

Exercise 1: Create an Array

Create an array called \$fruits that contains the following values:
“Apple”, “Banana”, “Cherry”.

Solution:

```
<?php
$fruits = ["Apple", "Banana", "Cherry"];
print_r($fruits);
?>
```

Exercise 2: Add Elements to an Array

Add “Date” and “Elderberry” to the \$fruits array.

Solution:

```
<?php
$fruits[] = "Date";
$fruits[] = "Elderberry";
print_r($fruits);
?>
```

Exercise 3: Remove Elements from an Array

Remove “Banana” from the `$fruits` array.

Solution:

```
<?php
$index = array_search("Banana", $fruits);
if ($index !== false) {
    unset($fruits[$index]);
}
print_r($fruits);
?>
```

Exercise 4: Associative Array

Create an associative array called \$person with the following key-value pairs:

- "name" => "John"
- "age" => 30
- "city" => "New York"

Solution:

```
<?php
$person = [
    "name" => "John",
    "age" => 30,
    "city" => "New York"
];
print_r($person);
?>
```

Exercise 5: Update an Associative Array

Update the age of the person in the \$person array to 31.

Solution:

```
<?php  
$person["age"] = 31;  
print_r($person);  
?>
```

Exercise 6: Iterate Over an Array

Print all the elements in the \$fruits array using a foreach loop.

Solution:

```
<?php
foreach ($fruits as $fruit) {
    echo $fruit . "\n";
}
?>
```

Exercise 7: Multidimensional Array

Create a multidimensional array called \$students that contains the following data:

Name	Age	Grade
Alice	20	A
Bob	22	B
Carol	23	C

Solution:

```
<?php
$students = [
    ["name" => "Alice", "age" => 20, "grade" => "A"],
    ["name" => "Bob", "age" => 22, "grade" => "B"],
    ["name" => "Carol", "age" => 23, "grade" => "C"]
];
print_r($students);
?>
```

Exercise 8: Access Multidimensional Array Elements

Print the grade of the second student (Bob).

Solution:

```
<?php  
echo $students[1] ["grade"] ;  
?>
```

Exercise 9: Sort an Array

Sort the `$fruits` array in alphabetical order.

Solution:

```
<?php  
sort($fruits);  
print_r($fruits);  
?>
```

Exercise 10: Filter an Array

Filter the `$students` array to include only students with a grade of "A".

Solution:

```
<?php
$top_students = array_filter($students, function($student) {
    return $student["grade"] === "A";
});
print_r($top_students);
?>
```

Exercise 11: Array Functions

Use the `array_map` function to create a new array with the ages of the students doubled.

Solution:

```
<?php
$doubled_ages = array_map(function($student) {
    return $student["age"] * 2;
}, $students);
print_r($doubled_ages);
?>
```

Exercise 12: Array Keys and Values

Print all the keys and values of the \$person array.

Solution:

```
<?php
foreach ($person as $key => $value) {
    echo "$key: $value\n";
}
?>
```

Exercise 13: Combine Arrays

Combine two arrays \$names and \$grades into an associative array where the names are the keys and the grades are the values.

```
<?php
$names = ["Alice", "Bob", "Carol"];
$grades = ["A", "B", "C"];
?>
```

Solution:

```
<?php
$names = ["Alice", "Bob", "Carol"];
$grades = ["A", "B", "C"];

$combined = array_combine($names, $grades);
print_r($combined);
?>
```

Exercise 14: Merge Arrays

Merge two arrays \$array1 and \$array2.

```
<?php  
$array1 = ["Apple", "Banana"];  
$array2 = ["Cherry", "Date"];  
?>
```

Solution:

```
<?php
$array1 = ["Apple", "Banana"];
$array2 = ["Cherry", "Date"];

$merged = array_merge($array1, $array2);
print_r($merged);
?>
```

Exercise 15: Find Duplicates

Find and print duplicates in an array \$items.

```
<?php  
$items = ["Apple", "Banana", "Cherry", "Apple", "Date",  
         "Banana"];  
?>
```

Solution:

```
<?php
$itemss = ["Apple", "Banana", "Cherry", "Apple", "Date",
"Banana"] ;

$duplicates = array_unique(array_diff_assoc($itemss,
array_unique($itemss)));
print_r($duplicates);
?>
```