webgurukul

सर्व हिताय शिक्षा

PHP

Question Set

Index

18	Object Oriented Programming	y
19	Database	y

PHP Operators

Arithmetic Operators:

1. Write a PHP program that calculates the sum of two numbers.

2. Create a program that calculates the product of two numbers.

3. Implement a program that calculates the quotient and remainder of a division operation.

4. Write a PHP script that calculates the square of a given number.

5. Create a program that calculates the area of a rectangle given its length and width.

Assignment Operators:

6. Write a PHP script that increments a variable by 5 using the increment assignment operator.

7. Create a program that decrements a variable by 3 using the decrement assignment operator.

8. Implement a program that calculates the square of a number using the assignment operator.

9. Write a PHP script that appends a string to an existing string variable using the assignment operator.

10. Create a program that combines and assigns two arrays using the assignment operator.

Comparison Operators:

11. Create a program that compares two numbers and determines if they are equal.

12. Write a PHP script that checks if a number is greater than another number.

13. Implement a program that checks if a user's age is equal to or greater than 18.

14. Write a PHP program that checks if a string is equal to another string (case-sensitive).

15. Create a program that checks if a string is not equal to another string (case-insensitive).

Logical Operators:

16. Write a PHP script that checks if a number is both even and positive.

17. Create a program that checks if a user's age is between 18 and 30.

18. Implement a program that checks if a user is either an administrator or a moderator.

19. Write a PHP program that checks if a user is a customer and not an administrator.

20. Create a program that checks if a number is divisible by both 2 and 3.

Bitwise Operators:

21. Implement a program that performs a bitwise AND operation on two numbers.

22. Write a PHP script that performs a bitwise OR operation on two numbers.

23. Create a program that performs a bitwise XOR operation on two numbers.

24. Implement a program that shifts a binary number left by one position.

25. Write a PHP program that checks if a specific bit is set in a binary number.

Ternary Operator:

26. Write a PHP script that determines if a user's age is greater than or equal to 18 and returns "Adult" or "Minor" accordingly.

27. Create a program that calculates and returns the absolute value of a number using the ternary operator.

28. Implement a program that checks if a number is odd or even and returns the result as a string.

29. Write a PHP program that checks if a user has administrator privileges and returns "Admin" or "User."

30. Create a program that calculates and returns the discount amount (10% or 20%) based on a user's membership status using the ternary operator.

PHP If-Else

1.Write a PHP program that checks if a number is even or odd and displays the result.

2.Create a program that checks if a user input string is empty and provides an appropriate message.

3.Write a PHP script that determines whether a user is eligible to vote based on their age.

4.Create a program that checks if a given number is positive, negative, or zero.

5.Implement a program that checks if a year is a leap year or not.

6.Write a PHP script that calculates the discount for a shopping cart based on the total price.

7.Create a program that determines if a triangle is equilateral, isosceles, or scalene based on user-provided side lengths.

8.Implement a grading system that converts a numerical grade to a letter grade (A, B, C, etc.).

9.Write a PHP script that determines if a given word is a palindrome (reads the same forwards and backwards).

10.Create a program that checks if a user-provided number is a prime number.

11.Write a PHP program that checks if a number is both even and a multiple of 5.

12.Create a program that checks if a user-provided year is a leap year and a multiple of 10.

13.Implement a program that checks if a user is eligible for a discount based on their age and membership status.

14.Write a PHP script that checks if a given number is positive, less than 100, and even.

15.Create a program that checks if a user input string contains both numbers and letters.

16.Create a program that determines the season (spring, summer, fall, or winter) based on a user-provided month.

17.Write a PHP script that checks if a number is divisible by both 3 and 5, or just one of them.

18.Implement a program that classifies a user's BMI as underweight, normal weight, overweight, or obese based on their height and weight.

19.Create a program that determines the type of a triangle (equilateral, isosceles, scalene, or not a triangle) based on user-provided side lengths.

20.Write a PHP script that checks if a user-provided year is a leap year, and if so, which month of the year February has 29 days.

PHP Switch Case

1. Write a PHP program that takes a number (1 to 7) and displays the corresponding day of the week using a switch-case statement.

2. Create a program that takes a month number (1 to 12) and displays the number of days in that month using a switch-case statement.

