Ministry of Higher Education and Scientific Research
Southern Technical University
Technological Institute of Basra
Department of Computer Networks
and Software Techniques

Learning package

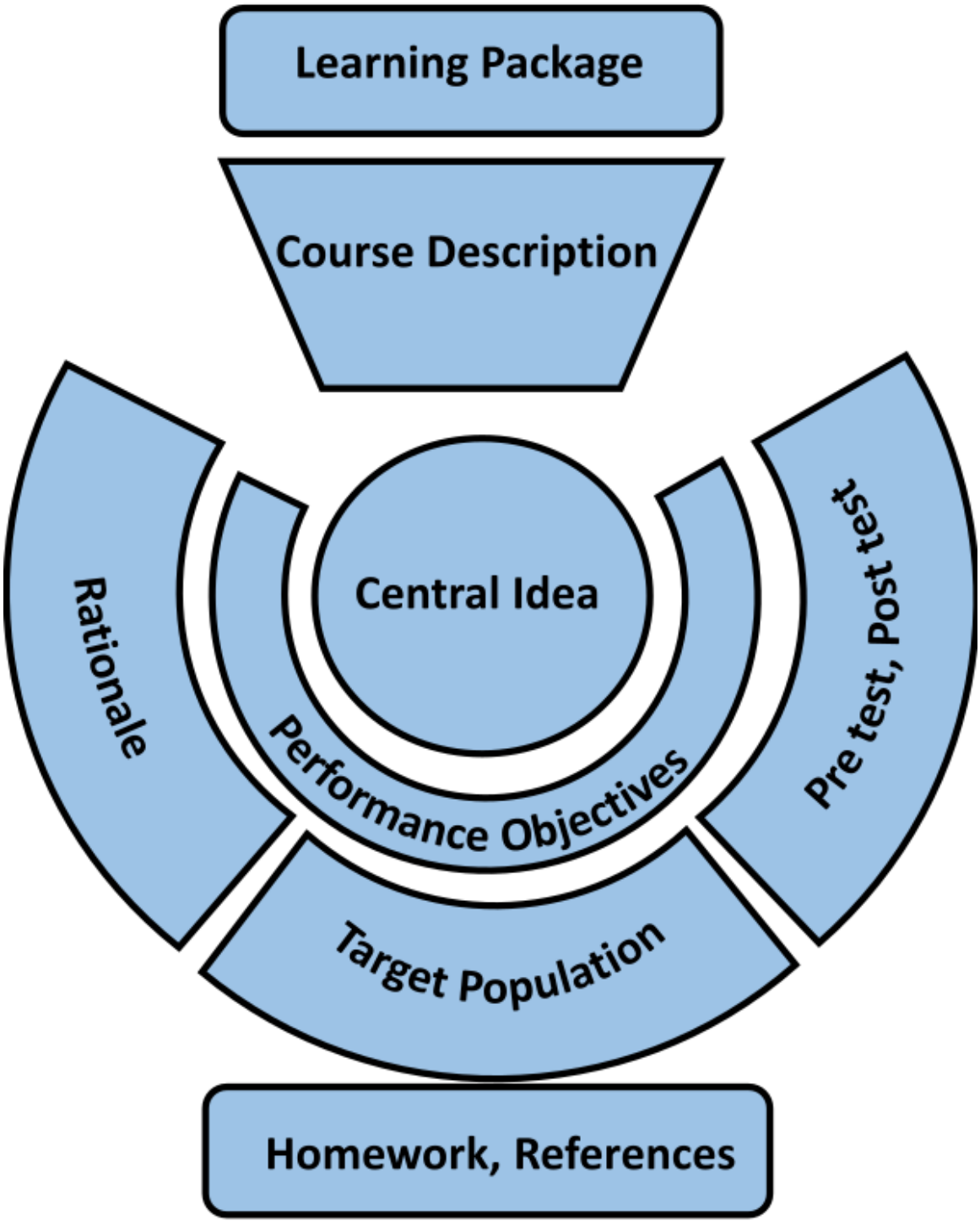
Advanced Web Design

For

Second Year Students

By

Mazin Salih Kadhim
Assistant Professor
Dep. Of Computer Networks
And Software Techniques
2025


Learning Package
Course Description
Rationale
Central Idea
Performance Objectives
Pre test, Post test
Target Population
Homework, References

وصف المقرر

1- اسم المقرر:
تصميم المواقع الالكترونية المتقدم
2- رمز المقرر:
لا يوجد
3- الفصل / السنة:
فصلي
4- تاريخ إعداد هذا الوصف :
١٤-٧-٢٠٢٥
5- أشكال الحضور المتاحة :
حضوري فقط
6- عدد الساعات الدراسية (الكلي)/ عدد الوحدات (الكلي):
60 ساعة فصليا / 4 ساعة اسبوعياً/ 4 وحدات
7- اسم مسؤول المقرر الدراسي (اذا اكثر من اسم يذكر)
الاسم: م.م مازن صالح كاظم الإيميل : mazin.s.kadhim@stu.edu.iq
اهداف المقرر
١. تمكين الطالب من برمجة مواقع إلكترونية ديناميكية باستخدام لغة PHP. ٢. إكساب الطالب مهارات الربط بين الواجهة الأمامية والخلفية باستخدام MySQL. ٣. تدريب الطالب على التعامل مع النماذج الإلكترونية ومعالجتها عبر طرق POST وGET. ٤. تعريف الطالب بمفاهيم الجلسات (Sessions) والكوكيز (Cookies) وكيفية استخدامها في حفظ البيانات. ٥. استخدام JavaScript لإضافة تفاعلية ديناميكية في صفحات الويب. ٦. تدريب الطالب على استخدام **الدوال**، المصفوفات، والتحكم في تدفق البرنامج بلغة الجافا سكريبت ٧. رفع وعي الطالب بأهمية الأمان البرمجي ومعالجة الأخطاء. ٨. تنمية قدرات الطالب على إنشاء مشاريع ويب متكاملة تجمع بين PHP, MySQL, وJavaScript.

9- استراتيجيات التعليم والتعلم	
الاستراتيجية	١ -استراتيجية التعليم تخطيط المفهوم التعاون ٢ -استراتيجية التعليم العصف الذهني. ٣ - استراتيجية التعليم سلسلة الملاحظات

10- بنية المقرر

الأسبوع	الساعات	مخرجات التعلم المطلوبة	اسم الوحدة او الموضوع	طريقة التعلم	طريقة التقييم
١-٦	٢ ساعة	• لغة PHP، مقدمه للغة PHP، تشغيل Windows IIS 5.0، إضافة PHP الى IIS، إضافة MySQL الى IIS، بنية ملفات PHP • بروتوكولات الأنترنت، التعليقات، المتغيرات، الأرقام، العمليات الحسابية، متغيرات النظام، الثوابت، معرفة وتحويل أنواع البيانات، دوال الوقت والتاريخ • النماذج GET، POST، الأوامر الشرطية، العبارة IF، المعاملات المنطقية، تعدد الشروط، تداخل العبارات الشرطية • العبارة Switch، التخلص من وسوم الـ html التكرارات والمصفوفات، دوال المصفوفات، فرز المصفوفات، دوال المصفوفات الاضافية، مصفوفات متعددة الابعاد • ترتيب الكود البرمجي Function، Print، مدى المتغيرات، المتغيرات المستقرة، اشتمال الملفات)، تتبع وتصيد ومنع الأخطاء (أنواع الأخطاء، الأخطاء المنطقية، تفادي الأخطاء، Regular Expressions، صناعة فئة الحروف) • التعامل مع العميل Cookies، Session، قراءة وكتابة معلومات في ملف txt	لغة PHP، مقدمه للغة PHP، تشغيل Windows IIS 5.0، إضافة PHP الى IIS، إضافة MySQL الى IIS، بنية ملفات PHP	التطبيق العملي	الامتحانات اليومية والواجبات والاختبارات العملية
٧-١٢	٢ ساعة	• استخدام Java Script لغة الجافا سكريبت، الشكل العام الذي سيكون عليه برنامج جافا سكريبت	استخدام Java Script لغة الجافا سكريبت، الشكل العام الذي سيكون	التطبيق العملي	الامتحانات اليومية والواجبات

٧-١٢	٢ ساعة	• الإعلان عن المتغيرات، المعاملات الحسابية، المعاملات المنطقية، عبارات التحكم، SWITCH، التكرار، الأحداث، • إنشاء زر لإرسال بريد إلكتروني WHILE، الدوال، النماذج، • المصفوفات، الكائنات، السلاسل الحرفية، معلومات تطبيقية	عليه برنامج جافا سكريبت	التطبيق العملي	والاختبارات العملية
١٣-١٥	٢ ساعة	• PhP • MySQL	لغات PhP and MySQL	التطبيق العملي	الامتحانات اليومية والواجبات والاختبارات العملية

11- تقييم المقرر

توزيع كالتالي: ٢٠ درجة امتحانات نظرية لمنتصف الفصل الاول. ٢٠ درجة امتحانات عملية لمنتصف الفصل الاول .10درجات امتحانات يومية وتقييم مستمر. ٥٠ درجة امتحان نهاية الفصل

12- مصادر التعلم والتدريس

الكتب المقررة المطلوبة (المنهجية أن وجدت)	Learning Web Design
المراجع الرئيسة (المصادر)	Alam, Mansaf, and Bashir Alam. *Digital Logic Design*. PHI Learning Pvt. Ltd., 2015.
الكتب والمراجع الساندة التي يوصى بها (المجلات العلمية، التقارير....)	ACM Transactions on the Web (TWEB). Welling, L., & Thomson, L. (2017). *PHP and MySQL Web Development*. Addison-Wesley.
المراجع الإلكترونية، مواقع الانترنيت	W3Schools Online Web Tutorials

وزارة التعليـم العالـي والبحـث العلمـي
الجامعـة التقنيـة الجنوبيـة
المعهـد التكنولوجـي بصـرة
قسـم التقنيـات الالكترونيـة

حقيبـــة تعليميـــة

فـــي

تصميم المواقع المتقدم

لطلبـــة المرحلـــة الثانية

(1)

By

Mazin Salih Kadhim
Assistant Professor
Dep. Of Computer Networks
And Software Techniques
2025

١/نظرة عامة

A / 1 الفئة المستهدفة:-

طلبــة المرحلــة الأولـى

المعهـد التقنــي في البصــرة

قســم تقنيات شبكات وبرامجيات الحاسوب

B/ 1الدوافع -:

- سد الفجوة بين المعرفة النظرية والتطبيق العملي في مجال تصميم المواقع المتقدمة، حيث تُعد لغة PHP أحد أهم لغات البرمجة المستخدمة في تطوير الويب الديناميكي.
- تشجيع الابتكار من خلال تطبيقات عملية تسمح للطلاب ببناء مشاريع واقعية، مثل أنظمة تسجيل الدخول أو نماذج التواصل.
- مواكبة التطور التكنولوجي في مجال تطوير الويب، حيث تُعد PHP لغة حيوية في أنظمة إدارة المحتوى مثل (WordPress) والتطبيقات الخدمية.

