

# PHP Course

## Lesson 1



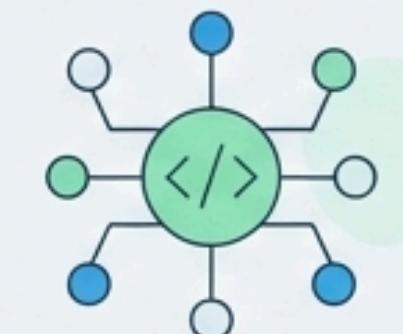
Created by Hossam Mahmoud

# خارطة الطريق (Roadmap)



**مفهوم برمجة الويب**  
(Web Programming Concepts)

[1]



**المحتوى динамички**  
(Dynamic Content)

[7]

**الفرق بين جانب العميل وجانب الخادم**  
(Client-side vs Server-side)

[2]

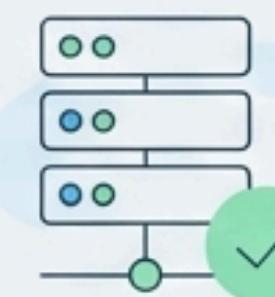
**المتغيرات وقواعد التسمية**  
(Variables & Naming)

[8]



**لماذا نستخدم MySQL و PHP ؟**

[3]

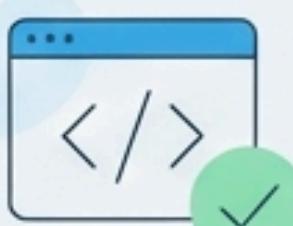


**تجهيز بيئة العمل**  
(XAMPP Installation)

[4]

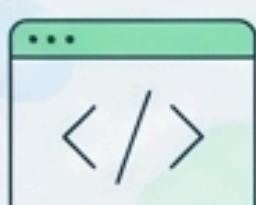
**أنواع البيانات**  
(Data Types)

[9]



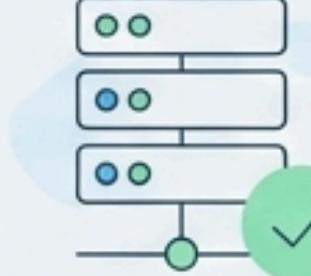
**البنية الأساسية للكود**  
(Basic Syntax)

[5]



**أوامر الطباعة والتعليقات**  
(Output & Comments)

[6]

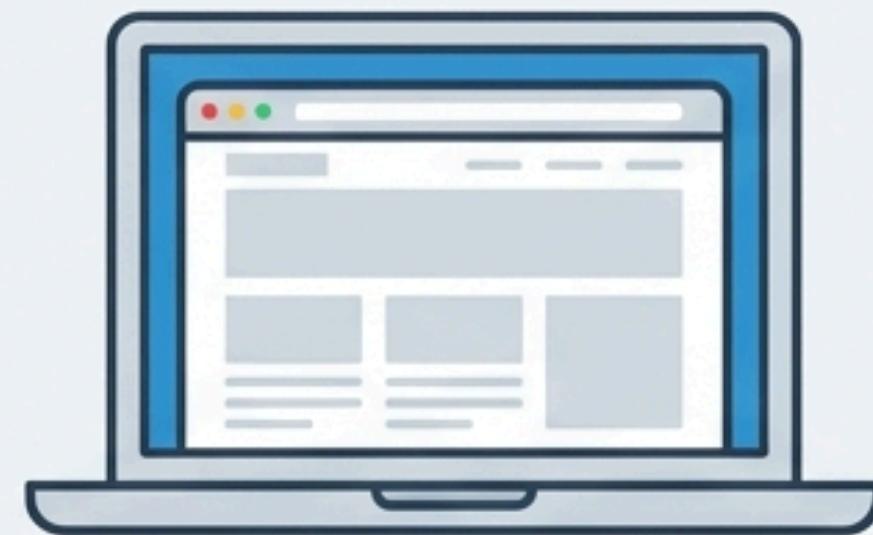


**الثوابت والمتغيرات المتقدمة**  
(Constants & Variable Variables)

[11]

# مفهوم برمجة الويب (Web Programming Architecture)

## Client Side (Front-end)



## Server Side (Back-end)



هو البرنامج الذي يعمل على متصفح المستخدم (Browser).

مسؤول عن العرض والواجهة (User Interface) وما يراه المستخدم.