3. Implement a program that takes a grade (A, B, C, D, or F) and provides a message based on the grade using a switch-case statement.

4.Write a PHP script that takes a user's choice (rock, paper, or scissors) and determines the winner in a rock-paper-scissors game using a switch-case statement.

5. Implement a program that takes an operator (+, -, *, /) and two numbers, and performs the corresponding arithmetic operation using a switch-case statement.

6.Write a PHP program that takes a number (1 to 12) and displays the season (spring, summer, fall, or winter) using a switch-case statement with fallthrough.

7. Create a program that takes a user's age and classifies them into age groups (child, teenager, adult, senior) using a switch-case statement with fallthrough.

8. Create a program that takes a product category (electronics, clothing, or books) and calculates the shipping cost based on the category using a switch-case statement.

9. Implement a program that takes a user's occupation and provides a customized greeting based on the occupation using a switch-case statement with fallthrough.

10. Write a PHP script that takes a year and a month and displays the number of days in that month, considering leap years, using a nested switch-case statement.

PHP Loops

1. Create a script that displays 1-2-3-4-5-6-7-8-9-10 on one line. There will be no hyphen(-) at starting and ending position.

2. Create a script using a for loop to add all the integers between 0 and 30 and display the total.

3. Create a script to construct the following pattern, using nested for loop.

```
*
* *
* * *
* * * *
* * * * *
```

4. Create a script to construct the following pattern, using a nested for loop.

```
*
* *
* * *
* * * *
* * * * *
* * * * *
* * * *
* * *
* *
*
```

5. Write a program to calculate and print the factorial of a number using a for loop. The factorial of a number is the product of all integers up to and including that number, so the factorial of 4 is 4*3*2*1= 24.

6. Write a program which will give you all of the potential combinations of a two-digit decimal combination, printed in a comma delimited format :
Sample output :

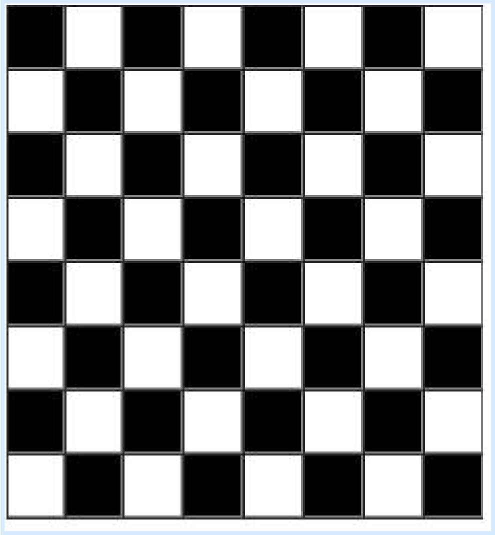
```
00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59,
60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79,
80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99,
```

7. Write a program which will count the "r" characters in the text "w3resource".

8. Write a PHP script that creates the following table using for loops. Add cellpadding="3px" and cellspacing="0px" to the table tag.

1 * 1 = 1	1 * 2 = 2	1 * 3 = 3	1 * 4 = 4	1 * 5 = 5
2 * 1 = 2	2 * 2 = 4	2 * 3 = 6	2 * 4 = 8	2 * 5 = 10
3 * 1 = 3	3 * 2 = 6	3 * 3 = 9	3 * 4 = 12	3 * 5 = 15
4 * 1 = 4	4 * 2 = 8	4 * 3 = 12	4 * 4 = 16	4 * 5 = 20
5 * 1 = 5	5 * 2 = 10	5 * 3 = 15	5 * 4 = 20	5 * 5 = 25
6 * 1 = 6	6 * 2 = 12	6 * 3 = 18	6 * 4 = 24	6 * 5 = 30

9. Write a PHP script using nested for loop that creates a chess board as shown below.
Use table width="270px" and take 30px as cell height and width.



