

Chapter 7 Answers

1. Which of the following functions removes the first element from the beginning of an array?

- a. `array_shift()`
- b. `array_unshift()`
- c. `array_push()`
- d. `array_pop()`

2. Explain the easiest way to add elements to the end of an array.

The easiest way to add additional elements to the end of an array is to simply use the array name and brackets syntax

3. Which of the following functions removes the last element from the end of an array? (Choose all that apply.)

- a. `array_shift()`
- b. `array_unshift()`
- c. `array_push()`
- d. `array_pop()`

4. Explain how to use the `array_splice()` function to add and remove elements to and from an array.

To add or remove elements anywhere else in an array, you need to use an array function. PHP includes numerous functions for working with arrays, including the `array_splice()` function, which adds or removes array elements. After adding or removing array elements, the `array_splice()` function also renumbers the indexes in the array. The syntax for the `array_splice()` function is `array_splice(array_name, start, characters_to_delete, values_to_insert);`. The `array_name` argument indicates the name of the array you want to modify. The `start` argument indicates the element within the array at which point elements should be added or removed. The `characters_to_delete` argument is an integer value that indicates the number of elements to remove from the array, starting with the element indicated by the `start` argument. The `values_to_insert` argument represents the values you want to add as new elements to an array.

5. After removing elements from an array, the `unset()` function automatically renumbers the remaining elements. True or False?

6. Which of the following functions removes duplicate elements from an array?

- a. `array_duplicates()`

b. `array_unique()`

c. `remove_duplicates()`

d. `unique()`

7. What is the correct syntax for declaring and initializing an associative array?

a. `$AutoMakers = array("Ford" . "Mustang", "Chevrolet" . "Corvette");`

b. `$AutoMakers = array("Ford"="Mustang", "Chevrolet"="Corvette");`

c. `$AutoMakers = array("Ford">"Mustang", "Chevrolet">"Corvette");`

d. `$AutoMakers = array("Ford"=>"Mustang", "Chevrolet"=>"Corvette");`

8. If an array contains a mixture of indexes and keys, what value or key is used if you do not specify one when adding a new element to the array?

If you create an associative array and then add a new element without specifying a key, PHP automatically assumes that the array is indexed and assigns the new element an index of 0 or the next available integer.

9. If you declare an array in PHP and use a starting index other than 0, empty elements are created for each index between 0 and the index value you specify. True or False?

10. Which of the following functions moves an array's internal pointer to the first element?

a. `first()`

b. `top()`

c. `start()`

d. `reset()`

11. Which of the following functions returns the value of an element where an array's internal pointer is positioned?

a. `current()`

b. `key()`

c. `array()`

d. `array_values()`

12. Explain the difference between the `in_array()` and `array_search()` functions.

You can use the `in_array()` and `array_search()` functions to determine whether a value exists in an array. The `in_array()` function returns a Boolean value of true if a given value exists in an array. The `array_search()` function determines whether a given value exists in an array and returns the index or key of the first matching element if it exists or false if it does not exist. Both functions accept two arguments: The first argument represents the value to search for, whereas the second argument represents the name of the array in which to search.

13. Which of the following locates a key named "Ford" in an array named `$AutoMakers[]`?

- a. `array_key_exists($AutoMakers=>"Ford");`
- b. `$AutoMakers = array_key_exists("Ford");`
- c. `array_key_exists($AutoMakers, "Ford");`
- d. `array_key_exists("Ford", $AutoMakers);`

14. Explain how to use the `array_slice()` function to return a portion of an array and assign it to another array.

You use the `array_slice()` function to return (copy) a portion of an array and assign it to another array. The syntax for the `array_slice()` function is `array_slice(array_name, start, characters_to_return);`. The `array_name` argument indicates the name of the array from which you want to extract elements. The `start` argument indicates the start position within the array to begin extracting elements. The `characters_to_return` argument is an integer value that indicates the number of elements to return from the array, starting with the element indicated by the `start` argument. The syntax for returning a portion of an array with the `array_slice()` function is very similar to the syntax for deleting a portion of an array with the `array_splice()` function. The main difference is that, whereas the `array_splice()` function removes elements, the `array_slice()` function only copies elements to another array.

