**TRUE/FALSE**

1. The array\_shift() function adds one or more elements to the beginning of an array.

ANS: F PTS: 1 REF: 315

2. The array\_unshift() function removes the first element from the beginning of an array.

ANS: F PTS: 1 REF: 315

3. The array\_pop() function removes the last element from the end of the array.

ANS: T PTS: 1 REF: 317

4. The array\_push() function adds one or more elements to the end of an array.

ANS: T PTS: 1 REF: 317-318

5. The array\_splice() function adds or removes array elements.

ANS: T PTS: 1 REF: 319

6. Aside from its main function, the array\_splice() function also replaces the indexes in the array.

ANS: F PTS: 1 REF: 319

7. The array\_name argument in the array\_splice() function indicates the name of the array you want to eliminate.

ANS: F PTS: 1 REF: 319

8. You can use the unset() function to remove array elements and other variables.

ANS: T PTS: 1 REF: 321

9. One problem with the unset() function is that is does not renumber the remaining elements in the array.

ANS: T PTS: 1 REF: 322

10. To remove an indexed array’s elements, you need to run the array\_values() function.

ANS: F PTS: 1 REF: 322

11. The array\_unique function creates elements that are unique to an array.

ANS: F PTS: 1 REF: 323

12. The array\_values() function operates directly on an array.

ANS: F PTS: 1 REF: 322

13. As with the array\_values() function, the array\_unique function does not operate directly on an array.

ANS: T PTS: 1 REF: 323

14. PHP creates indexed arrays by default with a starting element of 0.

ANS: T PTS: 1 REF: 325

15. With associative arrays, you can use any alphanumeric keys that you want for the array elements.

ANS: T PTS: 1 REF: 325

16. If you create an associative array and then add a new element with specifying a key, PHP will not assume that the array is indexed.

ANS: F PTS: 1 REF: 326

17. The internal array pointer is a special type of variable that refers to the currently selected elements in an array.

ANS: T PTS: 1 REF: 329

18. The internal array pointer is a way of keeping track of which element you are working with in an array.

ANS: T PTS: 1 REF: 329

19. If you use an iteration function to move the internal array pointer either before the first element or after the last element in an array, the only way to move the array pointer back to a valid element us to use the unset() function.

ANS: F PTS: 1 REF: 331

20. The in\_array() function returns a Boolean value of true if a value exists in an array.

ANS: T PTS: 1 REF: 332

21. The array\_search() function determines whether a given value exists in an array and returns the negative value of the first matching element is it exists or false if it does not exist.

ANS: F PTS: 1 REF: 332

22. The array\_key\_exists() function determines whether a given index or key exists.

ANS: T PTS: 1 REF: 335

23. In the array\_key\_exists() function, you pass two arguments: the first represents the key to search for, and the second represents the name of the array in which to search.

ANS: T PTS: 1 REF: 335

24. Use the array\_keys() function to return an indexed array containing all the keys in an associative array.

ANS: T PTS: 1 REF: 335

25. Use the array\_slice() function to return or copy a portion of an array and assign it to another array.

ANS: T PTS: 1 REF: 336

26. The most commonly used array sorting functions are sort() and rsort() for associative arrays.

ANS: F PTS: 1 REF: 338

27. To append one array to another use the append() function.

ANS: F PTS: 1 REF: 342

28. Instead of appending one array to another you can merge two or more arrays with the array\_merge() function.

ANS: T PTS: 1 REF: 343

29. You can create a new associative array that uses the values from one array as keys and values from another array as element values.

ANS: T PTS: 1 REF: 344

30. The array\_diff() function returns an array of elements that exist in one array but not in any other arrays to which it is compared.

ANS: T PTS: 1 REF: 344

**MULTIPLE CHOICE**

1. The array\_shift() function removes the \_\_\_\_ element from the beginning of an array.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | last | c. | second |
| b. | first | d. | middle |

ANS: B PTS: 1 REF: 315

2. The array\_unshift() function adds \_\_\_\_ element(s) to the beginning of the array.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | one or more | c. | no new |
| b. | two or more | d. | only one |

ANS: A PTS: 1 REF: 315

3. The array\_pop() function \_\_\_\_ from the end of the array.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | moves an element | c. | deletes the first element |
| b. | copies an element | d. | removes the last element |

ANS: D PTS: 1 REF: 317

4. The array\_push \_\_\_\_ one or more elements to the end of the array.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | adds | c. | moves |
| b. | deletes | d. | pushes |

ANS: A PTS: 1 REF: 317-318

5. The \_\_\_\_ function lets you add or remove elements anywhere else in the array.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | array\_add() | c. | array\_splice() |
| b. | array\_split() | d. | array\_values() |

ANS: C PTS: 1 REF: 319

6. The \_\_\_\_ argument of the array\_splice() function indicates the name of the array you want to modify.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | array\_name | c. | array\_function |
| b. | array\_value | d. | array\_mod |

ANS: A PTS: 1 REF: 319

7. To remove multiple elements using the unset() function, separate each \_\_\_\_ and element with commas.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | variable | c. | index name |
| b. | array | d. | argument |

ANS: C PTS: 1 REF: 321

8. To renumber an indexed array’s elements, you need to run the \_\_\_\_ function.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | unset() | c. | array() |
| b. | array\_values() | d. | array\_num() |

ANS: B PTS: 1 REF: 322

9. The \_\_\_\_ function removes duplicate elements from an array.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | array\_dup() | c. | array\_unique() |
| b. | array\_values() | d. | array() |

ANS: C PTS: 1 REF: 323

10. The syntax for declaring and initializing an associative array is \_\_\_\_.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | $array\_name=array(key=>value, ...); | c. | $array\_name=array(key=value, ...); |
| b. | $array\_name=array(key=<value, ...); | d. | $array\_name=array(key==value, ...); |