10. Write a PHP script that creates the following table (use for loops).

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40

```
5     10    15    20    25    30    35    40    45    50
6     12    18    24    30    36    42    48    54    60
7     14    21    28    35    42    49    56    63    70
8     16    24    32    40    48    56    64    72    80
9     18    27    36    45    54    63    72    81    90
10    20    30    40    50    60    70    80    90    100
```

11. Write a PHP program which iterates the integers from 1 to 50. For multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

12. Write a PHP program to generate and display the first n lines of a Floyd triangle. (use n=5 and n=11 rows).
According to Wikipedia Floyd's triangle is a right-angled triangular array of natural numbers, used in computer science education. It is named after Robert Floyd. It is defined by filling the rows of the triangle with consecutive numbers, starting with a 1 in the top left corner:
Sample output for n = 5 :

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

13. Write a PHP program to print alphabet pattern 'A'.
Expected Output:

```
  ***
 *   *
 *   *
 *****
 *   *
 *   *
 *   *
 *   *
```

14. Write a PHP program to print alphabet pattern 'B'.
Expected Output:

```
****
*    *
*    *
****
*    *
*    *
****
```

15. Write a PHP program to print alphabet pattern 'C'.
Expected Output:

```
 ***
*    *
*
*
*
*    *
 ***
```

16. Write a PHP program to print alphabet pattern 'D'.
Expected Output:

```
****
*    *
*    *
*    *
*    *
*    *
****
```

17. Write a PHP program to print alphabet pattern 'E'.
Expected Output:

```
*****
*
*
****
*
*
*****
```

18. Write a PHP program to print alphabet pattern 'F'.
Expected Output:

```
*****
*
*
****
*
*
*
```

19. Write a PHP program to print alphabet pattern 'G'.
Expected Output:

```
 ***
*   *
*
* ***
*   *
*   *
 ***
```

20. Write a PHP program to print alphabet pattern 'H'.
Expected Output:

```
*   *
*   *
*   *
*****
*   *
*   *
*   *
```

21. Write a PHP program to print alphabet pattern 'I'.
Expected Output:

```
*****
    *
    *
    *
    *
    *
*****
```

22. Write a PHP program to print alphabet pattern 'J'.
Expected Output:

```
   ***
     *
     *
     *
     *
 *  *
   *
```

23. Write a PHP program to print alphabet pattern 'K'.
Expected Output:

```
*      *
*    *
*   *
**
* *
*  *
*   *
*    *
```

24. Write a PHP program to print alphabet pattern 'L'.
Expected Output:

```
*
*
*
*
*
*
*****
```

25. Write a PHP program to print alphabet pattern 'M'.
Expected Output:

```
*    *
*    *
* *  **
*  *  *
*    *
*    *
*    *
```

26. Write a PHP program to print alphabet pattern 'N'.
Expected Output:

```
*   *
*   *
**  *
* * *
*  **
*   *
*   *
```

27. Write a PHP program to print alphabet pattern 'O'.
Expected Output:

```
 ***
*   *
*   *
*   *
*   *
*   *
 ***
```

28. Write a PHP program to print alphabet pattern 'P'.
Expected Output:

```
****
*   *
*   *
****
*
*
*
```

29. Write a PHP program to print alphabet pattern 'Q'.
Expected Output:

```
 ***
*   *
*   *
*   *
* * *
*  *
 ** *
```

30. Write a PHP program to print alphabet pattern 'R'.
Expected Output:

```
****
*   *
*   *
****
* *
*  *
*   *
```

31. Write a PHP program to print alphabet pattern 'S'.
Expected Output:

```
 ****
*
*
 ***
    *
    *
****
```

32. Write a PHP program to print alphabet pattern 'T'.
Expected Output:

```
*****
  *
  *
  *
  *
  *
```

33. Write a PHP program to print alphabet pattern 'U'.
Expected Output:

```
*   *
*   *
*   *
*   *
*   *
*   *
 ***
```

34. Write a PHP program to print alphabet pattern 'V'.
Expected Output:

```
*     *
*     *
*     *
*     *
*     *
 *   *
   *
```

35. Write a PHP program to print alphabet pattern 'W'.
Expected Output:

```
*    *
*    *
*    *
*    *
*  * *
*  * *
 *  *
```

36. Write a PHP program to print alphabet pattern 'X'.
Expected Output:

```
*   *
*   *
 * *
  *
 * *
*   *
*   *
```

37. Write a PHP program to print alphabet pattern 'Y'.
Expected Output:

```
 *   *
 *   *
  * *
   *
   *
   *
   *
```

38. Write a PHP program to print alphabet pattern 'Z'.
Expected Output:

```
*******
     *
    *
   *
  *
 *
*******
```

PHP String

1.Write a PHP script to : -
a) transform a string all uppercase letters.
b) transform a string all lowercase letters.
c) make a string's first character uppercase.
d) make a string's first character of all the words uppercase.