Languages: HTML, CSS, JavaScript.

هو البرنامج الذي يعمل على السيرفر لتوليد المحتوى والتعامل مع البيانات.

الاستعلام من قواعد البيانات، التعامل مع الملفات، ومعالجة مدخلات المستخدم.

Languages: PHP, Python, ASP.Net, Java.

# الثنائي الأقوى: MySQL و PHP

## PHP (The Manager)

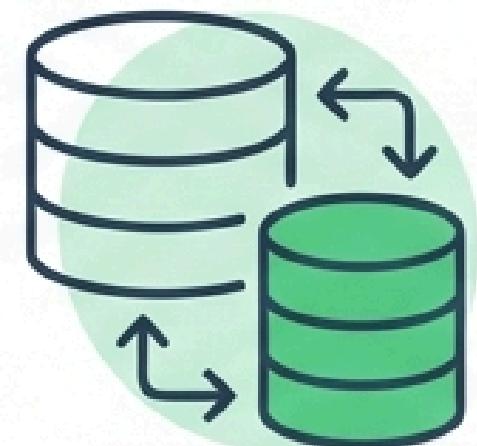
لغة برمجة (Scripting Language) تعمل على السيرفر.



مفتوحة المصدر (Open Source) جميع الأنظمة (Cross-platform). تقوم بتوليد المحتوى ديناميكيًّا وتعمل ك وسيط بين المستخدم وقاعدة البيانات.

## MySQL (The Warehouse)

نظام إدارة قواعد بيانات علاقية (RDBMS).