1 /C الفكرة الرئيسة -:

تهدف المادة إلى تحويل الطلاب إلى مبرمجين قادرين على بناء تطبيقات ويب فعالة وآمنة باستخدام عدة لغات ومن اهمها لغة PHP.

1 /D الأهداف السلوكية

بعد دراسة الوحدة الأولى، سيكون الطالب قادرًا على :

- فهم أساسيات لغة PHP واستخداماتها في تطوير الويب.
- التعامل مع المتغيرات، الجمل الشرطية، والحلقات فيPHP.
- فهم واستخدام المصفوفات والدوال فيPHP.
- تطبيق مفاهيم الأمان مثل التحقق من صحة البيانات ومعالجة الأخطاء.

٢/ الاختبارات القبلية

- ما هو ملف PHP؟
- أساسيات كتابة كود PHP (بنية الملفات، التعليقات، المتغيرات، البروتوكولات)

٣/مقدمه للغة PHP

ما هي لغة PHP ؟

- PHP هي اختصار لـ ""PHP: Hypertext Preprocessor
- PHPهي لغة برمجة مفتوحة المصدر مستخدمة على نطاق واسع
- يتم تنفيذ نصوص PHPعلى الخادم
- PHP مجانية للتنزيل والاستخدام

ما هو ملف PHP ؟

- تحتوي ملفات PHPعلى نصوص و HTMLو CSSو JavaScriptو PHP code
- يتم تنفيذ كود PHPعلى الخادم، ويتم إرجاع النتيجة إلى المتصفح بتنسيق HTML عادي
- ملفات PHPلها امتداد "."php

ما الذي يمكن أن تفعله PHP ؟

١ ـ يمكن لـ PHPإنشاء محتوى صفحة ديناميكي

٢ ـ يمكن لـ PHP إنشاء الملفات على الخادم وفتحها وقراءتها وكتابتها وحذفها وإغلاقها

٣ ـ يمكن لـ PHP جمع بيانات النموذج

٤ ـ يمكن لـ PHP إرسال واستقبال ملفات تعريف الارتباط

٥ ـ يمكن لـ PHPإضافة أو حذف أو تعديل البيانات في قاعدة البيانات الخاصة بك

٦ ـ يمكن استخدام PHP للتحكم في وصول المستخدم

٧ ـ يمكن لـ PHPتشفير البيانات

٨ ـ مع PHP، لا تقتصر على إخراج HTML. يمكنك إخراج الصور وملفات PDF وحتى أفلام فلاش. يمكنك أيضًا إخراج أي نص، مثل XHTMLوXML.

قواعد PHP الأساسية

- يمكن وضع نص PHPفي أي مكان في المستند.
- يبدأ نص PHPبـ <؟ php وينتهي بـ؟>:

```
<?php
// PHP code goes here
?>
```

بنية ملفات PHP

- يحتوي ملف PHPعادةً على علامات HTMLوبعض أكواد PHPالنصية.
- أدناه، لدينا مثال لملف PHPبسيط، مع برنامج نصي PHPيستخدم وظيفة PHP مضمنة " "echoلإخراج النص " "!Hello Worldعلى صفحة الويب:
- ملاحظة: تنتهي عبارات PHP بفاصلة منقوطة (;).

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

<h1>My first PHP page</h1>

<?php
echo "Hello World!";
?>

</body>
</html>
```

My first PHP page

Hello World!

حساسية حالة PHP

في PHP، الكلمات الرئيسية (على سبيل المثال if، else، while، echo، إلخ.)، الفئات، الوظائف، والوظائف المعرفة من قبل المستخدم ليست حساسة لحالة الأحرف.

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

<?php
ECHO "Hello World!<br>";
echo "Hello World!<br>";
EcHo "Hello World!<br>";
?>

</body>
</html>
```

Hello World!
Hello World!
Hello World!

- نظر للمثال ادناه؛ فقط العبارة الأولى ستعرض قيمة متغير اللون $! هذا بسبب التعامل مع $ color و $ COLORو $ coLOR على أنها ثلاث متغيرات مختلفة:

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

<?php
$color = "red";
echo "My car is " . $color . "<br>";
echo "My house is " . $COLOR . "<br>";
echo "My boat is " . $coLOR . "<br>";
?>

</body>
</html>
```

My car is red
My house is
My boat is

بروتوكولات الأنترنت

- عنوان بروتوكول الإنترنت (IP) هو رقم تعريف فريد يُخصص لكل جهاز متصل بالإنترنت. تعريف عنوان IP هو تسمية رقمية تُخصص للأجهزة التي تستخدم الإنترنت للتواصل. تتشارك أجهزة الكمبيوتر التي تتواصل عبر الإنترنت أو عبر الشبكات المحلية المعلومات مع موقع محدد باستخدام عناوين IP.

- كيف يعمل عنوان IP؟

يساعد عنوان IP جهازك، أيًا كان الجهاز الذي تتصل به بالإنترنت، على العثور على أي بيانات أو محتوى موجود لتسهيل استرجاعه.

- تشمل المهام الشائعة لعنوان IP تحديد هوية المضيف أو الشبكة، أو تحديد موقع الجهاز. عنوان IP ليس عشوائيًا، إذ يعتمد إنشاء عنوان IP على الرياضيات. تُخصص هيئة أرقام الإنترنت المخصصة (IANA) عناوين IP وتُنشئها. يتراوح النطاق الكامل لعناوين IP من 0,0,0,0 إلى 255.255.255.255

باستخدام التخصيص الرياضي لعنوان IP، يمكن إنشاء تعريف فريد للاتصال بوجهة معينة.

التعليقات في PHP

- التعليق في كود PHPهو سطر لا يتم تنفيذه كجزء من البرنامج. والغرض الوحيد منه هو أن يقرأه شخص يبحث في الكود.
- يمكن استخدام التعليقات من أجل ان يفهم الآخرين التعليمات البرمجية الخاصة بك
- ذكّر نفسك بما فعلته - لقد جبر معظم المبرمجين العودة إلى عملهم بعد عام أو عامين واضطرارهم إلى إعادة اكتشاف ما فعلوه.
- يمكن أن تذكرك التعليقات بما كنت تفكر فيه عندما كتبت الكود تدعم PHPعدة طرق للتعليق:

Syntax for single-line comments:

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

<?php
// This is a single-line comment

# This is also a single-line comment
?>

</body>
</html>
```

Syntax for multiple-line comments:

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

<?php
/*
This is a multiple-lines comment block
that spans over multiple
lines
*/
?>

</body>
</html>
```

Using comments to leave out parts of the code:

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

<?php
// You can also use comments to leave out parts of a code line
$x = 5 /* + 15 */ + 5;
echo $x;
?>

</body>
</html>
```

متغيرات في الPHP

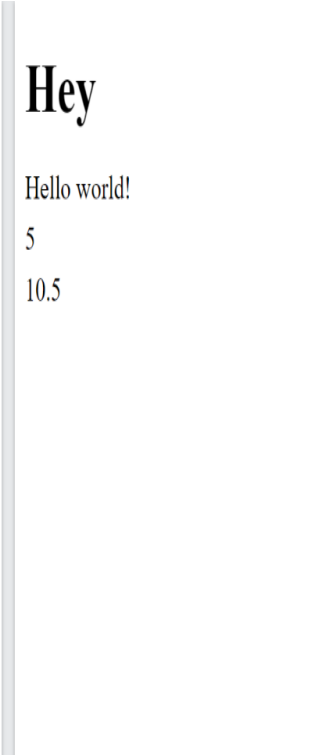
- المتغيرات هي "حاويات" لتخزين المعلومات.
- إنشاء (إعلان) متغيرات PHP في PHP، يبدأ المتغير بعلامة $ ، متبوعة باسم المتغير:

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

<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;

echo $txt;
echo "<br>";
echo $x;
echo "<br>";
echo $y;
?>

</body>
</html>
```



- يمكن أن يكون للمتغير اسم قصير (مثل x و y) أو اسم وصفي أكثر (العمر، اسم السيارة، إجمالي الحجم).
- قواعد متغيرات PHP:
- يبدأ المتغير بعلامة $، متبوعًا باسم المتغير
- يجب أن يبدأ اسم المتغير بحرف أو بشرطة سفلية
- لا يمكن أن يبدأ اسم المتغير برقم
- لا يمكن أن يحتوي اسم المتغير إلا على أحرف أبجدية رقمية وشرطات سفلية (A-z و ٠-9 و _)
- أسماء المتغيرات حساسة لحالة الأحرف ($ age و $ AGE متغيرين مختلفين)

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

<?php
$txt = "W3Schools.com";
echo "I love $txt!";
?>

</body>
</html>
```

I love W3Schools.com!

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

<?php
$x = 5;
$y = 4;
echo $x + $y;
?>

</body>
</html>
```

```
9
```

نطاق متغيرات PHP

- في PHP ، يمكن التصريح عن المتغيرات في أي مكان في البرنامج النصي.
- نطاق المتغير هو جزء من البرنامج النصي حيث يمكن استخدام / الرجوع إلى المتغير.
- PHP لها ثلاثة نطاقات متغيرة مختلفة:

- محلي
- عالمي
- ثابتة

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

<?php
$x = 5; // global scope

function myTest() {
  // using x inside this function will generate an error
  echo "<p>Variable x inside function is: $x</p>";
}
myTest();

echo "<p>Variable x outside function is: $x</p>";
?>

</body>
</html>
```

Variable x inside function is:

Variable x outside function is: 5

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

<?php
$x = 5;
$y = 10;

function myTest() {
  global $x, $y;
  $y = $x + $y;
}

myTest();  // run function
echo $y; // output the new value for variable $y
?>

</body>
</html>
```

15

الأرقام

في هذا المقرر، لا تُستخدم الأرقام فقط كقيم ظاهرة، بل تُوظف أيضًا في منطق البرنامج لتحديد عدد مرات التكرار، والتحكم في تدفق الشيفرة، وتنسيق العناصر بصريًا عبر CSS أو JavaScript.

كما أن التعامل مع الأرقام يُعتبر من المهارات المهمة لبرمجة نماذج إلكترونية دقيقة، وحساب أسعار المنتجات، وتنسيق البيانات المعروضة للمستخدم، مما يعزز من جودة تجربة المستخدم وسهولة التفاعل مع الموقع.

فهم طريقة التعامل مع الأرقام يُعد من **الأسس البرمجية الضرورية** لإنشاء مواقع ويب متقدمة وتفاعلية وذات أداء فعّال.

الاستخدامات الشائعة للأرقام في تصميم المواقع الإلكترونية المتقدم:

1-معالجة بيانات النماذج:(Forms)

- جمع القيم الحسابية المدخلة (مثل الأسعار، الكميات، الحسابات).
- التحقق من أن الأرقام ضمن نطاق معين (مثل رقم الهاتف، العمر، الرمز السري).

2-عمليات حسابية باستخدام PHP أو:JavaScript

- الحساب التلقائي (مثل جمع السعر × الكمية).
- تحويل العملات أو الوحدات(kg ↔ lb) ، .($ ↔ €)
- إيجاد المتوسطات، النسب المئوية، أو المجاميع.

3-التحكم في تنسيق العرض:

- تحديد أحجام العناصر (عرض، ارتفاع) بوحدات رقمية مثلpx ، em. ,%
- استخدام الأرقام في الـ CSS للتحكم في التخطيطpadding) (margin, flex, grid .

4-تنسيق الأرقام على الشاشة:

- تنسيق رقم عشري إلى عدد صحيح أو تحديد عدد المنازل العشرية:

١- مثلًا number.toFixed(2) :في.JavaScript

٢- أو استخدام number_format()في.PHP

5-تخزين واسترجاع الأرقام في قواعد البيانات:(MySQL)

- أنواع الحقول.INT, FLOAT, DECIMAL :
- الحساب المباشر داخل استعلامات:SQL

SELECT price * quantity AS total FROM orders;

6- التحكم في التكرار أو التفاعل:

- استخدام الأرقام في الحلقات (loops) مثل:

for ($i = 1; $i <= 10; $i++) { ... }

7- for ($i = 1; $i <= 10; $i++) { ... } أرقام الهوية أو التسلسلات:

- استخدام أرقام فريدة لتحديد المستخدمين.(User ID)
- العدادات التلقائية (Auto Increment) فيMySQL

العمليات الحسابية

تُعد العمليات الحسابية من الركائز الأساسية في برمجة المواقع الإلكترونية، خاصة عند التعامل مع النماذج الإلكترونية، قواعد البيانات، وإظهار البيانات للمستخدم. تُستخدم العمليات الحسابية في لغات مثل (JavaScript) لأداء العديد من المهام التفاعلية والحسابية.

مُعاملات PHP

تُقسّم المُعاملات إلى المجموعات التالية:

المُعاملات الحسابية

مُعاملات التعيين

مُعاملات المُقارنة

مُعاملات الزيادة/النقصان

المُعاملات المنطقية

مُعاملات السلاسل النصية

مُعاملات المصفوفات

مُعاملات التعيين الشرطية

Operator	Name	Example	Result
+	Addition	$x + $y	Sum of $x and $y
-	Subtraction	$x - $y	Difference of $x and $y
*	Multiplication	$x * $y	Product of $x and $y
/	Division	$x / $y	Quotient of $x and $y
%	Modulus	$x % $y	Remainder of $x divided by $y
**	Exponentiation	$x ** $y	Result of raising $x to the $y'th power

Operator	Name	Example	Result
==	Equal	$x == $y	Returns true if $x is equal to $y
===	Identical	$x === $y	Returns true if $x is equal to $y, and they are of the same type
!=	Not equal	$x != $y	Returns true if $x is not equal to $y
<>	Not equal	$x <> $y	Returns true if $x is not equal to $y
!==	Not identical	$x !== $y	Returns true if $x is not equal to $y, or they are not of the same type
>	Greater than	$x > $y	Returns true if $x is greater than $y
<	Less than	$x < $y	Returns true if $x is less than $y
>=	Greater than or equal to	$x >= $y	Returns true if $x is greater than or equal to $y
<=	Less than or equal to	$x <= $y	Returns true if $x is less than or equal to $y
<=>	Spaceship	$x <=> $y	Returns an integer less than, equal to, or greater than zero, depending on if $x is less than, equal to, or greater than $y. Introduced in PHP 7.

الكلمة الرئيسية الثابتة PHP

- عادة، عند اكتمال / تنفيذ وظيفة، يتم حذف جميع متغيراتها. ومع ذلك، في بعض الأحيان نريد عدم حذف متغير محلي. نحن في حاجة إليها لوظيفة أخرى.
- للقيام بذلك، استخدم الكلمة الأساسية الثابتة عند تعريف المتغير لأول مرة:

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

<?php
function myTest() {
  static $x = 0;
  echo $x;
  $x++;
}

myTest();
echo "<br>";
myTest();
echo "<br>";
myTest();
?>

</body>
</html>
```

```
0
1
2
```

أنواع بيانات في PHP

ممكن للمتغيرات تخزين أنواع مختلفة من البيانات، ويمكن لأنواع البيانات المختلفة القيام بأشياء مختلفة.

يدعم PHP أنواع البيانات التالية:

- String
- Integer
- Float (floating point numbers - also called double)
- Boolean
- Array
- Object
- NULL

العدد الصحيح PHP Integer

نوع بيانات العدد الصحيح هو رقم غير عشري بين 2 ,147,483,648 و 2,147,483,647

قواعد للأعداد الصحيحة:

- يجب أن يحتوي العدد الصحيح على رقم واحد على الأقل
- يجب ألا يحتوي العدد الصحيح على فاصلة عشرية

❖ يمكن أن يكون العدد الصحيح موجبًا أو سالبًا

❖ يمكن تحديد الأعداد الصحيحة في: التدوين العشري (الأساس 10) ، أو النظام الست عشري (الأساس 16) ، أو الرقم الثماني (الأساس 8) ، أو الترميز الثنائي (الأساس 2)

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

<?php
$x = 5985;
var_dump($x);
?>

</body>
</html>
```

```
int(5985)
```

العدد العشري PHP Float

العدد العشري (رقم الفاصلة او الكسري) هو رقم به فاصلة عشرية أو رقم في شكل أسي.

في المثال التالي، يمثل x عددًا عشريًا. ترجع الدالة var_dump () نوع البيانات وقيمتها:

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

<?php
$x = 10.365;
var_dump($x);
?>

</body>
</html>
```

```
float(10.365)
```

العدد المنطقي PHP Boolean

يمثل القيمة المنطقية حالتين محتملتين FALSE أو TRUE :

```
$x = true;
$y = false;
```

Booleans are often used in conditional testing. You will learn more about conditional testing in a later chapter of this tutorial.

دوال الوقت

أولاً: JavaScript : (جهة العميل)

JavaScript تستخدم كائن Date للتعامل مع الوقت والتاريخ مباشرة داخل المتصفح.

◇ أهم الدوال:

- new Date ( ) ← إنشاء تاريخ ووقت
- getFullYear ( ) ← ترجع السنة (مثل 2025).
- getMonth ( ) ← ترجع رقم الشهر (من 0 إلى 11).
- getDate ( ) ← ترجع اليوم في الشهر.
- getHours(), getMinutes(), getSeconds() ← للحصول على الوقت الحالي.

```
Edit  Copy

let now = new Date();
console.log(now.getFullYear());      // 2025
console.log(now.toLocaleString());  // عرض التاريخ والوقت بصيغة محلية
```

ثانيًا: PHP (جهة الخادم)

تُستخدم PHP لمعالجة الوقت على الخادم وهي مهمة لتسجيل الأحداث والتقارير الزمنية.

🔹 أهم الدوال:

- date ("Y-m-d") → ترجع التاريخ بصيغة مثل "20-7-2025"
- Time () →ترجع عدد الثواني منذ 1 يناير 1970
- Strtotime ("tomorrow") → تحويل نص إلى وقت.
- Mktime (0, 0, 0, 7, 20, 2025) →إنشاء طابع زمني لوقت مخصص.
- getdate() → إرجاع مصفوفة تحتوي على تفاصيل التاريخ.

```
echo date("Y-m-d H:i:s"); // 2025-07-20 13:45:00
```

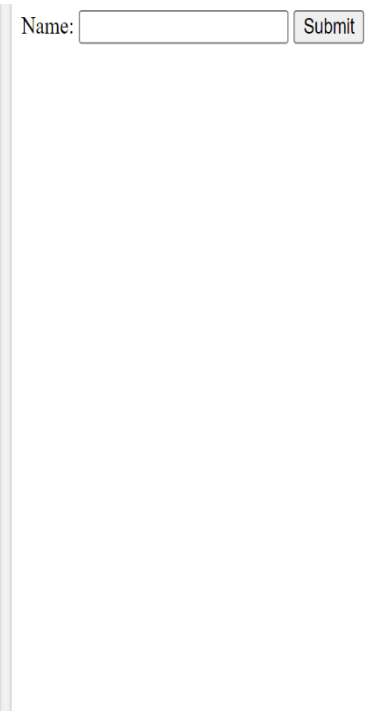
يستخدم لجمع بيانات النموذج بعد PHP هو متغير عالمي فائق في PHP $_POST على $_POST كما يستخدم ."post"=باستخدام طريقة HTML إرسال نموذج نطاق واسع لتمرير المتغيرات.

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

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // collect value of input field
    $name = $_POST['fname'];
    if (empty($name)) {
        echo "Name is empty";
    } else {
        echo $name;
    }
}
?>

</body>
</html>
```



PHP $_GET هو متغير عالمي فائق في PHP يُستخدم لجمع بيانات النموذج بعد إرسال نموذج HTML باستخدام طريقة."get" يمكن لـ $_GET أيضًا جمع البيانات المرسلة في عنوانURL.

الفرق بين GET و .POSTيُستخدم كلٌّ منهما لنقل البيانات من العميل إلى الخادم عبر بروتوكولHTTP ، ولكن الفرق الرئيسي بينهما يكمن في أن GET يحمل مُعامل الطلب مُلحقًا بسلسلة عناوينURL ، بينما يحمل POST مُعامل الطلب في نص الرسالة، مما يجعله أكثر أمانًا لنقل البيانات من العميل إلى الخادم.

كائن PHP

الفئات والكائنات هما الجانبان الرئيسيان للبرمجة كائنية التوجه.

الفئة هي قالب للكائنات، والكائن هو مثيل للفئة. عند إنشاء الكائنات الفردية، فإنها ترث جميع خصائصها وسلوكياتها، ولكن لكل كائن قيم مختلفة لهذه الخصائص. لنفترض أن لدينا فئة باسم "سيارة". يمكن أن تحتوي السيارة على خصائص مثل النموذج واللون، إلخ. يمكننا تعريف متغيرات مثل $model و$color، وهكذا، لحفظ قيم هذه الخصائص.

عند إنشاء الكائنات الفردية (فولفو، بي إم دبليو، تويوتا، إلخ)، فإنها ترث جميع خصائصها وسلوكياتها من الفئة، ولكن لكل كائن قيم مختلفة لهذه الخصائص. إذا أنشأت دالة __construct()، فستستدعي PHP هذه الدالة تلقائيًا عند إنشاء كائن من فئة.

مصفوفة PHP

تخزن المصفوفة قيمًا متعددة في متغير واحد. في المثال التالي cars عبارة عن مصفوفة.

ترجع الدالة PHP var_dump () نوع البيانات وقيمتها:

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

<?php
$cars = array("Volvo","BMW","Toyota");
var_dump($cars);
?>

</body>
</html>
```

```
array(3) {
  [0]=>
  string(5) "Volvo"
  [1]=>
  string(3) "BMW"
  [2]=>
  string(6) "Toyota"
}
```

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

<?php
class Car {
  public $color;
  public $model;
  public function __construct($color, $model) {
    $this->color = $color;
    $this->model = $model;
  }
  public function message() {
    return "My car is a " . $this->color . " " . $this->model . "!";
  }
}

$myCar = new Car("black", "Volvo");
echo $myCar -> message();
echo "<br>";
$myCar = new Car("red", "Toyota");
echo $myCar -> message();
?>

</body>
</html>
```

My car is a black Volvo!
My car is a red Toyota!

سلاسل PHP Strings

الرموز عبارة عن سلسلة من الأحرف ، مثل "Hello world!"."

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

<?php
echo strlen("Hello world!");
?>

</body>
</html>
```

```
12
```

تقوم دالة **PHP str_word_count()** بحساب عدد الكلمات في سلسلة.