2. Write a PHP script to split the following string.
Sample string : '082307'
Expected Output : 08:23:07

3. Write a PHP script to check whether a string contains a specific string?
Sample string : 'The quick brown fox jumps over the lazy dog.'
Check whether the said string contains the string 'jumps'.

4. Write a PHP script to convert the value of a PHP variable to string.

5. Write a PHP script to extract the file name from the following string.
Sample String : 'www.example.com/public_html/index.php'
Expected Output : 'index.php'

6. Write a PHP script to extract the user name from the following email ID.
Sample String : 'rayy@example.com'
Expected Output : 'rayy'

7. Write a PHP script to get the last three characters of a string.
Sample String : 'rayy@example.com'
Expected Output : 'com'

8. Write a PHP script to format values in currency style.
Sample values : value1 = 65.45, value2 = 104.35
Expected Result : 169.80

9. Write a PHP script to generate simple random password [do not use rand() function] from a given string.

Sample string : '1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZabcefghijklmnopqrstuvwxyz'

Note : Password length may be 6, 7, 8 etc.

10. Write a PHP script to replace the first 'the' of the following string with 'That'.

Sample date : 'the quick brown fox jumps over the lazy dog.'

Expected Result : That quick brown fox jumps over the lazy dog.

11. Write a PHP script to find the first character that is different between two strings.

String1 : 'football'

String2 : 'footboll'

Expected Result : First difference between two strings at position 5: "a" vs "o"

12. Write a PHP script to put a string in an array.

Sample strings : "Twinkle, twinkle, little star,\nHow I wonder what you are.\nUp above the world so high,\nLike a diamond in the sky.";

Expected Result (using var_dump()) : array(4) { [0]=> string(30) "Twinkle, twinkle, little star," [1]=> string(26) "How I wonder what you are." [2]=> string(27) "Up above the world so high," [3]=> string(26) "Like a diamond in the sky." }

13. Write a PHP script to get the filename component of the following path.

Sample path : "https://www.w3resource.com/index.php"

Expected Output : 'index'

14. Write a PHP script to print the next character of a specific character.

Sample character : 'a'

Expected Output : 'b'

Sample character : 'z'

Expected Output : 'a'

15. Write a PHP script to remove a part of a string from the beginning.
Sample string : 'rayy@example.com'
Expected Output : 'example.com'

16. Write a PHP script to get a hex dump of a string.
Sample string : 'rayy@example.com'

17. Write a PHP script to insert a string at the specified position in a given string.
Original String : 'The brown fox'
Insert 'quick' between 'The' and 'brown'.
Expected Output : 'The quick brown fox'

18. Write a PHP script to get the first word of a sentence.
Original String : 'The quick brown fox'
Expected Output : 'The'

19. Write a PHP script to remove all leading zeroes from a string.
Original String : '000547023.24'
Expected Output : '547023.24'

20. Write a PHP script to remove part of a string.
Original String : 'The quick brown fox jumps over the lazy dog'
Remove 'fox' from the above string.
Expected Output : 'The quick brown jumps over the lazy dog'

21. Write a PHP script to remove trailing slash from a string.
Original String : 'The quick brown fox jumps over the lazy dog///'
Expected Output : 'The quick brown fox jumps over the lazy dog'

22. Write a PHP script to get the characters after the last '/' in an url.
Sample URL : 'http://www.example.com/5478631'
Expected Output : '5478631'

23. Write a PHP script to replace multiple characters from the following string.
Sample String : '\"\1+2/3*2:2-3/4*3'
Expected Output : '1 2 3 2 2 3 4 3'

24. Write a PHP script to select first 5 words from the following string.
Sample String : 'The quick brown fox jumps over the lazy dog'
Expected Output : 'The quick brown fox jumps'

25. Write a PHP script to remove comma(s) from the following numeric string.
Sample String : '2,543.12'
Expected Output : 2543.12