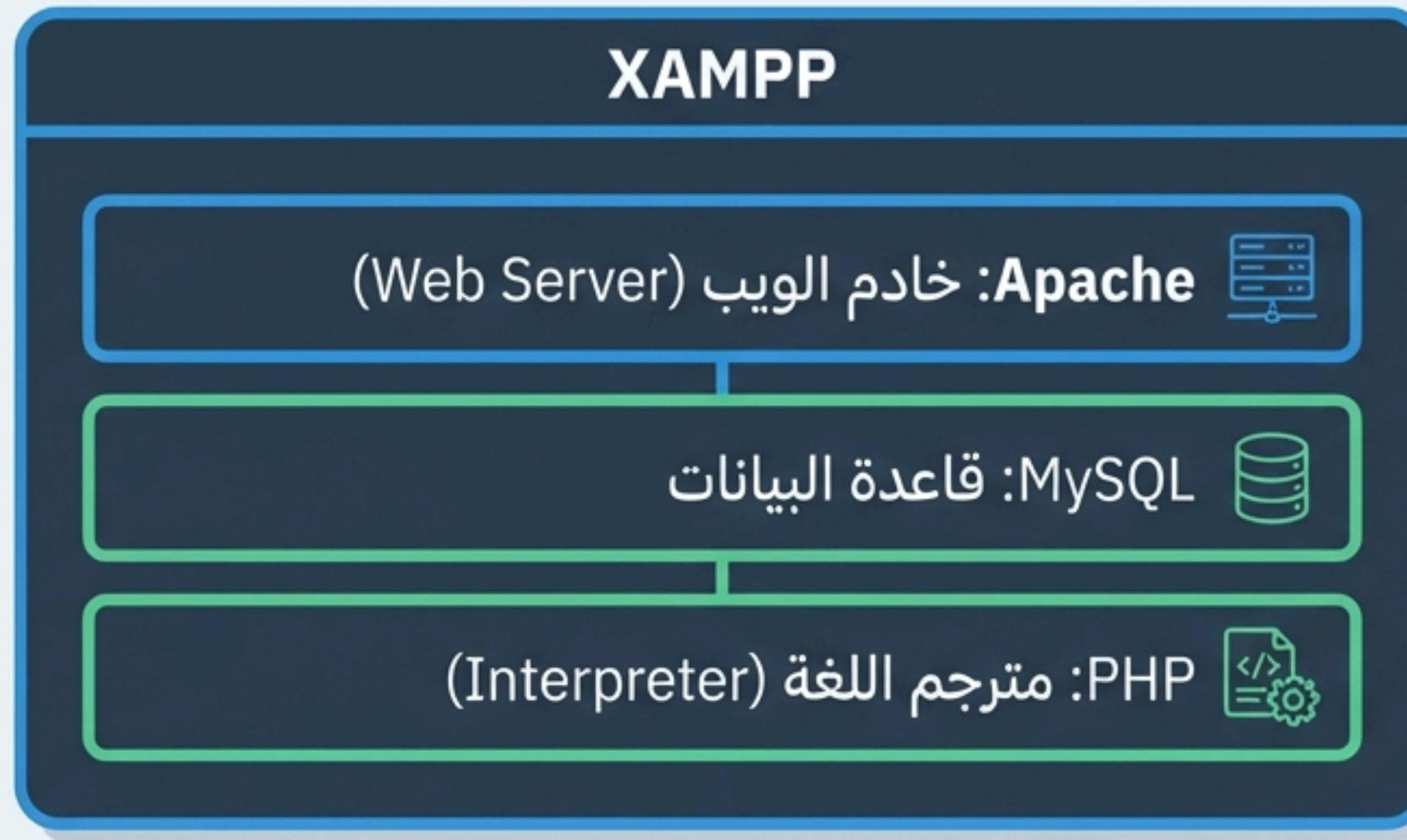


سريعة، قوية، ومجانية. تخزين البيانات (مثل المستخدمين والمنتجات) ليتم استرجاعها بواسطة استرجاعها بواسطة PHP.

العلاقة تكاملية: PHP تطلب البيانات و MySQL تخزنها وترسلها.



# تجهيز بيئة العمل (Setting up the Environment)



المتصفح (Browser) لا يفهم لغة PHP مباشرة. لكي يعمل الكود، نحتاج إلى "سيرفر" يقوم بترجمته.

الحل هو برنامج XAMPP Server

بمجرد تثبيت XAMPP، يتحول جهاز الكمبيوتر الشخصي إلى سيرفر محلي (localhost) قادر على تشغيل ملفات PHP.



# البنية الأساسية للكود (Basic Syntax)

## PHP Code

```
<!DOCTYPE html>
<html>
<body>

<?php
    // PHP code goes here
?>

</body>
</html>
```

١ يبدأ كود PHP بـ <?php وينتهي بـ ?.

٢ يمكن دمج كود PHP داخل ملفات HTML بسهولة.

٣ يتم تنفيذ الكود على السيرفر، ويرسل النتيجة ك'Plain HTML' للمتصفح.

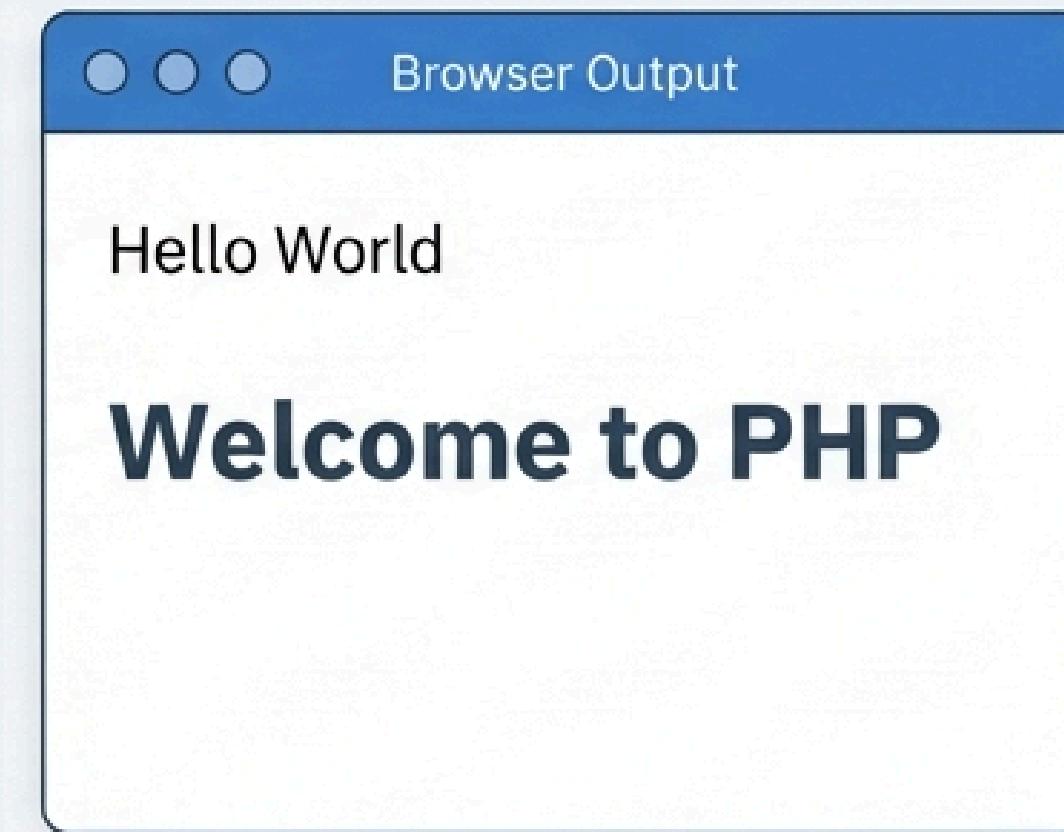
٤ هام جداً: امتداد الملف يجب أن يكون php (وليس .html).