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

<?php
echo str_word_count("Hello world!");
?>

</body>
</html>
```

```
2
```

تعمل دالة **PHP strrev()** على عكس السلسلة.

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

<?php
echo strrev("Hello world!");
?>

</body>
</html>
```

```
!dlrow olleH
```

تبحث دالة **strpos()** في **PHP** عن نص محدد داخل سلسلة نصية. إذا وُجد تطابق تُرجع الدالة موضع الحرف في التطابق الأول. إذا لم يُعثر على تطابق، تُرجع الدالة القيمة **FALSE**.

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

<?php
echo strpos("Hello world!", "world");
?>

</body>
</html>
```

```
6
```

تقوم دالة **PHP str_replace()** باستبدال بعض الأحرف بأحرف أخرى في سلسلة.

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

<?php
echo str_replace("world", "Dolly", "Hello world!");
?>

</body>
</html>
```

Hello Dolly!

العبارات الشرطية PHP if...else...else if Statements

في كثير من الأحيان، عند كتابة التعليمات البرمجية، قد ترغب في تنفيذ إجراءات مختلفة لشروط مختلفة.

يمكنك استخدام العبارات الشرطية في التعليمات البرمجية الخاصة بك للقيام بذلك.

في PHP، لدينا العبارات الشرطية التالية:

- جملة if - تُنفذ بعض التعليمات البرمجية إذا تحقق شرط واحد.
- جملة if...else - تُنفذ بعض التعليمات البرمجية إذا تحقق شرط ما وتعليمة برمجية أخرى إذا كان هذا الشرط خاطئًا.
- جملة if...elseif...else - تُنفذ تعليمات برمجية مختلفة لأكثر من شرطين.
- جملة switch - تختار إحدى كتل التعليمات البرمجية العديدة المراد تنفيذها.

العبارة – IF

Syntax

```
if (condition) {
  code to be executed if condition is true;
}
```

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

<?php
$t = 5;

if ($t < 10) {
  echo "Have a good day!";
}
?>

</body>
</html>
```

Have a good day!

العبارة - if...else

Syntax

```
if (condition) {
  code to be executed if condition is true;
} else {
  code to be executed if condition is false;
}
```

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

<?php
$t = 5;

if ($t < 10) {
  echo "Have a good day!";
} else {
  echo "Have a good night!";
}
?>

</body>
</html>
```

Have a good day!

if...else if...else – عبارة

Syntax

```
if (condition) {
  code to be executed if this condition is true;
} elseif (condition) {
  code to be executed if first condition is false and this condition is true;
} else {
  code to be executed if all conditions are false;
}
```

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

<?php
$t = 15;
echo "<p>The hour (of the server) is " . $t;
echo ", and will give the following message:</p>";

if ($t < 10) {
  echo "Have a good morning!";
} elseif ($t < 20) {
  echo "Have a good day!";
} else {
  echo "Have a good night!";
}
?>

</body>
</html>
```

The hour (of the server) is 15, and will give the following message:

Have a good day!

عبارة switch –

Syntax

```
switch (n) {
  case label1:
    code to be executed if n=label1;
    break;
  case label2:
    code to be executed if n=label2;
    break;
  case label3:
    code to be executed if n=label3;
    break;
    ...
  default:
    code to be executed if n is different from all labels;
}
```

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

<?php
$favcolor = "red";

switch ($favcolor) {
  case "red":
    echo "Your favorite color is red!";
    break;
  case "blue":
    echo "Your favorite color is blue!";
    break;
  case "green":
    echo "Your favorite color is green!";
    break;
  default:
    echo "Your favorite color is neither red, blue, nor green!";
}
?>

</body>
</html>
```

Your favorite color is red!

الحلقات التكرارية

تُستخدم الحلقات لتنفيذ نفس كتلة التعليمات البرمجية مرارًا وتكرارًا، طالما تحقق شرط معين. في PHP، لدينا أنواع الحلقات التالية:

- while - تكرار كتلة التعليمات البرمجية طالما تحقق الشرط المحدد.
- do...while - تكرار كتلة التعليمات البرمجية مرة واحدة، ثم تكرار الحلقة طالما تحقق الشرط المحدد.
- for - تكرار كتلة التعليمات البرمجية لعدد محدد من المرات.
- foreach - تكرار كتلة التعليمات البرمجية لكل عنصر في المصفوفة.

While - حلقة

تنفذ حلقة while كتلة من التعليمات البرمجية طالما أن الشرط المحدد صحيح.

Syntax

```
while (condition is true) {
  code to be executed;
}
```

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

<?php
$x = 1;

while($x <= 5) {
  echo "The number is: $x <br>";
  $x++;
}
?>

</body>
</html>
```

The number is: 1

The number is: 2

The number is: 3

The number is: 4

The number is: 5

for - حلقة

يتم استخدام حلقة for عندما تعرف مسبقًا عدد المرات التي يجب تشغيل البرنامج النصي فيها.

Syntax

```
for (init counter; test counter; increment counter) {
  code to be executed for each iteration;
}
```

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

<?php
for ($x = 0; $x <= 10; $x++) {
  echo "The number is: $x <br>";
}
?>

</body>
</html>
```

The number is: 0
The number is: 1
The number is: 2
The number is: 3
The number is: 4
The number is: 5
The number is: 6
The number is: 7
The number is: 8
The number is: 9
The number is: 10

حلقة - foreach

تعمل حلقة foreach فقط على المصفوفات، ويتم استخدامها للتنقل عبر كل زوج مفتاح/قيمة في المصفوفة.

Syntax

```
foreach ($array as $value) {
  code to be executed;
}
```

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

<?php
$colors = array("red", "green", "blue", "yellow");

foreach ($colors as $value) {
  echo "$value <br>";
}
?>

</body>
</html>
```

red
green
blue
yellow

عبارة -Break

يمكن أيضًا استخدام عبارة break للخروج من الحلقة.

المثال ادناه يخرج من الحلقة عندما يكون x يساوي 4:

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

<?php
for ($x = 0; $x < 10; $x++) {
  if ($x == 4) {
    break;
  }
  echo "The number is: $x <br>";
}
?>

</body>
</html>
```

The number is: 0

The number is: 1

The number is: 2

The number is: 3

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

<?php
for ($x = 0; $x < 10; $x++) {
  if ($x == 4) {
    continue;
  }
  echo "The number is: $x <br>";
}
?>

</body>
</html>
```

The number is: 0

The number is: 1

The number is: 2

The number is: 3

The number is: 5

The number is: 6

The number is: 7

The number is: 8

The number is: 9

المصفوفات PHP Arrays

في PHP، المصفوفات هي هياكل بيانات متعددة الاستخدامات تسمح بتخزين قيم متعددة في متغير واحد. يمكنها استيعاب قيم من أنواع مختلفة، مثل السلاسل النصية، والأعداد الصحيحة، وحتى مصفوفات أخرى.

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

<?php
$cars = array("Volvo", "BMW", "Toyota");
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
?>

</body>
</html>
```

I like Volvo, BMW and Toyota.

إذا كان لديك قائمة من العناصر (قائمة بأسماء السيارات، على سبيل المثال)، فإن تخزين السيارات في متغيرات فردية قد يبدو على النحو التالي:

```
$cars1 = "Volvo";
$cars2 = "BMW";
$cars3 = "Toyota";
```

المصفوفة المفهرسة PHP Indexed Arrays

المصفوفات المفهرسة هي نوع من المصفوفات فيPHP ، حيث يرتبط كل عنصر فيها بفهرس رقمي .فهرس العنصر الأول هو 0، وفهرس العنصر الثاني هو 1، وهكذا .تتيح هذه الفهرسة التسلسلية سهولة الوصول إلى عناصر المصفوفة ومعالجتها.

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

<?php
$cars = array("Volvo", "BMW", "Toyota");
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
?>

</body>
</html>
```

I like Volvo, BMW and Toyota.

التكرار عبر مصفوفة مفهرسة

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

<?php
$cars = array("Volvo", "BMW", "Toyota");
$arrlength = count($cars);

for($x = 0; $x < $arrlength; $x++) {
  echo $cars[$x];
  echo "<br>";
}
?>

</body>
</html>
```

Volvo
BMW
Toyota

مصفوفات PHP الترابطية (PHP associative arrays)

المصفوفات الترابطية هي مصفوفات تستخدم مفاتيح مُسمّاة تُعيّنها لها.

هناك طريقتان لإنشاء مصفوفة ترابطية:

```php
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
```

or:

```php
$age['Peter'] = "35";
$age['Ben'] = "37";
$age['Joe'] = "43";
```

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

<?php
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
echo "Peter is " . $age['Peter'] . " years old.";
?>

</body>
</html>
```

Peter is 35 years old.

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

<?php
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

foreach($age as $x => $x_value) {
  echo "Key=" . $x . ", Value=" . $x_value;
  echo "<br>";
}
?>

</body>
</html>
```

Key=Peter, Value=35

Key=Ben, Value=37

Key=Joe, Value=43

مصفوفات PHP متعددة الأبعاد

المصفوفة متعددة الأبعاد هي مصفوفة تحتوي على مصفوفة واحدة أو أكثر. تدعم PHP المصفوفات متعددة الأبعاد التي يبلغ عمقها مستويين أو ثلاثة أو أربعة أو خمسة أو أكثر. ومع ذلك، يصعب على معظم المستخدمين إدارة المصفوفات التي يزيد عمقها عن ثلاثة مستويات.

PHP - Two-dimensional Arrays

A two-dimensional array is an array of arrays (a three-dimensional array is an array of arrays of arrays).

First, take a look at the following table:

Name	Stock	Sold
Volvo	22	18
BMW	15	13
Saab	5	2
Land Rover	17	15

We can store the data from the table above in a two-dimensional array, like this:

```
$cars = array (
  array("Volvo",22,18),
  array("BMW",15,13),
  array("Saab",5,2),
  array("Land Rover",17,15)
);
```

Now the two-dimensional $cars array contains four arrays, and it has two indices: row and column.