26. Write a PHP script to print letters from 'a' to 'z'.
Expected Result : abcdefghijklmnopqrstuvwxyz

PHP Function

1. Write a PHP function that takes two numbers as arguments and returns their sum.

2. Create a function that takes a string and returns its length.

3. Implement a function that checks if a number is even and returns a boolean result.

4. Write a function that calculates the area of a rectangle when given its length and width.

5. Create a function that takes a name and returns a customized greeting.

6. Write a PHP function that takes an array of numbers and returns their average.

7. Create a function that calculates the volume of a cube when given the side length as an argument.

8. Implement a function that takes a string and a number and repeats the string that many times.

9. Write a function that takes two numbers and a mathematical operation (add, subtract, multiply, divide) and returns the result.

10. Create a function that takes an array of strings and a separator, and returns a concatenated string with the separator in between.

11. Create a function that demonstrates variable scope by declaring a variable inside the function and trying to access it outside the function.

12. Implement a function that modifies a global variable from within the function.

13. Write a function that uses static variables to keep track of the number of times it's called.
14. Create a function that takes an array as a parameter, sorts it in descending order, and returns the sorted array.

15. Write a function that returns the sum of all even numbers from 1 to 100 using a loop.

16. Implement a function that generates a random password of a specified length.

17. Create a function that calculates the factorial of a number using an iterative approach (without recursion).

18. Write a function that checks if a given string is a palindrome.

19. Implement a function that validates an email address using regular expressions.

20. Create a function that takes a sentence and returns an array of unique words in the sentence.

21. Write a function that takes an array and a callback function to apply the callback to each element of the array.

22. Implement a function that filters an array based on a callback function (e.g., return all even numbers).

23. Create a function that sorts an array in ascending order using a callback function.

24. Write a function that calculates the total price of items in a shopping cart, allowing for custom discount calculations using a callback function.

25. Implement a function that generates an HTML list from an array of items, allowing for custom item formatting using a callback function.

PHP Recursive Functions

1. Write a recursive function to calculate the factorial of a number.

2. Create a recursive function to compute the sum of all natural numbers from 1 to n.

3. Implement a recursive function to find the nth Fibonacci number.

4. Write a recursive function to calculate the sum of digits in a number.

5. Create a recursive function to reverse a string.

6. Implement a recursive function to compute the sum of elements in an array.

7. Write a recursive function to find the maximum value in an array.

8. Create a recursive function to check if an element exists in an array.

9. Implement a recursive function to flatten a nested array (e.g., `[1, [2, [3, 4]]]` becomes `[1, 2, 3, 4]`).

10. Write a recursive function to find the factorial of each element in an array and return the result as a new array.

11. Create a recursive function to traverse and print the elements of a binary tree.

12. Implement a recursive function to check if a given word is a palindrome.

13. Write a recursive function to compute the height of a binary tree.

14. Implement a recursive function to search for an element in a linked list.

15. Create a recursive function to generate all possible combinations of elements from an array.

PHP Date function

1. Write a PHP script which will display the copyright information in the following format. To get current year you can use the date() function.
Expected Format : © 2013 PHP Exercises - w3resource

2. Create a simple 'birthday countdown' script, the script will count the number of days between current day and birthday.

3. Write a PHP script to print the current date in the following format. To get current date's information you can use the date() function.
Sample format : (assume current date is September 01, 2013)
2013/09/01
13.09.01
01-09-13

4. Write a PHP script to calculate the difference between two dates.
Sample dates : 1981-11-04, 2013-09-04
Expected Result : 31 years, 10 months, 11 days

5. Write a PHP script to convert a date from yyyy-mm-dd to dd-mm-yyyy.
Sample date : 2012-09-12
Expected Result : 12-09-2012

6. Write a PHP script to convert the date to timestamp.
Sample date : 12-05-2014
Expected Result : 1399852800

7. Write a PHP script to calculate a number of days between two dates.

8. Write a PHP script to get the first and last day of a month from a specified date.

9. Write a PHP script to print like : Saturday the 7th.

10. Write a PHP script to check whether the given dates are valid or not?

11. Write a PHP script to get time difference in days and years, months, days, hours, minutes, seconds between two dates.
Note : Use DateTime class.