# أمر الطباعة (Output Statement)

Dark Mode

```
<?php  
    echo "Hello World";  
    echo "<h1>Welcome to PHP</h1>";  
?>
```

**Don't forget!**



**echo**: هو الأمر المستخدم لطباعة النصوص أو كود HTML للمتصفح.

**Semicolon (;)**: كل سطر برمجي في PHP يجب أن ينتهي بفاصلة منقوطة. نسيانها هو الخطأ الأكثر شيوعاً.

**Quotes**: النصوص توضع داخل علامات تنسيص " ".

# الملاحظات والتعليقات (Comments)

أسطر لا ينفذها البرنامج، هدفها شرح الكود للمبرمج وتسهيل القراءة مستقبلاً.

PHP Code

```
<?php
```

```
// This is a single-line comment
```

```
# This is also a single-line comment
```

```
/*
```

```
    This is a multi-line comment block  
    that spans over multiple lines
```

```
*/
```

```
?>
```

سطر واحد (Single line)  
(JetBrains Mono)

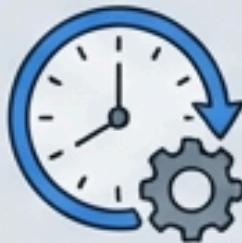
عدة أسطر (Multi-line)

# المحتوى الديناميكي (Dynamic Content)



## Static (HTML)

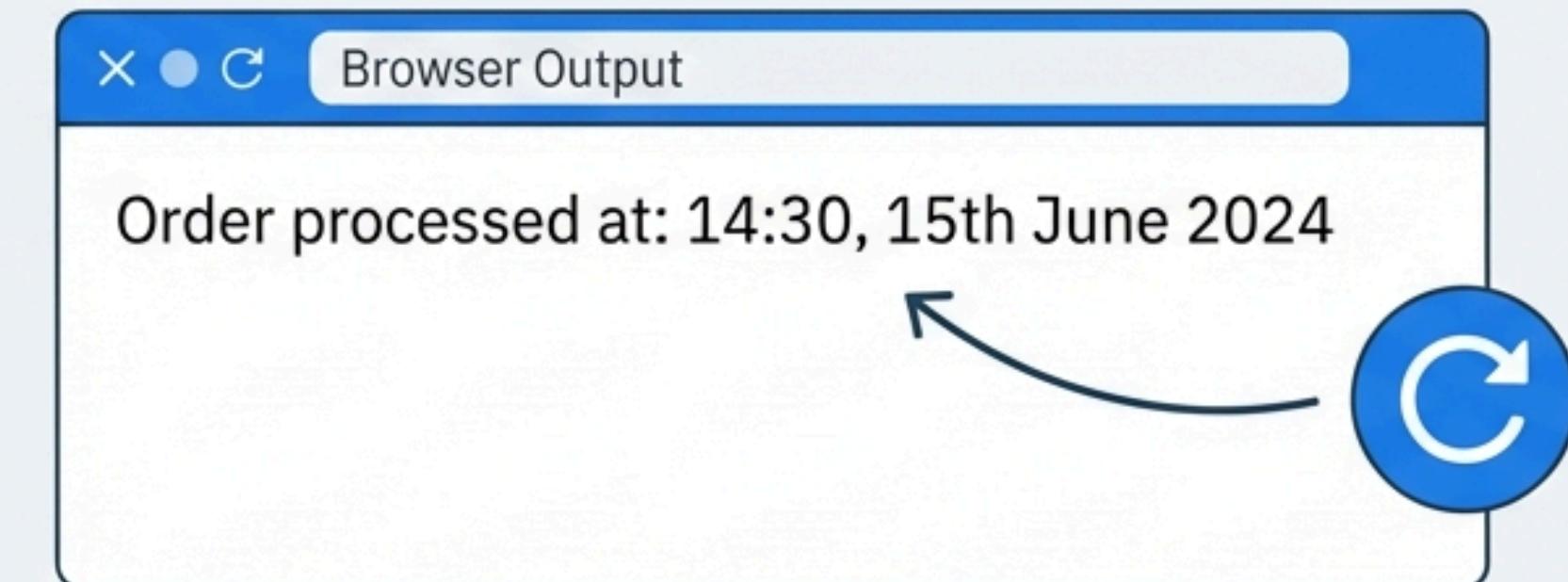
المحتوى ثابت لا يتغير إلا بتعديل الكود يدوياً.