To get access to the elements of the $cars array we must point to the two indices (row and column):

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

<?php
$cars = array (
  array("Volvo",22,18),
  array("BMW",15,13),
  array("Saab",5,2),
  array("Land Rover",17,15)
);

echo $cars[0][0].": In stock: ".$cars[0][1].", sold: ".$cars[0][2].".<br>";
echo $cars[1][0].": In stock: ".$cars[1][1].", sold: ".$cars[1][2].".<br>";
echo $cars[2][0].": In stock: ".$cars[2][1].", sold: ".$cars[2][2].".<br>";
echo $cars[3][0].": In stock: ".$cars[3][1].", sold: ".$cars[3][2].".<br>";
?>

</body>
</html>
```

Volvo: In stock: 22, sold: 18.
BMW: In stock: 15, sold: 13.
Saab: In stock: 5, sold: 2.
Land Rover: In stock: 17, sold: 15.

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

<?php
$cars = array (
  array("Volvo",22,18),
  array("BMW",15,13),
  array("Saab",5,2),
  array("Land Rover",17,15)
);

for ($row = 0; $row < 4; $row++) {
  echo "<p><b>Row number $row</b></p>";
  echo "<ul>";
  for ($col = 0; $col < 3; $col++) {
    echo "<li>".$cars[$row][$col]."</li>";
  }
  echo "</ul>";
}
?>

</body>
</html>
```

Row number 0

- Volvo
- 22
- 18

Row number 1

- BMW
- 15
- 13

Row number 2

- Saab
- 5
- 2

Row number 3

- Land Rover
- 17
- 15

دوال PHP المُعرّفة للمستخدم

بالإضافة إلى دوال PHP المُدمجة، يُمكنك إنشاء دوال خاصة بك.

الدالة هي مجموعة من العبارات البرمجية التي يُمكن استخدامها بشكل متكرر في البرنامج.

لا تُنفّذ الدالة تلقائيًا عند تحميل الصفحة.

تُنفّذ الدالة عن طريق استدعاء الدالة.

❖ إنشاء دالة محددة للمستخدم في PHP

Syntax

```
function functionName() {
  code to be executed;
}
```

Note: A function name must start with a letter or an underscore. Function names are NOT case-sensitive.

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

<?php
function writeMsg() {
  echo "Hello world!";
}

writeMsg();
?>

</body>
</html>
```

Hello world!

وسيطات دالة PHP (PHP Function Arguments)

- يمكن تمرير المعلومات إلى الدوال عبر الوسائط. الوسائط تشبه المتغير تمامًا.
- تُحدَّد الوسائط بعد اسم الدالة، داخل الأقواس. يمكنك إضافة أي عدد تريده من الوسائط، ما عليك سوى فصلها بفاصلة.

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

<?php
function familyName($fname) {
  echo "$fname Refsnes.<br>";
}

familyName("Jani");
familyName("Hege");
familyName("Stale");
familyName("Kai Jim");
familyName("Borge");
?>

</body>
</html>
```

Jani Refsnes.

Hege Refsnes.

Stale Refsnes.

Kai Jim Refsnes.

Borge Refsnes.

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

<?php
function familyName($fname, $year) {
  echo "$fname Refsnes. Born in $year <br>";
}

familyName("Hege","1975");
familyName("Stale","1978");
familyName("Kai Jim","1983");
?>

</body>
</html>
```

Hege Refsnes. Born in 1975

Stale Refsnes. Born in 1978

Kai Jim Refsnes. Born in 1983

```php
<?php
function addNumbers(int $a, int $b) {
  return $a + $b;
}
echo addNumbers(5, "5 days");
// since strict is NOT enabled "5 days" is changed to int(5), and it will return 10
?>
```

```
10
```

```php
<?php declare(strict_types=1); // strict requirement ?>
<!DOCTYPE html>
<html>
<body>

<?php
function setHeight(int $minheight = 50) {
  echo "The height is : $minheight <br>";
}

setHeight(350);
setHeight();
setHeight(135);
setHeight(80);
?>

</body>
</html>
```

```
The height is : 350
The height is : 50
The height is : 135
The height is : 80
```

```php
<?php declare(strict_types=1); // strict requirement ?>
<!DOCTYPE html>
<html>
<body>

<?php
function sum(int $x, int $y) {
  $z = $x + $y;
  return $z;
}

echo "5 + 10 = " . sum(5,10) . "<br>";
echo "7 + 13 = " . sum(7,13) . "<br>";
echo "2 + 4 = " . sum(2,4);
?>

</body>
</html>
```

```
5 + 10 = 15
7 + 13 = 20
2 + 4 = 6
```

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

<?php
function add_five(&$value) {
  $value += 5;
}

$num = 2;
add_five($num);
echo $num;
?>

</body>
</html>
```

```
7
```

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

<?php
function maximum($x, $y) {

    if ($x > $y) {

        return $x;
    } else {

        return $y;
    }
}

$a = 23;
$b = 32;

$val = maximum($a, $b);
echo "The max of $a and $b is $val \n";
?>

</body>
</html>
```

```
The max of 23 and 32 is 32
```

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

<?php
function power($a, $b=2) {

    if ($b == 2) {

        return $a * $a;
    }

    $value = 1;

    for ($i = 0; $i < $b; $i++) {

        $value *= $a;
    }

    return $value;
}

$v1 = power(5);
$v2 = power(5, 4);

echo "5^2 is $v1 \n";
echo "5^4 is $v2 \n";
?>

</body>
</html>
```

```
5^2 is 25 5^4 is 625
```

❖ اكتب دالة PHP تقوم بالتحقق مما إذا كانت السلسلة بأكملها بأحرف صغيرة.

PHP Code:

```php
<?php
function is_str_lowercase($str1)
    {
     for ($sc = 0; $sc < strlen($str1); $sc++) {
            if (ord($str1[$sc]) >= ord('A') &&
            ord($str1[$sc]) <= ord('Z')) {
        return false;
            }
            }
        return true;
         }
var_dump(is_str_lowercase('abc def ghi'));
var_dump(is_str_lowercase('abc dEf ghi'));
?>
```

❖ اكتب دالة للتحقق مما إذا كان الرقم أوليًا أم لا.

PHP Code:

```php
<?php
function IsPrime($n)
{
 for($x=2; $x<$n; $x++)
    {
        if($n %$x ==0)
            {
             return 0;
            }
     }
   return 1;
    }
$a = IsPrime(3);
if ($a==0)
echo 'This is not a Prime Number.....'."\n";
else
echo 'This is a Prime Number..'."\n";
?>
```

متغيرات PHP العالمية – Superglobals

بعض المتغيرات المُعرّفة مسبقًا في PHP تُسمى "متغيرات شاملة فائقة"، مما يعني أنها قابلة للوصول دائمًا، بغض النظر عن نطاقها - ويمكنك الوصول إليها من أي دالة أو فئة أو ملف دون الحاجة إلى أي إجراء خاص.

- $GLOBALS
- $_SERVER
- $_REQUEST
- $_POST
- $_GET
- $_FILES
- $_ENV
- $_COOKIE
- $_SESSION

PHP $GLOBALS

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

<?php
$x = 75;
$y = 25;

function addition() {
  $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}

addition();
echo $z;
?>

</body>
</html>
```

```
100
```

PHP $_SERVER

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

<?php
echo $_SERVER['PHP_SELF'];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER['HTTP_REFERER'];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?>

</body>
</html>
```

```
/demo/demo_global_server.php
35.194.26.41
35.194.26.41
https://tryphp.w3schools.com/showphp.php?filename=demo_global_server
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0
Safari/537.36
/demo/demo_global_server.php
```

- $_SERVER['PHP_SELF'] يُرجع اسم ملف البرنامج النصي قيد التنفيذ حاليًا.
- $_SERVER['SERVER_NAME'] يُرجع اسم خادم المضيف (مثل www.w3schools.com).
- $_SERVER['SERVER_PROTOCOL'] يُرجع اسم بروتوكول المعلومات ومراجعته (مثل HTTP/1.1).
- $_SERVER['HTTP_HOST'] يُرجع رأس المضيف من الطلب الحالي.
- $_SERVER['HTTP_REFERER'] يُرجع عنوان URL الكامل للصفحة الحالية.
- $_SERVER['SCRIPT_NAME'] يُرجع مسار البرنامج النصي الحالي.

PHP $_REQUEST

PHP $_REQUEST هو متغير PHP عالمي فائق يُستخدم لجمع البيانات بعد إرسال نموذج HTML. تحوّل دالة htmlspecialchars() بعض الأحرف المحددة

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

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // collect value of input field
    $name = htmlspecialchars($_REQUEST['fname']);
    if (empty($name)) {
        echo "Name is empty";
    } else {
        echo $name;
    }
}
?>

</body>
</html>
```

Name: Submit
Ahmed

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

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // collect value of input field
    $name = $_POST['fname'];
    if (empty($name)) {
        echo "Name is empty";
    } else {
        echo $name;
    }
}
?>

</body>
</html>
```

عبارات PHP echo و print

- مع PHP، هناك طريقتان أساسيتان للحصول على الإخراج: echo و print.
- يمكن استخدام بيان echo مع أو بدون أقواس: echo أو echo ().

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

<?php
echo "<h2>PHP is Fun!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This ", "string ", "was ", "made ", "with multiple parameters.";
?>

</body>
</html>
```

PHP is Fun!

Hello world!

I'm about to learn PHP!

This string was made with multiple parameters.

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

<?php
$txt1 = "Learn PHP";
$txt2 = "W3Schools.com";
$x = 5;
$y = 4;

echo "<h2>" . $txt1 . "</h2>";
echo "Study PHP at " . $txt2 . "<br>";
echo $x + $y;
?>

</body>
</html>
```

Learn PHP

Study PHP at W3Schools.com

9

تتبع وتصيد ومنع الأخطاء في تصميم المواقع المتقدم (Error Handling & Debugging)

أثناء برمجة المواقع الإلكترونية، من الطبيعي أن تقع أخطاء (Errors) سواءً في الشيفرة البرمجية أو أثناء تنفيذ العمليات. لذا، يُعد تتبع الأخطاء (Debugging) ومعالجتها (Error Handling)جزءًا أساسيًا من بناء مواقع احترافية وموثوقة .إن مصطلح الـ debugهو من المصطلحات الشائعة والشيقة في عالم البرمجة، هذا المصطلح يشير إلى كيفية إصلاح أخطاء البرنامج وتوقعها قبل حدوثها

قد يكون من أهداف تتبع الأخطاء الحماية بقدر أهميه البرنامج الجاري العمل عليه أو الموقع فكلما

طرق تتبع الأخطاء:(Debugging)

1. استخدام أداة Developer Tools في المتصفح) مثل:Chrome)
   - علامة التبويب "Console" تظهر الأخطاء.
   - يمكنك رؤية قيم المتغيرات وتتبع التنفيذ خطوة بخطوة.
2. كتابة رسائل باستخدام :()console.log
   - تساعدك على طباعة القيم والتحقق منها أثناء التنفيذ.

```javascript
console.log("The total is: " + total);
```

3. استخدام النقاط الفاصلة:(Breakpoints)

- إيقاف مؤقت لتنفيذ الكود لفحص القيم قبل الاستمرار.

منع الأخطاء:(Error Prevention)

1. التحقق من صحة البيانات:(Validation)

- التأكد من أن المستخدم أدخل رقمًا صحيحًا، أو لم يترك الحقل فارغًا.

```javascript
if (name === "") {
alert("Please enter your name.");
}
```

2. كتابة الكود بطريقة منظمة ومقسّمة إلى دوال :(Functions)

- هذا يسهل التعديل والاختبار.

