**The array\_map()**

تعد array\_map() وظيفة مدمجة في PHP وتساعد على تعديل جميع العناصر في مصفوفة واحدة أو أكثر وفقًا لبعض الشروط المحددة من قبل المستخدم بطريقة سهلة. يقوم بشكل أساسي بإرسال كل عنصر من عناصر المصفوفة إلى دالة معرفة من قبل المستخدم وإرجاع مصفوفة بقيم جديدة كما تم تعديلها بواسطة تلك الوظيفة

**Array\_reduce()**

يتم استخدام هذه الوظيفة المضمنة في PHP لتقليل عناصر المصفوفة إلى قيمة واحدة يمكن أن تكون ذات قيمة عائمة أو عدد صحيح أو سلسلة. تستخدم الوظيفة وظيفة رد الاتصال المعرفة من قبل المستخدم لتقليل صفيف الإدخال

**Array\_filter()**

يتم استخدام هذه الوظيفة المضمنة في PHP لتصفية عناصر المصفوفة باستخدام وظيفة محددة من قبل المستخدم والتي تسمى أيضًا وظيفة رد الاتصال. تتكرر الدالة array\_filter()‎ على كل قيمة في المصفوفة، وتمررها إلى الدالة المعرفة من قبل المستخدم أو وظيفة رد الاتصال. إذا كانت دالة رد الاتصال ترجع صحيحًا، فسيتم إرجاع القيمة الحالية للمصفوفة إلى مصفوفة النتائج وإلا فلن يتم إرجاعها. بهذه الطريقة يتم الحفاظ على مفاتيح المصفوفة، أي أن مفتاح العنصر في المصفوفة الأصلية ومصفوفة الإخراج متماثلان.

**(…)spread or splat operator**

ضمن دوال php تستخدم الثلاث نقاط لتعريف دوال functions تأخذ عدد متغير من المدخلات (الوسطاء) parameters.

يسهل جمع المصفوفات

**Callable vs Closure**

الفرق الرئيسي بين callable و Closure هو أن callable هو نوع بينما Closure عبارة عن فئة. وظيفة مجهولة هي الإغلاق. يمكن تمرير الإغلاق عندما يكون من المتوقع وجود كائن قابل للاستدعاء، ولكن ليس العكس، لأن فئة الإغلاق تطبق الأسلوب \_\_invoca.

Ex1)

**array\_push: Adds one or more elements to the end of an array**

$fruits = ['apple', 'banana'];

array\_push($fruits, 'cherry', 'date');

Ex2)

**array\_pop: Removes and returns the last element of an array**

$fruits = ['apple', 'banana', 'cherry'];

$lastFruit = array\_pop($fruits);

Ex3)

**array\_shift: Removes and returns the first element of an array**  
$fruits = ['apple', 'banana', 'cherry'];

$firstFruit = array\_shift($fruits);

Ex4)

**array\_unshift: Adds one or more elements to the beginning of an array**

$fruits = ['banana', 'cherry'];

array\_unshift($fruits, 'apple', 'date');

Ex5)

**array\_merge: Combines two or more arrays into a single array**

$fruits1 = ['apple', 'banana'];

$fruits2 = ['cherry', 'date'];

$allFruits = array\_merge($fruits1, $fruits2);

Ex6)

**array\_slice: Extracts a portion of an array based on a start and length**

$fruits = ['apple', 'banana', 'cherry', 'date'];

$slice = array\_slice($fruits, 1, 2);

Ex7)

**array\_reverse: Reverses the order of elements in an array**

$fruits = ['apple', 'banana', 'cherry'];

$reversed = array\_reverse($fruits);

Ex8)

**array\_search: Searches for a value in an array and returns its key if found**

$fruits = ['apple', 'banana', 'cherry'];

$key = array\_search('banana', $fruits);

Ex9)

**array\_filter: Filters elements of an array based on a callback function**

$numbers = [1, 2, 3, 4, 5, 6];

$evenNumbers = array\_filter($numbers, function($num) {

return $num % 2 === 0;

});

Ex10)

**array\_map: Applies a callback function to each element of an array**

$numbers = [1, 2, 3, 4, 5];

$squared = array\_map(function($num) {

return $num \* $num;

}, $numbers);