## Dynamic (PHP)

المحتوى يتغير تلقائياً بناءً على الوقت، المستخدم، أو البيانات.

```
1 <?php  
2 echo "Order processed at: ";  
3 echo date("H:i, jS F Y");  
4 ?>
```



في كل مرة يقوم المستخدم بتحديث الصفحة (Refresh)، سيتغير الوقت تلقائياً.

# المتغيرات: قواعد التسمية (Variables & Identifiers)

المتغير هو "صندوق" في الذاكرة لتخزين البيانات، يبدأ دائمًا بعلامة الدولار \$.



يجب أن يبدأ بحرف أو شرطة سفلية -  
(مثال: \$var, الدولار).

\$var, \$\_name



لا يمكن أن يبدأ برقم (خطأ: .(\$1var :).

\$1var



يمكن أن يحتوي على حروف وأرقام و -.

\$User\_id123



حساسية الأحرف (Case Sensitive)  
.\$name يختلف تماماً عن \$Name

\$Name, \$name

# أنواع البيانات (Data Types)



**String** (نص)

```
$name = "Ahmed";
```



**Integer** (عدد صحيح)

```
$count = 5;
```



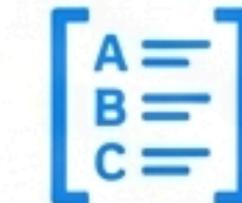
**Float** (عدد عشري)

```
$price = 10.5;
```



