أنواع البيانات (DataType) في PHP

1. **String (السلسلة النصية):** وهي عبارة عن أي قيمة يتم وضعها بين إشارتين تنصيص ثنائية (" ") ، أو إشارتين تنصيص مفردة (' ') .

```
$str ="hello";
$str1="hello";
```

ملاحظة: الدالة (Var-dump()) ، تعطي نوع المتغير وقيمته .

❖ الدوال المستخدمة مع ال **Strings** :

- ✓ **Strtoupper** : تقوم بتحويل النصوص إلى أحرف كبيرة .
- ✓ **Strtolower** : تقوم بتحويل النصوص إلى أحرف صغيرة .
- ✓ **Trim** : تقوم بحذف الفراغات من بداية ونهاية النصوص .
- ✓ **Srt_replace(search,replace,subject)** : تقوم بإستبدال جزء من نص بنص أخر و تأخذ ثلاث بارامترات
  - النص المراد إستبداله(**search**).
  - القيمة التي سنضعها مكان النص (**replace**) .
  - المتغير المراد إستبدال النص منه (**subject**).

2. **Intger (الأعداد الصحيحة)** ، مثلا :

```
$x=10;
```

3. **Float & Double (الأعداد العشرية) :**

```
$x=10.24252;
```

4. **Boolean :** يأخذ إحدى القيمتين (**true**) أو (**false**) :

```
$bool=true;
$bool=false;
```

5. **المصفوفات:** (هي مجموعة من المواقع المتسلسلة في الذاكرة)، تسخدم لإنشاء مجموعة متغيرات متسلسلة في الذاكرة ، حيث نستخدم ال **Arrays** لتسهيل الوصول إلى المتغيرات وإدارتها، حيث تكون جميع المتغيرات تابعة لل **Array** ، أي أننا نصل إليها باستخدام اسم واحد وهو اسم ال **Array** مع رقم المتغير(**Index**) ضمن ال **Array** كمايلي (**Arrayname[index_var]**) .

- ✓ مثال :لدينا (**7Emails**) عبارة عن سبع مواقع في الذاكرة ، لكنها مخزنة باسم واحد وهو اسم ال **Array** وليكن (**Emails**)، أي لدينا اسم واحد يحمل سبعة مواقع في الذاكرة كما، يوضح الشكل التالي :

✓ يتم تعريف المصفوفات في **PHP** ، كمايلي :

```
$ArrayName=array(value1,value2,value3,.........);
```

مثلا

```
$arrs=array("html","php","css","javascript");
$arri=(1,2,3,4,5,6);
```

✓ ويتم الوصول الى عناصر المصفوفة كما يلي :

```
$arrayname[index];
```

مثلا

$arrs[0]; ⟶ (html)

$arri[0]; ⟶ (1)

ولطباعته على الشاشة نسبقه بالدالة echo

✓ يتم تغير قيمة عنصر من عناصر المصفوفة وذلك بإسناد قيمة للعنصر كما يلي :

```
$arrs[2]=”bootstrap”;
```

✓ يتم إضافة عنصر جديد الى المصفوفة ، كما يلي :

```
$arrayName[]=value;
```

مثلا لإضافة عنصر الى المصفوفة ($arrs) كما يلي

```
$arrs[]=”jquery”;
```

- ✓ من الممكن أن تحتوي المصفوفة على قيم للعناصر من أنواع مختلفة ، (**int,String,float**)، وأيضا يمكن للمصفوفة أن تحتوي أحد عناصرها مصفوفة ثانية، وبالتالي للوصول لعناصر المصفوفة الثانية، كما يلي :

```
$Arr1[index_arr2][index_item_arr2];
```

```
$arr1=array(1,”software”,array(“php”,”css”,”html”);
```

للوصول الى العنصر "CSS" في المصفوفة الثانية كما يلي :

```
$arr1[2][1];
```

دوال المصفوفات :

الدالة (MAX) : تستخدم للوصول إلى أكبر عنصر في مصفوفة الارقام .

الدالة (min): تستخدم للوصول إلى أصغر عنصر في المصفوفة .

الدالة (count) : تستخدم لإيجاد عدد عناصر المصفوفة .

الدالة (sort) : تستخدم لترتيب عناصر المصفوفة .

