداء:

PHP 7 قدم تحسينات كبيرة في الأداء مقارنة بالنسخ السابقة (مثل PHP 5)، بفضل محرك Zend Engine 3.0، مما جعلها أسرع بمقدار الضعف في بعض الحالات.

PHP 8 أضاف (Just In Time Compiler) JIT
لتحسين الأداء أكثر، خاصةً في العمليات الحسابية
الثقيلة. ١

٢. إضافات جديدة في PHP 8:

أ. (Just-In-Time Compiler) JIT:

PHP 8 يضيف محرك JIT، الذي يحسن الأداء في
بعض السيناريوهات التي تعتمد على الحسابات
الرياضية المعقدة أو العمليات الثقيلة.

مثال على استخدام JIT:

```
php?>
```

```
// تم تفعيل JIT لتحسين الأداء في العمليات الحسابية  
الثقيلة.
```

```
} function fibonacci($n)  
    {  
        if ($n <= 1)  
            return $n;  
        return fibonacci($n - 1) +  
            fibonacci($n - 2);  
    }
```

echo fibonacci(30); // هذا المثال سيستفيد من JIT لتحسين السرعة في حسابات الأرقام الكبيرة
<?

ب. Attributes (أو Annotations):

تُتيح Attributes إضافة بيانات وصفية (مثل التعليقات) إلى الكود بطريقة أكثر هيكلية.

مثال:

```
[Route("/home")]#  
} class HomeController  
    {  
        public function index()  
        {  
            return "Welcome Home";  
        }  
    }
```

هذا يمكن أن يكون مفيداً لأطر العمل مثل Symfony لتحديد المسارات في التطبيق.

ج. Named Arguments:

في PHP 8، أصبح من الممكن استخدام المتغيرات المسماة بدلاً من ترتيب المعاملات.

مثال:

```
<?php>

function greet($name, $age)
return "Hello, $name! You are $age
    ;".years old
{

// ;echo greet(age: 25, name: "Alice")
استخدمنا "المتغيرات المسماة"
```

. Match Expressions :

PHP 8 يقدم Match Expressions التي تعتبر أكثر مرونة من Switch. تدعم المقارنات الدقيقة (strict comparisons).

مثال:

```
php?>
;value = 2$
// استخدام Match بدلاً من Switch
} echo match($value)
    , "One" <= ١
    , "Two" <= ٢
    , "Three" <= ٣
```

```
, "default" => "Unknown"
```

```
};
```

```
<?
```

هـ. Constructor Property Promotion:

PHP 8 يسمح بتعريف المتغيرات مباشرة داخل الباني (constructor)، مما يقلل من الكتابة.

مثال:

```
>?php
```

```
} class Person
```

```
)public function __construct
```

```
,public string $name
```

```
public int $age
```

```
{}
```

```
{
```

```
person = new Person(name: "Alice", $  
                        ;age: 30)  
echo $person->name; // Alice  
<?
```

و. Union Types:

Union Types يسمح بتحديد أكثر من نوع للمتغير.
يمكن أن يكون للمتغير أكثر من نوع، مثل
int|string أو array|object.

مثال:

```
php?>  
{ function printValue(int|string $value)  
    ;echo $value  
}
```



```
printValue(١٢٣); // يعمل  
printValue("Hello"); // يعمل  
<?
```

ز. مزيد من التحقق الصارم للأنواع:

PHP 8 قدم تحسناً في التحقق من الأنواع في اللغة،
مثل التأكد من أن أنواع البيانات تتطابق بشكل أكثر
دقة.

مثال:

```
>?php  
function addNumbers(int $a, int $b): int  
    {  
        ;return $a + $b  
    }
```

echo addNumbers(٥, ١٠); // يعمل بشكل صحيح

// echo addNumbers(٥, "١٠"); // ستؤدي إلى خطأ

ج. التعامل مع الأخطاء:

PHP 8 قدم تحسينات في التعامل مع الأخطاء، حيث أصبحت بعض الأخطاء أكثر وضوحًا ودقة، مثل TypeError و ValueError.

مثال:

```
php?>
function divide($a, $b)
{
    if ($b === 0)
        throw new ValueError("Cannot
                                ;divide by zero")
    ;return $a / $b
}
```

ValueError سترتفع // سترتفع (٠, ١) ; echo divide

٣. فوائد PHP 8:

تحسين الأداء: مع إضافة JIT، سيكون الأداء أفضل في العمليات الحسابية المكثفة.

كتابة أكثر وضوحًا: مثل Attributes و Named Arguments و Match Expressions، التي تجعل الكود أكثر وضوحًا وأسهل في الصيانة.

مرونة أكبر: مع Union Types و Constructor Property Promotion، يمكن كتابة كود مرن وقابل للتوسيع بسهولة أكبر.

تحسينات في التعامل مع الأخطاء: تم تحسين كيفية رفع الأخطاء والرسائل.

وبشكل مختصر:

PHP 8 قدم العديد من الميزات التي تعزز الأداء وتزيد من مرونة كتابة الكود.

JIT يساعد في تحسين الأداء خاصة في العمليات الحسابية.

Match Expressions و Named Arguments و Attributes تجعل الكود أكثر مرونة وسهولة في الكتابة والصيانة.

مع Union Types و Constructor Property Promotion، أصبحت الكتابة أسهل وأكثر فعالية.

تحديثات PHP 8 تمثل قفزة كبيرة مقارنة
بالإصدارات السابقة، خاصةً من حيث الأداء وتنظيم
الكود.