**Boolean** (منطقي)

```
$active = true;
```

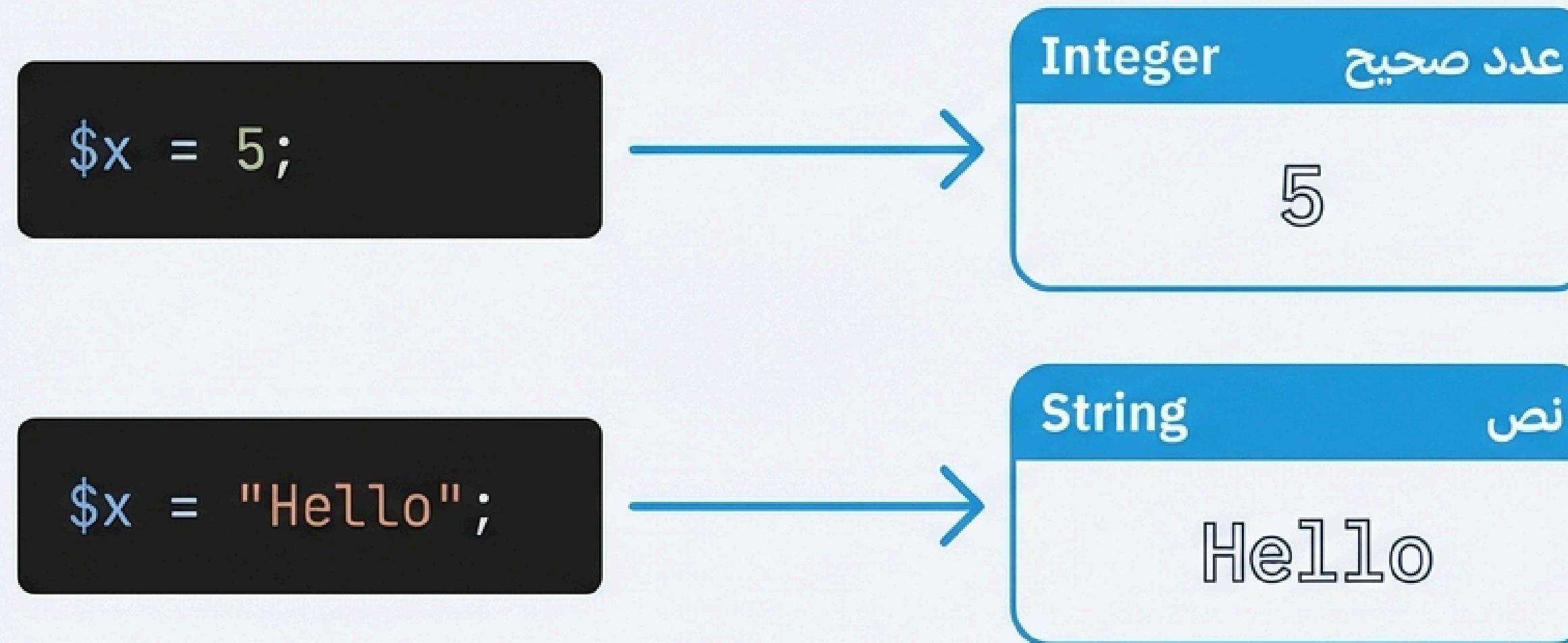


**Array** (مصفوفة)

```
$colors = array("Red", "Blue");
```

## المرونة في الأنواع (Dynamic Typing)

لغة PHP هي Loosely Typed Language. لا يلزم تعريف نوع المتغير مسبقاً، فهو يتحدد تلقائياً بناءً على القيمة.



نوع المتغير يمكن أن يتغير أثناء عمل البرنامج، مما يعطي مرونة كبيرة للمطور.

# تحويل الأنواع (Type Casting)

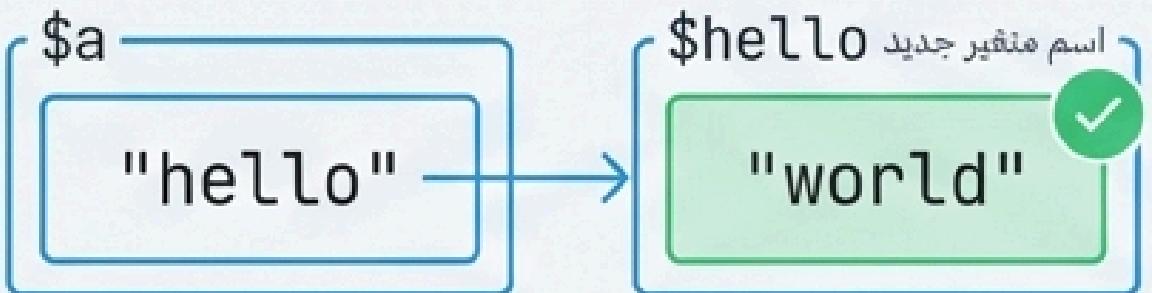
إجبار المتغير على التحول لنوع آخر مؤقتاً باستخدام الأقواس (type).

```
$total_qty = "10";  
Casting Operator → $int_qty = (int)$total_qty; ← Becomes Integer
```

المتغير الأصلي يبقى كما هو (String)، بينما المتغير الجديد يحتوي على القيمة الرقمية.

# مفاهيم متقدمة: المتغيرات والثوابت

## Variable Variables (\$\$)



تم إنشاء متغير جديد باسم 'hello' وقيمته .'world'

```
$a = "hello";  
$$a = "world";
```

استخدام قيمة متغير كاسم لمتغير آخر.

اسم الثابت مكتوب بأحرف كبيرة (Uppercase) للتاكيد على العرف (Convention).

```
define("TAX_RATE", 0.15);
```

TAX\_RATE

0.15

القيمة ثابتة ولا يمكن تغييرها لاحقاً

## (الثوابت) Constants

قيم يتم تعريفها مرة واحدة ولا يمكن تغييرها. لا نستخدم علامة \$ معها.

# ملخص المحاضرة (Lecture Summary)

- العقل المدبر ومخزن البيانات للموقع الديناميكية.
- ضوري لتحويل جهازك إلى سيرفر محلي.
- ال코드 يبدأ ب `?php` وينتهي ب `;`.
- تبدأ ب `$` وهي حساسة لحالة الأحرف.