الدالة (in_array): تستخدم للتأكد من وجود عنصر ما في المصفوفة وتأخذ بارامترين :

- العنصر المراد التأكد من وجوده .
- المصفوفة .

```
$number=array(1,10,12,90);
Echo (max($number);
Echo(min($number);
Echo(count($number);
Echo(sort($number);
Echo (in_array(item,arrayName));
```

المتغير objects: وهو عبارة عن نسخة يتم أخذها من كلاس ما، يحيث يمكننا الوصول الى جميع جميع الخصائص والدوال ضمن الكلاس، وسنتعرف بشكل أكبر على المتغير **objects** والكلاسات ، عند دراسة البرمجة غرضية التوجه في **PHP**:

```
Class car{
Var $price=50000;
}
$copyCar=new car();
Echo $copyCar -> price;
```

المتغير null : وهو المتغير الذي لم تسند له قيمة أو أسندت له القيمة (**null**).

بعض الدوال المستخدمة مع المتغيرات :

الدالة **is_null** : وتستخدم للتأكد إذا كان المتغير له قيمة أم لا، وتعيد الرقم **1** في حال كان المتغير **null** ولا تعيد قيمة في حال أسندت قيمة للمتغير.

الدالة **isset**: وتستخدم للتحقق من أن المتغير قد أسندت له قيمة ما ، وتعيد **1** إذا كان له قيمة و لا تعيد قيمة إذا كان المتغير **null**.

الدالة **is_bool**: للتأكد من أن المتغير من نوع **Boolean**.

الدالة **is_float**: للتأكد من أن المتغير من نوع **float**.

Casting تحويل المتغير من نوع بيانات الى أخر :

ملاحظة : الدالة **gettype** تستخدم لتعيد نوع المتغير الممرر لها .

ملاحظة : في **php** في حال كان لدينا متغير نصي ويحتوي على رقم، وقمنا بتطبيق عملية حسابية على هذا المتغير مع أرقام أخرى ، فإنه يقتص الرقم ويقوم بتطبيق العملية الحسابية عليه ، ويقوم بتحويله الى متغير عددي (**integer** ، **float** .....) :

```
$var1=" 4  month" ;  >>>>>>DataType :String
$var1=var1+5;>>>>>>result=9 & DataType :integer
```

يتم تحويل المتغير من نوع بيانات إلى أخر وذلك بأن نسبق المتغير بنوع البيانات المراد تحويل نوع المتغير اليه ، كما يلي:

```
$var1=10; >>>>>>>Datatype:intger.
$var2=(string)$var1; >>>>>Datatype :intger.
$var3=(intger)$var2>>>>>>>>.Datatype :intger.
```

ملاحظة : نستطيع تغير نوع بيانات المتغير باستخدام الدالة **settype** ، والتي تأخذ بارامترين :

- نوع البيانات المراد التحويل اليه .
- المتغير المراد تحويله .

Constant الثوابت :وهو المتغير الذي لايمكننا تغير قيمته، ويتم تعريف الثابت باستخدام الدالة **define** ، والتي تأخذ الشكل التالي :

```
Define (nameConstant,value,case_senstive);

Defin ("x","test",false);

Echo X;>>>>>>>>result:error;

Defin ("x","test",true);

Echo X;>>>>>>>>Result: test.
```

ال **case_senstive** تأخذ إما **true** أو **false** و في حالة ال **false** يصبح الثابت حساس لحالة الأحرف ، أما في حالة ال **true** يكون غير حساس لحالة الأحرف .

وطباعته أو التعامل معه يتم باستخدام اسمه فقط،كما يلي:

```
Echo x;
```

العوامل operator :

العوامل الحسابية (Arithmatic operator) :

- الجمع (+)
- الطرح (−)
- الضرب (*)
- القسمة (/)
- باقي القسمة (%)
- الأس (**)

```
$x=6;
$y=2;
Echo $x+$y;>>>>>8
Echo $x-$y;>>>>>>4
Echo $x*$y;>>>>>>12
Echo $x/$y;>>>>>>3
Echo $x%$y;>>>>>>>0
Echo $x**$y;>>>>>>36
```

معاملات الاسناد (Assignment operator) :

- معامل (=)
- معامل(=+)
- معامل (=-)
- معامل (=*)
- معامل (=/)
- معامل (=%)

```
$x=2;
$y=3;
$y+=2;>>>>5
$y-=2;>>>>1
$y*=2;>>>>6
$y/=2;>>>>>1
$y%=2;>>>>>>1
```

عوامل الزيادة والنقصان بواحد (increment&decrement operators) :