ANS: A PTS: 1 REF: 325

11. The \_\_\_\_ is a special type of variable that refers to the currently selected element in an array.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | external array pointer | c. | internal array pointer |
| b. | external element pointer | d. | internal element pointer |

ANS: C PTS: 1 REF: 329

12. The \_\_\_\_ function returns a Boolean value of true if a given value exists in the array.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | array\_search() | c. | array() |
| b. | array\_boole() | d. | in\_array() |

ANS: D PTS: 1 REF: 332

13. The \_\_\_\_ function determines whether a given value exists in an array.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | array\_search() | c. | find\_array() |
| b. | in\_array() | d. | array\_values() |

ANS: A PTS: 1 REF: 332

14. The \_\_\_\_ function determines whether a given index or key exists.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | array\_value() | c. | array\_key() |
| b. | array\_key\_exists() | d. | array\_set() |

ANS: B PTS: 1 REF: 335

15. The \_\_\_\_ function returns an indexed array containing all the keys in an associative array.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | array\_key\_exists() | c. | array\_find() |
| b. | array\_search() | d. | array\_keys() |

ANS: D PTS: 1 REF: 335

16. You use the \_\_\_\_ function to return a portion of an array and assign it to another array.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | array\_slice() | c. | array\_assign() |
| b. | array\_set() | d. | array\_key() |

ANS: A PTS: 1 REF: 336

17. To sort an associative array by key, use the \_\_\_\_ function.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | natsort() | c. | rsort() |
| b. | ksort() | d. | asort() |

ANS: B PTS: 1 REF: 337

18. One of the most commonly used array sorting functions is the \_\_\_\_ function for indexed arrays.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | ksort() | c. | uk\_sort() |
| b. | krsort() | d. | sort() |

ANS: D PTS: 1 REF: 338

19. The symbol \_\_\_\_ appends one array to another.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | \* | c. | + |
| b. | = = | d. | && |

ANS: C PTS: 1 REF: 342

20. Merge two or more arrays with the \_\_\_\_ function.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | array\_merge() | c. | array\_paste() |
| b. | array\_plus | d. | array\_values() |

ANS: A PTS: 1 REF: 343

21. The \_\_\_\_ function returns an array of elements that exist in one array but not in any other arrays to which it is compared.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | array\_compare() | c. | array\_search() |
| b. | array\_diff() | d. | array\_match() |

ANS: B PTS: 1 REF: 344

**COMPLETION**

1. The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ function removes the first element from the beginning of an array.

ANS: array\_shift()

PTS: 1 REF: 315

2. You pass to the array\_shift() function the name of the array whose first element you want to \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

ANS: remove

PTS: 1 REF: 315

3. The array\_pop() function removes the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ from the end of the array.

ANS: last elements

PTS: 1 REF: 317

4. The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ function adds one or more elements to the end of an array.

ANS: array\_push()

PTS: 1 REF: 317-318

5. The syntax for the array\_splice() function is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

ANS: array\_splice(array\_name, start, characters\_to\_delete, values\_to\_insert);

PTS: 1 REF: 319

6. The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ function removes array elements and other variables.

ANS: unset()

PTS: 1 REF: 321

7. You pass the unset() function the array name and the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ of the element you want to remove.

ANS: index number

PTS: 1 REF: 321

8. One problem with the unset() function is that it does not \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ the remaining elements in the array.

ANS: renumber

PTS: 1 REF: 322

9. Run the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ function to renumber an indexed array’s element.

ANS: array\_values()

PTS: 1 REF: 322

10. The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ function does not operate directly on an array; but, returns a new array with the renumbered indexes.

ANS: array\_values()

PTS: 1 REF: 322

11. The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ function removes duplicate elements from an array.

ANS: array\_unique()

PTS: 1 REF: 323

12. As with the array\_values() function the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ function does not operate directly on an array.

ANS: array\_unique()

PTS: 1 REF: 323

13. With \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, you can use any alphanumeric key that you want for the array elements.

ANS: associative arrays

PTS: 1 REF: 325

14. The syntax for declaring and initializing an associative array is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

ANS: $array\_name = array(key=>value, ....);

PTS: 1 REF: 325

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

ANS: indexed

PTS: 1 REF: 326

16. The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is a special type of variable that refers to the currently selected element in an array.

ANS: internal array pointer

PTS: 1 REF: 329

17. The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ function returns a Boolean value of true if a given value exists in an array.

ANS: in\_array()

PTS: 1 REF: 332

18. The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 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.

ANS: array\_search()

PTS: 1 REF: 332

19. The array\_key\_exists() function determines whether a given \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ exists.

ANS:

index or key

PTS: 1 REF: 335

20. The array\_slice() function \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ a portion of an array and assigns it to another array.

ANS: returns or copies

PTS: 1 REF: 336

21. The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ sorts an associative array by key

ANS: ksort()

PTS: 1 REF: 337

22. The most commonly used array sorting functions are \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and rsort() for indexed arrays.

ANS: sort()

PTS: 1 REF: 338

23. To append one array to another, you use the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ or the compound assignment operator (+=).

ANS: addition

PTS: 1 REF: 342

24. Instead of appending one array to another, you can merge two or more arrays with the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ function.

ANS: array\_merge()

PTS: 1 REF: 343

**ESSAY**

1. What functions can you use to compare the contents of two or more arrays?

ANS:

PHP includes several functions for comparing the contents of two or more arrays. Two of the most basic comparison functions are array\_diff() and array\_intersect().

PTS: 1 REF: 344-345