3. استخدام شروط الحماية عند التعامل مع قواعد البيانات:

- مثل التحقق من الاتصال أو فحص الاستعلام.

أنواع الأخطاء البرمجية:

النوع	الوصف
Syntax Errors 🟥	أخطاء في كتابة الكود (مثل نسيان فاصلة أو قوس).
Runtime Errors 🟧	تظهر أثناء تشغيل البرنامج (مثل قسمة على صفر).
Logic Errors 🟨	الكود يعمل لكن النتيجة غير صحيحة بسبب منطق خاطئ.

الأخطاء المنطقية (Logical Errors)

إن الأخطاء المنطقية هي الأكثر صعوبة في التتبع فقد تجد برنامجك يعمل بشكل صحيح وبكل سلامة ولكنه عند نقطة ما لا يتم تنفيذها كما تريد أنت.

مثال:

Edit Copy javascript

```javascript
let average = total + count / 2; // خطأ منطقي
// الصحيح:
let average = (total + count) / 2;
```

طريقة اكتشافها:

- استخدام console.log()لتتبع القيم.
- مراجعة منطق العمليات الحسابية والشروط.
- اختبار الكود ببيانات مختلفة.

تفادي الأخطاء

نصائح لتقليل الأخطاء في الشيفرة:

1. اكتب الكود بشكل منظم مع فواصل واضحة وتعليقات.
2. تحقق من صحة البيانات قبل استخدامها.
3. قسّم الكود إلى دوال صغيرة يمكن اختبارها بسهولة.
4. استخدم أدوات المتصفح .(Console + Breakpoints)
5. استعمل جمل try...catch لمعالجة الأخطاء:

Edit Copy

```
try {
  let result = x / y;
} catch (error) {
  alert("Error occurred: " + error.message);
}
```

هناك نوع آخر من الأخطاء المنطقية (unexpected) وهو لا يقوم بإيقاف البرنامج نهائياً بل يقوم بإخراج رسالة الخطأ في مكان الخطأ أو قد يقوم بتنفيذ البرنامج وإخراج البيانات بشكل غير صحيح أو قد لا يقوم بإخراج بيانات.

التعبيرات النمطية(Regular Expressions)

هي طريقة لفحص أو البحث عن نمط معين في سلسلة نصية)مثل البريد الإلكتروني، الأرقام، الكلمات

✳️ أمثلة الاستخدام في النماذج:

🔹 التحقق من بريد إلكتروني:

Edit Copy javascript

```javascript
let emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
if (!emailPattern.test(email)) {
  alert("Invalid email format");
}
```

🔹 التحقق من رقم هاتف:

Edit Copy javascript

```javascript
let phonePattern = /^[0-9]{10}$/;
```

.

صناعة فئة الحروف (Character Classes)

فئة الحروف أو Character Classesهي إحدى الأدوات القوية في التعبيرات النمطية (Regex) تُستخدم لتحديد مجموعة من الأحرف التي يمكن أن تُطابق جزءًا معينًا من النص. وهي مفيدة جدًا في تصميم المواقع عند التحقق من مدخلات المستخدم(كالاسم،كلمة المرور،الإيميل...).

أنواع فئات الحروف الشائعة:

التعبير	المعنى	مثال
`[abc]`	أي حرف من a أو b أو c	تطابق "a", "b", أو "c"
`[abc^]`	أي حرف **ليس** a أو b أو c	تطابق جميع الأحرف عدا تلك الثلاثة
`[a-z]`	أي حرف صغير من a إلى z	تطابق "m", "k", "z"...
`[A-Z]`	أي حرف كبير من A إلى Z	تطابق "B", "M", "T"...
`[0-9]`	أي رقم من 0 إلى 9	تطابق "3", "8", "0"...
`[a-zA-Z0-9]`	أي حرف أو رقم	تطابق "5", "d", "A"...

التعامل مع العميل في تصميم المواقع الإلكترونية المتقدم

في تصميم المواقع الإلكترونية، يشير التعامل مع العميل إلى كل ما يحدث على جهاز المستخدم باستخدام المتصفح، مثل التفاعل مع الصفحات، التحقق من البيانات، أو تخزين معلومات محلية باستخدام Cookies وSessions.

(ملفات تعريف الارتباط) Cookies

هي ملفات صغيرة تُ خزنعلى جهاز المستخدم من قبل المتصفح. تُستخدم لحفظ معلومات بسيطة مثل Cookies تفضيلات المستخدم أو حالة تسجيل الدخول.

◈ مثال على إنشاء Cookie باستخدام JavaScript:

```
document. Cookie = "username=Ali; expires=Fri, 31 Dec
2025 23:59:59 GMT; path=/";
```

الجلسات (Session)

تُستخدم لحفظ معلومات المستخدم على الخادم مؤقتًا خلال زيارته للموقع، وتستمر أو حتى يُغلق المتصفح Sessions تنتهي الجلسة.

باستخدام مثال PHP

```
session_start();
$_SESSION["username"] = "Ali";
```

قراءة وكتابة معلومات في ملف txt

1- كتابة إلى ملف

```php
<?php
$file = fopen("example.txt", "w"); // أو "a" للإضافة دون الحذف
fwrite($file, "مرحبا! هذه معلومات مكتوبة في الملف.\n");
fwrite($file, "سطر آخر من المعلومات.");
fclose($file);
?>
```

- ❖ لإنشاء الملف أو الكتابة عليه، ويمسح المحتويات السابقة "w"
- ❖ يضيف إلى نهاية الملف دون حذف المحتوى السابق "a"

2- قراءة محتوى الملف

```php
<?php
$content = file_get_contents("example.txt");
echo nl2br($content); // يعرض المحتوى مع تحويل \n في فواصل أسطر إلى HTML
?>
```

٤/ الاختبارات البعدية

- اكتب كود PHP يقوم بإنشاء نموذج (Form) يحتوي على **حقل اسم** وزر إرسال، وعند الإرسال يعرض "مرحباً [اسم]" في نفس الصفحة.
- اكتب دالة في **PHP** تأخذ رقمًا وتتحقق إن كان زوجيًا أو فرديًا، وتطبع النتيجة.

٥/ واجبات منزلية

- كتابة نصوص **PHP** بسيطة لطباعة الجمل واستخدام المتغيرات.
- إنشاء نموذج **HTML** وجمع البيانات باستخدام **$_POST** و**$_GET**
- مشروع نهائي يشمل إنشاء موقع ويب ديناميكي باستخدام**PHP.**
- تمارين على تصحيح الأخطاء وتحليل ناتج الأكواد (**debugging**).

وزارة التعليـم العالـي والبحـث العلمـي
الجامعـة التقنيـة الجنوبيـة
المعهـد التكنولوجـي بصـرة
قسـم التقنيـات الالكترونيـة

حقيبـــة تعليميـــة

فـــي

تصميم المواقع المتقدم

لطلبـــة المرحلـــة الثانية

(2)

By

Mazin Salih Kadhim
Assistant Professor
Dep. Of Computer Networks
And Software Techniques
2025

١/نظرة عامة

A / 1 الفئة المستهدفة:-

طلبــة المرحلــة الأولـى

المعهـد التقنــي في البصــرة

قســم تقنيات شبكات وبرامجيات الحاسوب

B/ 1الدوافع -:

- الحاجة لتطوير مواقع ديناميكية تعالج البيانات وتمكّن من بناء تطبيقات ويب تفاعلية وحديثة.
- زيادة أهمية لغات البرمجة وتقنيات مثل PHP وMySQL وJavaScript في سوق العمل ومجال تطوير المواقع.
- تمكين الطالب من التعامل مع المشكلات البرمجية الحقيقة وتعقيدات ربط واجهة الموقع بالبيانات، وبالتالي تحفيزه لتطوير مهارات عملية.

C/ 1الفكرة الرئيسة -:

- إتقان مفهوم تصميم المواقع المتقدم من خلال استخدام لغات وتقنيات مثل PHP لإنشاء مواقع تُعالج البيانات ديناميكيًا.
- ربط الجوانب النظرية (أنواع البيانات، البرمجة الشيئية، العمليات الحسابية والمنطقية) مع التطبيق العملي في مشاريع ويب حقيقية.

D/ 1 الأهداف السلوكية

بنهاية الوحدة، يتوقع من الطالب أن يكون قادراً على:

- تطبيق مبادئ البرمجة الكائنية وتوظيف الحلقات والدوال والمصفوفات بفعالية.

- التعامل مع قواعد البيانات (إنشاء واستعلام وتعديل محتوى قواعد البيانات باستخدام MySQL).

- التعامل مع الأخطاء البرمجية وتوظيف أساليب تصحيح الأخطاء (debugging).

٢/ الاختبارات القبلية

- أسئلة عامة لقياس معرفة الطالب السابقة حول HTML، مفاهيم المتغيرات، الحلقات، الشروط.

- تمارين قصيرة على أساسيات البرمجة (كتابة جمل شرطيّة أو معرفة أوامر بسيطة).

- تقييم نظري مبسط في بداية الكورس حول المفاهيم الرئيسة للبرمجة وقواعد البيانات.

٣/ استخدام JavaScript

تُعتبر لغة جافا سكريبت من أهم لغات البرمجة المستخدمة في تطوير واجهات تحديد المواقع الإلكترونية. فهي لا تحتوي على موقع تفاعلي وبرمجي، أي ماهي مخصصة لتصرفات المستخدم دون الحاجة إلى تحميل الصفحة.

في المقرر أن يتعلم الطالب استخدام JavaScript لبرمجة وظائف متكاملة ومتكاملة للذكاء والتفاعل للموقع، ونستخدم بشكل خاص لغات مثل HTML وCSS وPHP.

أهم استخدامات JavaScript في تصميم المواقع المتقدم:

1-معالجة النماذج:(Forms)

التحقق من صحة البيانات المدخلة قبل إرسالها.

منع إرسال النموذج إذا كانت الحقول غير مكتملة أو خاطئة.

2- **تنفيذ العمليات الحسابية مباشرة في المتصفح:**

حساب الفواتير، الكميات، الخصومات.

عرض النتائج فورًا دون الرجوع إلى الخادم.(Server)

3- **التحكم في تفاعل المستخدم:**

إظهار أو إخفاء عناصر عند الضغط على زر.

تغيير ألوان أو نصوص أو صور تلقائيًا.

4- **التعامل مع الأحداث:(Events)**

مثل الضغط، التمرير، إدخال بيانات.

تشغيل وظائف مخصصة بناءً على الحدث.

5- استخدام الشروط والحلقات والدوال:

بناء منطق تفاعلي متكامل داخل الموقع.

مثلاً: تغيير محتوى الصفحة بناءً على خيار المستخدم.

6- **التفاعل مع:DOM (Document Object Model)**

التعديل المباشر على عناصر الصفحة (نصوص – صور – جداول – أزرار).

7- **إرسال واستقبال البيانات:(AJAX)**

التواصل مع الخادم لتحميل بيانات جديدة بدون إعادة تحميل الصفحة.