- معامل الزيادة (++) ولدينا نوعين :
- **Post increment** : $x++;
- **Pre increment** : ++$x;
- معامل النقصان (--) وأيضا لدينا نوعين :
- **Post decrement** : $x--;
- **Pre decrement** : --$x;

```
$x=2;
Echo $x++;
```

يطبع 2 وتصبح قيمة X تساوي3

```
Echo ++$x;
```

يطبع القيمة 4

```
Echo $x--;
```

يطبع 4 وتصبح قيمة x تساوي3

```
Echo --$x;
```

يطبع 2

عوامل المقارنة (comparison operator) :

- (==): يختر تساوي طرفين العامل (هل الطرف الأيمن يساوي الطرف الأيسر)، من حيث القيمة فقط.
- (===): يختر تساوي طرفين العامل (هل الطرف الأيمن يساوي الطرف الأيسر)، من حيث القيمة والنوع.
- (=!) (< >): يختبر عدم تساوي طرفين العامل، من حيث القيمة .
- (==!): يختبر عدم تساوي طرفين العامل، من حيث القيمة و النوع.
- (<): يخبر في حال كان الطرف الأيسر أكبر من الطرف الأيمن .
- (>): يخبر في حال كان الطرف الأيسر أصغر من الطرف الأيمن .
- (= <): يخبر في حال كان الطرف الأيسر أصغر من أو يساوي الطرف الأيمن.
- (=>) : يخبر في حال كان الطرف الأيسر أصغر من أو يساوي الطرف الأيمن.

م.أحمد ضوا

العوامل المنطقية (logical operator):

- and(&&) :يجب أن يكون طرفين العامل نتيجته (true)، ليعطي نتيجة (true)(جميع الشروط محققة يعطي True).
- or (| |):يجب أن يكون إحدى طرفين العامل نتيجته (true)، ليعطي نتيجة (true)، وفي حال كان الطرفين نتيجته false يعطي النتيجة false، شرط واحد على الأقل محقق يعطي true).
- xor : يعطي نتيجة (true) في حال كان إحدى الطرفين نتيجته true، والطرف الأخر false، وفي حال كان كلا من الطرفين (true) أو (false) فإنه يعطي النتيجة (false).
- !: يستخدم معامل النفي لعكس نتيجة شرط ما .

عوامل النصوص (String operator) :

- عامل (.) : يستخدم لدمج النصوص .
- عامل الإسناد النصي للدمج (=.): ويستخدم لدمج متغير بمتغير أخر ، كما يلي :

```
$x="soft";
$y="ware";
Echo $x . $y; >>>>>>>software
$y .= $x;
Echo $y;>>>>>>>software
```

عوامل المصفوفات (Array operator) :

ملاحظة: نستطيع تغير ال **index** لعناصر المصفوفة ، من أرقام إلى أسماء من إختيارنا ، ويصبح تعريف المصفوفة كما يلي:

```
$x=array ( “index name” => value1, “index2 name” => value2,.....)
$arr=array(“a” => “txt1” , “b” => “txt2” , “c” => “txt3”);
Echo $arr[“a”]; >>>>>>>>>>txt1.
```

- المعامل (+) : يستخدم لجمع المصفوفات في مصفوفة واحدة .
- المعامل (==) : يستخدم لمقارنة تساوي عناصر المصفوفات من حيث ال(**index**) وقيم العناصر في هذه ال (**index**)، دون مراعاة الترتيب والنوع .
- المعامل (===) : يستخدم لمقارنة تساوي عناصر المصفوفات من حيث ال(**index**) وقيم العناصر في هذه ال (**index**)، مع مراعاة الترتيب والعناصر من نفس النوع .

```
$x=array(“a” => “txt1” , “b” => “txt2”);
$y=array(“b” => “txt2”, “a” => “txt1”);
Var_dump($x == $y);>>>>>>>>>>(true)
Var_dump($x === $y);>>>>>>>>>>(false)
```

- المعامل (!=) (< >) : يستخدم لإختبار عدم تساوي عناصر المصفوفات من حيث ال(**index**) وقيم العناصر في هذه ال (**index**)، دون مراعاة الترتيب والنوع .

- **المعامل (==!): يستخدم لإختبار عدم تساوي عناصر المصفوفات من حيث ال(index) وقيم العناصر في هذه ال (index)، مع مراعاة الترتيب والنوع.**

```
$x=array("a" => "txt1" , "b" => "txt2");
$y=array("b" => "txt2", "a" => "txt1");
Var_dump($x != $y);>>>>>>>>>>(False)
Var_dump($x !== $y);>>>>>>>>>>(true)
```