15. Which of the following functions performs a reverse sort on an array? (Choose all that apply.)

- a. `asort()`
- b. `usort()`
- c. `rsort()`
- d. `krsort()`

16. Which of the following operators can you use to append one array to another? (Choose all that apply.)

- a. `.`
- b. `+`
- c. `+=`
- d. `=>`

17. If you use the `array_merge()` function with indexed arrays, all elements in one array are appended to another array and renumbered. **True** or False?
18. Which of the following returns an array of elements that exist in all of the arrays that are compared?
- a. `usort()`
 - b. `array_common()`
 - c. `array_diff()`
 - d. `array_intersect()`**
19. Explain how to create an associative two-dimensional array using separate statements to build the array and how to create the same array with a single statement.

To create two-dimensional associative arrays, you need to specify the key for each element, as follows:

```
$USDollars = array("U.S. $"=>1, "Yen"=>104.61, "Euro"=>0.7476, "U.K. Pound"=>0.5198, "Canadian $"=>1.2013, "Swiss Franc"=>1.1573);
```

```
$Yen = array("U.S. $"=>0.009559, "Yen"=>1, "Euro"=>0.007146, "U.K. Pound"=>0.004969, "Canadian $"=>0.011484, "Swiss Franc"=>0.011063);
```

```
$Euro = array("U.S. $"=>1.3377, "Yen"=>139.9368, "Euro"=>1, "U.K. Pound"=>0.6953, "Canadian $"=>1.6070, "Swiss Franc"=>1.5481);
```

```
$UKPound = array("U.S. $"=>1.9239, "Yen"=>201.2592, "Euro"=>1.4382, "U.K. Pound"=>1, "Canadian $"=>2.3112, "Swiss Franc"=>2.2265);
```

```
$CanadianDollar = array("U.S. $"=>0.8324, "Yen"=>87.0807, "Euro"=>0.6223, "U.K. Pound"=>0.4327, "Canadian $"=>1, "Swiss Franc"=>0.9634);
```

```
$SwissFranc = array("U.S. $"=>0.8641, "Yen"=>90.3914, "Euro"=>0.6459, "U.K. Pound"=>0.4491, "Canadian $"=>1.0380, "Swiss Franc"=>1);
```

The following example demonstrates how to declare an associative version of the multidimensional `$ExchangeRates[]` array with a single declaration statement:

```
$ExchangeRates = array(
    "U.S. $"=>array("U.S. $"=>1, "Yen"=>104.61, "Euro"=>0.7476,
    "U.K. Pound"=>0.5198, "Canadian $"=>1.2013,
    "Swiss Franc"=>1.1573), "Yen"=>array("U.S. $"=>0.009559, "Yen"=>1,
    "Euro"=>0.007146, "U.K. Pound"=>0.004969, "Canadian $"=>0.011484,
    "Swiss Franc"=>0.011063), "Euro"=>array("U.S. $"=>1.3377,
    "Yen"=>139.9368, "Euro"=>1, "U.K. Pound"=>0.6953,
    "Canadian $"=>1.6070, "Swiss Franc"=>1.5481),
    "U.K. Pound"=>array("U.S. $"=>1.9239, "Yen"=>201.2592,
    "Euro"=>1.4382, "U.K. Pound"=>1, "Canadian $"=>2.3112,
    "Swiss Franc"=>2.2265), "Canadian $"=>array("U.S. $"=>0.8324,
    "Yen"=>87.0807, "Euro"=>0.6223, "U.K. Pound"=>0.4327,
    "Canadian $"=>1, "Swiss Franc"=>0.9634),
    "Swiss Franc"=>array("U.S. $"=>0.8641, "Yen"=>90.3914,
    "Euro"=>0.6459, "U.K. Pound"=>0.4491, "Canadian $"=>1.0380,
    "Swiss Franc"=>1)
);
```

