الفصل السادس

استخدام JSON لتخزين وجلب البيانات :-

تخزين البيانات

① مدى المتغيرات :-
هي الفترة من لحظة تعريف المتغير الى انه يصبح غير معروف ولا تستطيع استخدامه والوصول للقيمة التي يحملها

*اذا اردنا استخدام نفس المتغير داخل الدالة واجراء تعديل ما عليه يجب تمرير عنوان المتغير كوسيط للدالة باستخدام & قبل اسم المتغير كالتالي

```
$Var1 = 'Value'
Fanction test(&$Var1)
{
$Var1 = 'another value <br>
    ($Var1)
echo ($Var1);
```

او استخدام الدالة Gobal
لتعرف الدالة ان هذا المتغير عام وليس خاص

```
global $Var1;
```

* التنسيقة JSON :-

وهي اختصار Java script object Notation وهي طريقة في لغة جافا سكريبت للتعامل مع البيانات - وايضاً تستخدم في جلب البيانات من مواقع شهيرة مثل تويتر، وحالة الطقس من Yahoo

← يتم حفظ البيانات في تنسيقة JSON على شكل كائن وتوضع العناصر بين الاقواس { }

{ "Var1":10, "Var2":true, "Var3":Null ---- }

او على شكل مصفوفة توضع العناصر بين [ ]

[10, 2015, "Value", null, true]

يتم اسناد القيم للعناصر باستخدام (:) ، الفصل بينهم (,)
ويجب انه يكون العنصر بين علامتي اقتباس.

← كائن يحتوي على مصفوفة

{ "Var1":10, "Var2":[10, 20, 30] }

← مصفوفة تحتوي على كائن

من الدوال التي تتعامل مع التنسيق :-

① دالة Json_encod ← لتحويل الى تنسيق Json

② دالة Json_decode ← لتحويل تنسيق Json الى كائنات ومصفوفات يمكن التعامل معها .

ملاحظة : المصفوفات الترابطية hashtable في اللغة يتم تحويلها الى كائنات في تنسيق Json

أولاً : تحويل البيانات الى صيغة Json باستخدام Json_encode

* حالة (1) لدينا مصفوفة ترابطية بها قيم مختلفة

```
$data['var1'] = 10;
$data['var2'] = 2013;
$data['var3'] = null;
$data['var4'] = true;
$data['var5'] = 'value';
echo json_encode ($data);
```

{"var1":10,"var2":2013,"var3":null,"var4":true,"var5":"value"}

* حالة (2) المصفوفة العادية :-

```
$data[] = 10;
$data[] = 2013;
$data[] = null;
$data[] = true;
$data[] = 'value';
echo json_encode ($data);
```

[10, 2013, null, true, "value"]

ملاحظة :- يتم تحويل كل مصفوفة عادية الى Json وكل مصفوفة ترابطية تكون Json

حالة (3) : مصفوفة عادية تحتوي ع قيم و ع مصفوفات ترابطية و عادية

```
$data[] = 300;
$data[] :array(10,20,30);
$data[] = array("var1"=>12.3,12.8, "var2=> "value, 9000, "var3=>array, false;))
   echo json_encode($data);
```

عند تنفيذ البرنامج سيعطي مثالة للتالي :-

[300, [10,20,30] {"var1" = 12.3 , "0" :12.8 ; "var2" :"value, "1" :9000, "var5":[true,false]

حالة (4) مصفوفة ترابطية تحتوي ع قيم و ع مصفوفة عادية :-

```
$data = array("var1"=>12.3 ,12.8 , "var2"=> array("value1" "value2","value3"),900
 echo json_encode($data);
```

عند تنفيذ البرنامج السابق سيعطي النتيجة التالية :-

{ "var1":12.3, "0":128, "var2":["value1", "value2", "value3"], "1":9000}

خامساً : تحويل صيغة json الى كائنات ومصفوفات يمكن التعامل معها من خلال php
باستخدام دالة json_decode

ملاحظة بما اننا لم نتطرق للتعامل مع الكائنات حتى الان فدالة json_encode
تاخذ وسيط ثانيا في حالة اعطاء القيمة true يتم تحويل كائنات json الى
مصفوفات ترابطية hash table واذ اردت استخدام الكائن بدون
تحويله الى مصفوفة يمكن الوصول للعناصر باستخدام ←

① جلب كائن في تنسيق json وتحويله الى مصفوفة ترابطية

```
$json = '{"var1":10,"var2":true,"var3":null,"var4":"value","var5":[1,2,5]}';
$data1 = json_decode($json);
$data2 = json_decode($json,true);
// الوصول للعناصر من خلال الكائن
echo $data1->var4;
echo "<br>"
// الوصول للعناصر عن طريق مصفوفة ترابطية
echo $data2['var4'];
```

يطبع ⇐ value
value

دالة file_get_contents ← تقوم بحفظ البيانات التي يتم جلبها من الملف وفي حالة عدم جلب محتوى من الملف تعود الدالة بالقيمة false

fopen ← فتح الملف

die ← للخروج من الكود في حالة حدوث خطأ

fwrite ← الكتابة في الملف