أنواع الإعلان عن المتغيرات:

في JavaScript الحديثة (ES6 وما بعدها)، نستخدم ثلاث كلمات رئيسية للإعلان عن المتغيرات:

الكلمة	الاستخدام	الملاحظات
`var`	قديمة، لا يُنصح بها	لها نطاق واسع (function scope)، وقد تسبب مشاكل
`let`	حديثة ومستخدمة بكثرة	تُستخدم للمتغيرات التي تتغير قيمتها
`const`	للثوابت	تستخدم لمتغيرات لا تتغير بعد تعريفها

عمليات الجمع في لغة جافا سكريبت

عمليات الجمع Addition Operator يستخدم المعامل (+) في جمع قيمتين ويقوم بأرجاع نتيجة الجمع بين متغيرين او قيمتين أو نصين.

عملية الجمع (+) Addition Operator في لغة جافا سكريبت بين المتغيرات

```
let price = 1000;
let tax = 14;
let total = price + tax;
document.write("<b> Total : "+ total +"<b><br>");
```

عملية الجمع (+) Addition Operator في لغة جافا سكريبت بين الارقام.

```
document.write(5 + 9);
```

عملية الجمع (+) Addition Operator في لغة جافا سكريبت بين النصوص وتسمي هذة العمليةconcatenate

```
let fname = "Ahmed";
let lname = "Adly";
let fullName = fname +" " + lname;
document.write("<b> Full name is: "+ fullName +"<b>");
```

عمليات الطرح في لغة جافا سكريبت

عمليات الطرح Subtraction Operator يستخدم المعامل (-) في طرح قيمتين ويقوم بأرجاع نتيجة الطرح بين متغيررين او قيمتين.

عملية الطرح (-) Subtraction Operator في لغة جافا سكريبت بين المتغيرات.

```
let price = 1000;
let discound = 200;
let total = price - discound;
document.write("<b> Total : "+ total +"<b>");
```

عملية الطرح Subtraction Operator (-) في لغة جافا سكريبت بين الارقام.

```
document.write(9 - 4);
```

عمليات الضرب في لغة جافا سكريبت
عمليات الضرب Multiplication Operator يستخدم المعامل (*) في ضرب قيمتين ويقوم بأرجاع نتيجة الضرب بين متغيررين او قيمتين.

```
let price = 1000;
let quantity  = 3;
let total = price * quantity;
document.write("<b> Total : "+ total +"<b>");
```

عملية الضرب (*) Multiplication Operator في لغة جافا سكريبت.

```
document.write(5 * 10);
```

عملية الضرب (*) Multiplication الضرب بين الارقام والنصوص سوف يحدث مشكلة اثناء التشغيل ويقوم بأرجاع not a number بقيمة NaN.

```
let fname = "Ahmed";
let lname = "Adly";
document.write("<b>  "+ fname  * 10 +"<b>");
```

عمليات القسمة في لغة جافا سكريبت

عمليات Division Operator يستخدم المعامل (/) في قسمة قيمتين ويقوم بأرجاع نتيجة قسمة بين متغيرين او قيمتين.

```
document.write(300 / 3);
```

عملية القسمة (/) Division القسمة بين الارقام والنصوص سوف يحدث مشكلة اثناء التشغيل ويقوم بأرجاع not a number بقيمةNaN.

```
let fname = "Ahmed";
let lname = "Adly";
document.write("<b>  "+ fname  / 10 +"<b>");
```

المعاملات المنطقية (Logical Operators) في **JavaScript**

يتم استخدام **المعاملات المنطقية في JavaScript** بشكل أساسي للتحكم في تدفق البرنامج أثناء التحقق من المدخلات، التفاعل مع النماذج، أو بناء شروط لتنفيذ تعليمات معينة.

أهم المعاملات هي:

1- AND (&&) : ترجع **true** إذا كانت جميع الشروط صحيحة.

```
let username = "Ali";
let password = "12345";

if (username === "Ali" && password === "12345") {
  alert("تم تسجيل الدخول بنجاح");
}
```

2- OR (||): ترجع **true** إذا كان أحد الشروط على الأقل صحيحًا

```
let role = "admin";

if (role === "admin" || role === "editor") {
  alert("لديك صلاحية الوصول");
}
```

3- معامل NOT (!) : يعكس قيمة الشرط.

```
let isLoggedIn = false;

if (!isLoggedIn) {
  alert("يرجى تسجيل الدخول أولاً");
}
```

عبارات التحكم (Control Statements) في JavaScript

في مادة تصميم المواقع المتقدم، تُستخدم عبارات التحكم للتحكم في تسلسل تنفيذ البرنامج وفقًا لشروط أو حلقات تكرارية أو اختيار من بين عدة احتمالات.

1- if Statement

تنفذ كتلة من الكود إذا تحقق الشرط.

```
let age = 20;
if (age >= 18) {
  alert("مرحباً! يمكنك الدخول للموقع.");
}
```

2- if...else Statement

تنفذ كتلة إذا تحقق الشرط، وكتلة أخرى إذا لم يتحقق.

```
let age = 16;
if (age >= 18) {
  alert("مرحباً!");
} else {
  alert("عذراً، يجب أن يكون عمرك 18 أو أكثر.");
}
```

3- if...else if...else Statement

تستخدم لاختبار عدة شروط.

```
let grade = 85;
if (grade >= 90) {
  alert("ممتاز");
} else if (grade >= 75) {
  alert("جيد جداً");
} else {
  alert("يحتاج إلى تحسين");
}
```

4- switch Statement

تستخدم لاختيار حالة معينة من عدة احتمالات.

```
let day = 3;
switch (day) {
  case 1: alert("الأحد"); break;
  case 2: alert("الإثنين"); break;
  case 3: alert("الثلاثاء"); break;
  default: alert("يوم غير معروف");
}
```

عبارات التكرار(Loops)

- **for Loop**

تكرر الكود لعدد محدد من المرات.

```
for (let i = 1; i <= 5; i++) {
  console.log("العد: " + i);
}
```

- **while Loop**

تكرر الكود طالما تحقق الشرط.

```
let i = 1;
while (i <= 5) {
  console.log(i);
  i++;
}
```

- **do...while Loop**

تشبه while لكن تنفذ الكود مرة واحدة على الأقل.

```
let i = 1;
do {
  console.log(i);
  i++;
} while (i <= 5);
```

- **for...in / for...of**

تستخدم for...in مع خصائص الكائنات

تستخدم for...of مع عناصر المصفوفة

```
let colors = ["أزرق" ,"أخضر" ,"أحمر"];
for (let color of colors) {
  console.log(color);
}
```

لإنشاء زر لإرسال بريد إلكتروني في موقع ويب هناك طريقتين :

1- هذه أبسط طريقة لإنشاء زر يفتح نافذة إنشاء رسالة بريد إلكتروني في برنامج البريد الافتراضي للمستخدم، لكن لا يتم إرسال البريد مباشرة

```
<button onclick="window.location.href='mailto:example@example.com?subject=موضوع البريد&body=نص الرسالة'">
  أرسل بريد إلكتروني
</button>
```

2. باستخدام نموذج HTML مع زر إرسال (send) + خدمة خارجية أو خادم (لإرسال البريد فعلياً)

لإرسال البريد فعلياً من الموقع دون فتح برنامج البريد، تحتاج إلى:

نموذج HTML لجمع بيانات البريد (مثل البريد، الرسالة).

برمجة خادم (مثل PHP, Node.js) أو خدمة بريد خارجية (Formspree, EmailJS, SendGrid ...) مع JavaScript لإرسال البريد.

```
<form id="emailForm">
  <input type="email" name="email" placeholder="بريدك الإلكتروني" required>
  <textarea name="message" placeholder="اكتب رسالتك" required></textarea>
  <button type="submit">إرسال</button>
</form>

<script>
  document.getElementById('emailForm').addEventListener('submit', function(e) {
    e.preventDefault();
    // هنا يمكن إرسال البيانات عبر fetch أو خادم إلى API للبريد الإلكتروني
    alert("تم إرسال الطلب، سيتم معالجته!");
  });
</script>
```

الدوال Functions

الدالة هي كتلة من الأسطر البرمجية التي تنفّذ مجموعة من المهامّ وفق ترتيب محدَّد، مثلا: "آخذ السطل، أذهب إلى البئر، أسحب الماء منه وأعود.

تُعرَّف الدالة بالصياغة التالية:

```
function functionName (parameters) {
  // نفّذ المهام هنا
}
```

المصفوفات (Arrays) في جافا سكريبت

- المصفوفة هي بنية بيانات تُستخدم لتخزين مجموعة من العناصر (قيم)، حيث يمكن أن تحتوي على أنواع بيانات مختلفة مثل أرقام، نصوص، كائنات، ... إلخ.
- المصفوفة في جافا سكريبت هي في الأصل كائن (object) لكن تُعامل كمجموعة مرتبة من العناصر.
- تُستخدم الفهارس العددية للوصول إلى العناصر، وتبدأ الفهارس دائمًا من الرقم صفر (0).

إنشاء مصفوفة في جافا سكريبت

```javascript
let colors = ["أحمر", "أخضر", "أزرق"]; // مصفوفة تحتوي على 3 عناصر
```

للوصول لأي عنصر داخل المصفوفة، نستخدم الفهرس (index) الذي يبدأ من 0 مثلاً:

```javascript
console.log(colors[0]); // "أحمر"
console.log(colors[2]); // "أزرق"
```

يمكن إضافة عناصر جديدة باستخدام ()push، أو حذف آخر عنصر باستخدام ()pop:

```javascript
colors.push("أصفر"); // يضيف "أصفر" إلى نهاية المصفوفة
colors.pop(); // يحذف آخر عنصر من المصفوفة (هنا "أصفر")
```

٤/ الاختبارات البعدية

- اختبار كتابة دوال جافا سكريبت تقوم بتنفيذ العمليات الحسابية (جمع، طرح، ضرب، قسمة) وعرض النتائج مباشرة على الصفحة.
- اكتب كود **JavaScript** يقوم بما يلي:

١- إنشاء زر باسم "غيّر اللون."

٢- عند الضغط على الزر، تتغير خلفية الصفحة إلى لون عشوائي.

- صمم صفحة تحتوي على نموذج إدخال اسم، عند الإرسال:

تتحقق من أن الاسم غير فارغ.**JavaScript**

٥/ واجبات منزلية

- باستخدام **JSP + JDBC**، صمم تطبيق ويب بسيط:

١- صفحة إدخال بيانات طالب (اسم، عمر، تخصص).

٢- عند الإرسال، يتم تخزين البيانات في قاعدة بيانات **MySQL**.

٣- صفحة أخرى تعرض جميع الطلاب المخزنين في جدول.

- كتابة تقرير حول مشروع نهائي أو مقترح تحسين لموقع ويب برمجي.