20. Suppose you are working with an indexed two-dimensional array named `$InterestRates[]` that begins with an index of 0. Which of the following refers to the second element in the first dimension and the third element in the second dimension?

a. `$InterestRates[1],[2]`

- b. `$InterestRates[1][2]`
- c. `$InterestRates[1, 2]`
- d. `$InterestRates[1].[2]`

Chapter 8 Answers

1. A flat-file database consists of a single table. **True** or False?
2. Explain how relational databases are organized.

Relational databases consist of one or more related tables. In fact, large relational databases can consist of dozens or hundreds of related tables. Although relational databases can consist of many tables, you create relationships within the database by working with two tables at a time. One table in a relationship is always considered to be the primary table, whereas the other table is considered to be the related table. A primary table is the main table in a relationship that is referenced by another table. A related table (also called a child table) references a primary table in a relational database. Tables in a relationship are connected using primary and foreign keys. A primary key is a field that contains a unique identifier for each record in a primary table. A primary key is a type of index, which identifies records in a database to make retrievals and sorting faster. An index can consist of just a primary key or it can be a combination of multiple fields. A foreign key is a field in a related table that refers to the primary key in a primary table. Primary and foreign keys link records across multiple tables in a relational database.

3. What is the correct term for the individual pieces of information that are stored in a database record?
 - a. element
 - b. field**
 - c. section
 - d. container
4. What is the name of one table's primary key when it is stored in another table? (Choose all that apply.)
 - a. key symbol
 - b. record link
 - c. foreign key**
 - d. unique identifier
5. Breaking tables into multiple related tables to reduce redundant and duplicate information is called _____.
 - a. normalization**

- b. redundancy design
 - c. splitting
 - d. simplification
6. Suppose you have a relational database for a dry cleaning company. Each customer of the dry cleaning company can have multiple items in a cleaning order. What type of relationship is this?
- a. one-to-one
 - b. one-to-many
 - c. many-to-one
 - d. many-to-many
7. _____ has become somewhat of a standard data manipulation language among many database management systems.
- a. Java
 - b. SQL
 - c. ASP.NET
 - d. PERL
8. Files created by different database management systems are completely interchangeable. True or False?
9. What is the default value of the mysql command's -h argument?
- a. database
 - b. mysqlmonitor
 - c. mysqladmin
 - d. localhost
10. What character must terminate SQL commands in the MySQL Monitor?
- a. colon (:)
 - b. semicolon (;)
 - c. ampersand (&)
 - d. period (.)
11. With what characters do you quote identifiers that include special characters?

- a. quotation marks (')
- b. double quotation marks (")
- c. backticks (`)
- d. tildes (~)

11. SQL keywords are case sensitive in the MySQL Monitor. True or False?

12. Explain case sensitivity issues when it comes to file and directory names.

Even though SQL keywords are not case sensitive, the case sensitivity of database and table identifiers depends on your operating system. MySQL stores each database in a directory of the same name as the database identifier. Tables are stored in the database directory in files of the same name as the table identifier. Directory and filenames are not case sensitive on Windows platforms, but are case sensitive on UNIX/Linux systems. This means although you do not need to worry about case sensitivity in database and table names on Windows platforms, you do need to observe letter case when referring to database and table names on UNIX/Linux systems.

13. Which of the following statements displays the available databases in your MySQL installation?

- a. `SHOW DATABASES ;`
- b. `SHOW DATABASES () ;`
- c. `LIST FILES ;`
- d. `GET LIST () ;`

14. What's the first thing you should do after creating a new database?

- a. Save the database.
- b. Restart the MySQL Monitor.
- c. Select the database.
- d. Create a table.

15. Explain the required steps for securing the root account and anonymous account that are initially installed with MySQL.

The anonymous user account allows anyone to log in to MySQL without specifying a username or password, whereas the root account is created without a password. Instead of allowing anyone to access your databases anonymously, you need to delete the anonymous account and assign a password to the root account.

16. Explain what the term "proxy" means in relation to user accounts.

The term proxy refers to someone or something that acts or performs a request for another person.

17. A GRANT statement does not create new user accounts. **True** or False?

18. Explain how to add multiple records to a table by using a single SQL statement.

To add multiple records to a database, you use the LOAD DATA statement with a local text file containing the records you want to add. Place each record in the text file on a separate line and place tabs between each field. The values on each line must be in the same order in which you defined the table fields. For missing field values, use \N instead of NULL; MySQL converts \N character sequences into NULL values when it loads the records.

19. Which of the following keywords performs a reverse sort of database records?

a. DESC

b. REVERSE

c. DESCEND

d. SORTR

20. Which of the following is the correct string for a filter that narrows a recordset to include only records in which the State field is equal to Massachusetts?

a. "WHERE State = 'Massachusetts' "

b. "State = 'Massachusetts' "

c. "WHERE 'State' = Massachusetts "

d. "'State' = 'Massachusetts' "

Chapter 9 Answers

1. MySQL support is enabled in PHP by default. True or **False**?

2. Which of the following functions closes a database connection?

a. close()

b. mysqli_close()

c. mysqli_free()

d. mysqli_free_connect()

3. To which of the following functions do you need to pass a variable representing the database connection? (Choose all that apply.)

a. mysqli_get_client_info()

b. `mysqli_get_host_info()`

c. `mysqli_get_proto_info()`

d. `mysqli_get_server_info()`

4. What is the correct syntax for selecting a database with the `mysqli_select_db()` function?

a. `mysqli_select_db(connection)`

b. `mysqli_select_db(database)`

c. `mysqli_select_db(connection, database)`

d. `database = mysqli_select_db(connection)`

5. Explain the types of errors that can occur when accessing MySQL databases and other types of data sources with PHP.

One of the most important errors that you need to consider occurs when you cannot connect to a database server. Reasons that you may not be able to connect to a database server include the database server is not running, you do not have sufficient privileges to access the data source, or you entered an invalid user and password.

6. The following code structure prevents error messages from printing in the event that the database connection is not available. True or **False**?

```
$DBConnect = mysqli_connect("localhost", "dongosselin",  
    "rosebud", "flightlog");  
  
if (!$DBConnect)  
    echo "<p>The database server is not available.</p>";  
else {  
    echo "<p>Successfully connected to the database server.</p>";  
    mysqli_close($DBConnect);  
}
```

7. Explain the concept of bulletproofing your code.

Although standard error messages that are generated by programming languages such as PHP are very helpful to programmers, they tend to scare users, who might think that they somehow caused the error. Errors can and will occur, but you should never let your users think that they did something wrong. Your goal should be to write code that anticipates any problems that may occur and includes graceful methods of dealing with those problems. Writing code that anticipates and handles potential problems is often called bulletproofing.

8. Which of the following characters suppresses error messages in PHP?

- a. *
- b. &
- c. #
- d. @

9. Which of the following functions terminates script execution? (Choose all that apply.)

- a. `exit()`
- b. `bye()`
- c. `die()`
- d. `quit()`

10. Which of the following functions reports the error message from the last failed database connection attempt?

- a. `mysqli_connect_errno()`
- b. `mysqli_connect_error()`
- c. `mysqli_errno()`
- d. `mysqli_error()`

11. Explain what a result pointer is and how to create and use one.

A result pointer is a special type of variable that refers to the currently selected row in a resultset. The query pointer is a way of keeping track of where you are in a resultset. You assign the result pointer to a variable, which you can use to access the resultset in PHP. The `mysqli_query()` function returns a value of false for any SQL statements that fail, regardless of whether they return results.