وزارة التعليـم العالـي والبحـث العلمـي
الجامعـة التقنيـة الجنوبيـة
المعهـد التكنولوجـي بصـرة
قسـم التقنيـات الالكترونيـة

حقيبـــة تعليميـــة

فـــي

تصميم المواقع المتقدم

لطلبـــة المرحلـــة الثانية

(3)

By

Mazin Salih Kadhim
Assistant Professor
Dep. Of Computer Networks
And Software Techniques
2025

١/نظرة عامة

A / 1 الفئة المستهدفة:-

طلبــة المرحلــة الأولــى

المعهــد التقنــي في البصــرة

قســم تقنيات شبكات وبرامجيات الحاسوب

B/ 1الدوافع -:

- أهمية قواعد البيانات MySQL في مجال تصميم المواقع المتقدم، حيث أصبحت قواعد البيانات جزءًا رئيسيًا في بناء المواقع الديناميكية.
- الحاجة إلى تعلم كيف يتم تخزين البيانات واسترجاعها ومعالجتها بشكل فعّال باستخدام SQL وPHP.
- زيادة الطلب على المواقع التي تعتمد على قواعد بيانات لإدارة العملاء، المنتجات، الطلبات، والموظفين.
- تمكين الطالب من التعامل العملي مع قواعد البيانات وربطها بتطبيقات الويب.

C/ 1الفكرة الرئيسة -:

- التعرف على أساسيات قواعد البيانات MySQL وأنواع البيانات التي تدعمها.
- توظيف لغة PHP لربط الموقع الإلكتروني بقاعدة البيانات وإجراء عمليات الإدخال والاستعلام.
- التطبيق العملي على جداول مثل (Customers, Orders, Employees, Products) وتطوير القدرة على التعامل مع قواعد البيانات لإدارة معلومات المواقع.

1 /Dالأهداف السلوكية

بعد دراسة الوحدة الأولى، سيكون الطالب قادرًا على :

- فهم المفاهيم الأساسية لقواعد البيانات MySQL.
- إنشاء قاعدة بيانات وجدول جديد باستخدام أوامر SQL.
- استخدام PHP للاتصال بقواعد البيانات وتنفيذ الاستعلامات.
- ربط البيانات بصفحات الويب وتحديثها بطريقة ديناميكية.

٢/ الاختبارات القبلية

- اختبار المفاهيم الأساسية لـ SQL (هل يعرف الطالب معنى SELECT، INSERT، وما إلى ذلك).
- أسئلة تراكمية حول مفاهيم نظم إدارة قواعد البيانات والدالات البرمجية المرتبطة بها.
- "ما هو الأمر المستخدم لاسترجاع البيانات في SQL؟"

٣ /مقدمه للغة MySQL

تعتبر لغة MySQL أشهر نظام قواعد بيانات مفتوح المصدر . ان MySQLخاصة بقواعد البيانات .

تخزن البيانات في MySQL في كائنات قواعد بيانات تسمى جداول .

ان الجدول هو عبارة عن مجموعة من البيانات المدخلة و المتصلة كما يتألف الجدول من أعمدة و صفوف.

تعتبر قواعد البيانات مفيدة جداً عند تخزين المعلومات بشكل تصنيفي أي عند استخدام شركة لقواعد البيانات يمكن أن تستخدم الجداول التالية :

"Employees" و "Products" و "Customers" و "Orders" .

جداول قواعد البيانات

تحتوي غالباً قواعد البيانات على جدول او أكثر يتم تعريف كل جدول باسم مثل (Customers أو Orders) . كما يحتوي الجدول على صفوف تمسى rows أو records بالاضافة الى البيانات . في المثال التالي جدول يسمى **Persons** .

FIRSTNAME	LASTNAME	ADDRESS	COUNTRY
Mutaz	Hakmi	Timoteivn 10	SAR
Abdo	Mando	Borgvn 23	SAR
Abdo	Fajr	Storgt 20	SAR

يحتوي الجدول السابق على 4 اعمدة و 3 صفوف لكل شخص صف يتألف من الاسم والكنية والعنوان والدولة :

الأوامر Queries

الأوامر هي عبارة عن طلبات أو أسئلة.

باستخدام MySQL يمكن تقديم أوامر لقواعد البيانات معلومات معينة والتي تحتوي على امكانية تقديم النتائج من خلال الصفوف أو الأعمدة.

لاحظ الأمر التالي:

```
SELECT LastName FROM Persons
```

يختار الأمر السابق جميع البيانات الموجودة في عمود الكنية LastName من الجدول المسمى Persons ثم سيقوم بإعادة النتيجة كالتالي :

LASTNAME
Hakmi
Mando
Fajr

تحميل قواعد البياناتMySQL

اذا لم يكن لديك تطبيق يحتوي على PHP مع قواعد البيانات MySQL يمكنك تحميلها من موقعها الرسمى من خلال الرابط التالي :

http://www.mysql.com/downloads/

حقائق حول قواعد البياناتMySQL

أحد الحقائق المذهلة عن MySQL أنها تقوم بتخزين البيانات الضمنية الموجودة في التطبيقات و هذه الحقيقة مذهلة بسبب أم معظم الناس تعتقد أن MySQL تستخدم فقط مع التطبيقات التي تخزن بيانات صغيرة أو متوسطة.

لكن الحقيقة بأن MySQL تستخدم مع قواعد بيانات مع مواقع الانترنت التي تستخدم لتخزين هائل للبيانات و المستخدمين مثل **Friendster**و **Yahoo**و. **Google**

لمشاهدة الشركات التي تستخدم MySQL لقواعد البيانات من خلال الرابط التالي:

http://www.mysql.com/customers/

الاتصال مع قاعدة البيانات MySQL

تستخدم عادة MySQL مع لغة PHP .

إنشاء اتصال لقاعدة البيانات MySQL

قبل الوصول الى البيانات في قاعدة البيانات يجب إنشاء اتصال مع قاعدة البيانات .

باستخدام لغة PHP يمكن انشاء اتصال مع قاعدة البيانات من خلال

الوظيفة**mysql_connect()** .

التركيبة

```
mysql_connect(servername,username,password);
```

المتغيرات	الشرح
servername	اختياري . يحدد وجهة الاتصال مع السيرفر و غالباً مايكون localhost
username	اختياري . يحدد اسم المستخدم مع السيرفر . يمكن استخدام افتراضياً root .
password	اخيتاري . يحدد كلمة المرور مع السيرفر و غالباً ماتكون فارغة اذا لم تعيين كلمة مرور خاصة مسبقاً .

ملاحظة : هناك الكثير من القيم المستخدمة مع قواعد البيانات ولكن في الجدول أهمها فقط لكن لمزيد من

ادراج بيانات من نموذج الى قاعدة بيانات

الآن سنقوم بإنشاء نموذج HTML والذي بدوره سيقوم بإضافة صف جديد للجدول. Persons

هنا كود نموذج: *HTML*

```
<html>
<body>
<form action="insert.php" method="post">
Firstname: <input type="text" name="firstname" />
Lastname: <input type="text" name="lastname" />
Age: <input type="text" name="age" />
<input type="submit" />
</form>
</body>
</html>
```

عند ضغط المستخدم على زر الارسال في نموذج HTML سيتم ارسال المعلومات الى الملف insert.php

بالاتصال بقاعدة البيانات ثم يحوّل المعلومات المسجلة في النموذج الى متغيرات $_POST ثم يرسل mysql_query()أمراً الى INTO INSERT ليتم ادراج صف جديد الى جدول. Personss

هنا الكود الموجود في صفحة : insert.php

```
<?php
$con = mysql_connect("localhost","root","root");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);
$sql="INSERT INTO Persons (FirstName, LastName, Age)
VALUES
('$_POST[firstname]','$_POST[lastname]','$_POST[age]')";
if (!mysql_query($sql,$con))
{
die('Error: ' . mysql_error());
}
echo "1 record added";
mysql_close($con)

?>
```

SELECT الاختيار

يستخدم التعبير SELECT لاختيار البيانات من قاعدة البيانات.
التركيبة

```
SELECT column_name(s)
FROM table_name
```

لاستخدام التعبير يمكن استخدام الوظيفة الأوامر الى mysql_query() و بهذه الوظيفة يمكن استخدام لارسال الاتصال مع MySQLL.

مثال:

يختار المثال التالي جميع البيانات المخزنة في جدول Persons تستخدم الاشارة * لاختيار جميع البيانات الموجودة في الجدول:

```php
<?php
$con = mysql_connect("localhost","root","root");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);
$result = mysql_query("SELECT * FROM Persons");
while($row = mysql_fetch_array($result))
{
echo $row['FirstName'] . " " . $row['LastName'];
echo "<br />";
}
mysql_close($con);

?>
```

يخزن المثال السابق البيانات العائدة من خلال الوظيفة mysql_query()في المتغير $result . ثم سنقوم باستخدام mysql_fetch_array() لاسترجاع الصف الثاني من صفوف الجدول . أما الحلقة whilee فإنها ستدور حول جميع الصفوف في جدول البيانات وتقدم النتيجة.

لعرض النتيجة لجميع الصفوف يمكن استخدام متغير $row كالتالي:

```
($row['FirstName'] and $row['LastName'])
```

ستظهر النتيجة كالتالي:

Mutaz
Hakmi
Abdo
Mando
Abdo Fajr

عرض النتيجة في جدول HTML

يختار المثال التالي نفس البيانات في المثال السابق لكن سيتم عرض البيانات في جدول HTML كالتالي

```
<?php
$con = mysql_connect("localhost","root","root");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}
mysql_select_db("my_db", $con);
$result = mysql_query("SELECT * FROM Persons");
echo "<table border='1'>
<tr>
<th>Firstname</th>
<th>Lastname</th>
</tr>";
while($row = mysql_fetch_array($result))
{
echo "<tr>";
echo "<td>" . $row['FirstName'] . "</td>";
echo "<td>" . $row['LastName'] . "</td>";
echo "</tr>";
}
echo "</table>";
mysql_close($con);

?>
```

٤/ الاختبارات البعدية

- تقييم قدرة الطالب على كتابة استعلامات **SQL** صحيحة.

- اختبار عملي على إنشاء قاعدة بيانات وجداول باستخدام**.MySQL**

- اختبارات تطبيقية لكتابة كود **PHP** يتصل بقاعدة البيانات وينفذ العمليات الأساسية.

٥/ واجبات منزلية

- إنشاء مشروع بسيط يحتوي على اتصال بين موقع ويب وقاعدة بيانات باستخدام **PHP** و**MySQL**.
- مهام عملية على بناء استمارة **HTML** تربطها بملفات **PHP** لمعالجة البيانات.

٦/المصادر

١- Dally, W. J., & Harting, R. C. (2012). *Digital Design: A Systems Approach*. Cambridge University Press

٢- MySQL AB. (n.d.). *MySQL downloads*. Retrieved July 28, 2025, from

٣- W3Schools. (n.d.). *Learn web development*. Retrieved July 28, 2025

٤- Welling, L., & Thomson, L. (2017). *PHP and MySQL Web Development*. Addison-Wesley

٥- Duckett, J. (2014). *JavaScript and JQuery: Interactive Front-End Web Development*. Wiley

٦- ixon, R. (2021). *Learning PHP, MySQL & JavaScript*. O'Reilly Media