12. Which of the following functions returns the fields in the current row of a resultset into an indexed array?

- a. `mysqli_data_fetch()`
- b. `mysqli_data_seek()`
- c. `mysqli_index_row()`
- d. `mysqli_fetch_row()`

13. Which of the following functions returns the fields in the current row of a resultset into an associative array?

- a. `mysqli_assoc_fetch()`
- b. `mysqli_fetch_keys()`
- c. `mysqli_fetch_assoc()`
- d. `mysqli_fetch_index()`

14. Write a simple code segment that demonstrates how to use the `mysqli_num_rows()` and `mysqli_num_fields()` functions to determine whether a SQL query returned results.

```

$SQLstring = "SELECT * FROM inventory";

$queryResult = @mysqli_query($DBConnect, $SQLstring)

    Or die("<p>Unable to execute the query.</p>"

    . "<p>Error code " . mysqli_errno($DBConnect)

    . ": " . mysqli_error($DBConnect)) . "</p>";

echo "<p>Successfully executed the query.</p>";

$numRows = mysqli_num_rows($queryResult);

$numFields = mysqli_num_fields($queryResult);

if ($numRows != 0 && $numFields != 0)

    echo "<p>Your query returned " . mysqli_num_rows($queryResult) .

    " rows and "

    . mysqli_num_fields($queryResult) . " fields.</p>";

else

    echo "<p>Your query returned no results.</p>";

mysqli_close($DBConnect);

```

15. Which of the following functions closes a resultset to ensure that it doesn't keep taking up space in your computer's memory?

- a. `mysqli_free_result()`
- b. `mysqli_result_close()`
- c. `mysqli_free()`
- d. `mysqli_close_result()`

16. Write a simple code segment that demonstrates how to use the `mysqli_db_select()` function to check whether a database exists before you create or delete it.

```
$DBName = "real_estate";

if (@mysqli_select_db($DBConnect, $DBName))

    echo "<p>The $DBName database already exists!</p>";

else {

    $SQLstring = "CREATE DATABASE $DBName";

    $QueryResult = @mysqli_query($DBConnect, $SQLstring)

        Or die("<p>Unable to execute the query.</p>"

            . "<p>Error code " . mysqli_errno($DBConnect)

            . ": " . mysqli_error($DBConnect)) . "</p>";

    echo "<p>Successfully created the database.</p>";

    mysqli_select_db($DBConnect, $DBName)

}

mysqli_close($DBConnect);
```

17. Write a simple code segment that demonstrates how to use a `mysqli_query()` function to prevent your code from attempting to create a table that already exists.

```
$DBName = "real_estate";

...
```

```

$TableName = "commercial";

$SQLstring = "SELECT * FROM $TableName";

$queryResult = @mysqli_query($DBConnect, $SQLstring);

if ($queryResult)

    echo "<p>The $TableName table already exists!</p>";

else {

    $SQLstring = "CREATE TABLE commercial (city VARCHAR(25),

        state VARCHAR(25), sale_or_lease VARCHAR(25),

        type_of_use VARCHAR(40), price INT, size INT)";

    $queryResult = @mysqli_query($DBConnect, $SQLstring)

        Or die("<p>Unable to execute the query.</p>")

        . "<p>Error code " . mysqli_errno($DBConnect)

        . ": " . mysqli_error($DBConnect)) . "</p>";

    echo "<p>Successfully created the table.</p>";

}

mysqli_close($DBConnect);

```

18. Which of the following SQL keywords creates an autoincrementing field?

- a. AUTO
- b. INCREMENT
- c. AUTO_INCREMENT
- d. AUTOINCREMENT

19. Which of the following functions returns the number of rows affected by queries that do not return results, such as INSERT, UPDATE, and DELETE queries?

a. `mysqli_affected_rows()`

b. `mysqli_rows()`

c. `mysqli_get_changed()`

d. `mysqli_fetch_rows()`

20. The ____ function returns the number of operations for various types of actions, depending on the type of query.

a. `mysqli_get_info()`

b. `mysqli_operations()`

c. `mysqli_info()`

d. `mysqli_fetch_actions()`