12. Write a PHP script to change month number to month name.

13. Write a PHP script to get yesterday's date.

14. Write a PHP script to get the current date/time of 'Australia/Melbourne'.

15. Write a PHP script to check whether a date is a weekend or not.

16. Write a PHP script to add/subtract the number of days from a particular date.
Sample Output : Original date : 2011-01-01
Before 40 days : 2010-11-22
After 40 days : 2011-02-10

17. Write a PHP function to get start and end date of a week (by week number) of a particular year.
Sample week and year : 12, 2014
Output :
Starting date of the week: 2014-3-24
End date the week: 2014-3-30

18. Write a PHP script to calculate the current age of a person.
Sample date of birth : 11.4.1987
Output : Your age : 27 years, 1 month, 29 days

19. Write a PHP script to calculate weeks between two dates.
Sample Output : Weeks between 1/1/2014 and 12/31/2014 is 52

20. Write a PHP script to get the number of the month before the current month.

21. Write a PHP script to convert seconds into days, hours, minutes and seconds.
Sample seconds : 200000
Expected output : 2 days, 7 hours, 33 minutes and 20 second

22. Write a PHP script to get the last 6 months from the current month.

23. Write a PHP script to get the current month and previous three months.

24. Write a PHP script to increment date by one month
Sample date : 2012-12-21
Expected Output : 2013-01-21

25. Write a PHP script to get the current date in Italian.
Sample Output : Today is lun on ott 06, 2014

26. Write a PHP script to convert the number to month name.

27. Write a PHP script to get the number of days of the current month.

28. Write a PHP script to display time in a specified timezone.

PHP Math Function

1. Write a PHP script to find the maximum and minimum marks from the following set of arrays.
Sample arrays :
$marks1 = array(360,310,310,330,313,375,456,111,256);
$marks2 = array(350,340,356,330,321);
$marks3 = array(630,340,570,635,434,255,298);
Expected Output :
Maximum marks : 635
Minimum marks : 111

2. Write a PHP script which rounds the following values with 1 decimal digit precision.
Sample values and Output :
1.65 --> 1.7
1.65 --> 1.6
-1.54 --> -1.5

3. Write a PHP script to generate random 11 characters string of letters and numbers.

4. Write a PHP script to convert scientific notation to an int and a float.
Sample scientific notation : 4.5e3
Expected Output : 4 & 4500

5. Write a PHP script to convert a date from yyyy-mm-dd to dd-mm-yyyy.
Sample floating value : 0.0456
Expected Output :
Exponent part : -4
Mantissa part : 0.7296

6. Write a PHP script to get the information regarding memory usage in KB or MB etc.

7. Find earliest and latest dates from a list of dates.

8. Write a PHP function to round a float away from zero to a specified number of decimal places.
Sample Data :

```
(78.78001, 2)
(8.131001, 2)
(0.586001, 4)
(-.125481, 3)
-.125481
```

Sample Output :

```
78.79
8.14
0.5861
-0.126
-1
```

9. Write a PHP function to convert Arabic Numbers to Roman Numerals.

10. Write a PHP function to get random float numbers.

11. Write a PHP function to create a human-readable random string for a captcha.

Sample Output :

```
hoboh
tynzh
```

12. Write a PHP function to get the distance between two points on the earth.

PHP JSON

1. Reading JSON Data

Read and display the content of any JSON file using PHP.

2. Parsing JSON Data

Read a JSON string, decode it into a PHP array, and then print a specific value from the array.

3. Modifying JSON Data

Read JSON data from a file, update a specific field in one of the records, and save the updated JSON back to the file.

4. Filtering JSON Data

Read JSON data from a file, filter and display records that match a specific condition (e.g., all products with a price below $50).

5. Adding Records to JSON

Read JSON data from a file, add a new record to the JSON, and save the updated JSON back to the file.

6. Searching in JSON Data

Read JSON data from a file, search for a specific record by a given property (e.g., searching for a product by name) and display the result.

7. Deleting Records from JSON

Read JSON data from a file, delete a record with a specific ID, and save the updated JSON back to the file.

8. Sorting JSON Data

Read JSON data from a file, sort the records by a specific property (e.g., by price or date), and display the sorted result.

9. Aggregating Data from JSON

Read JSON data from a file and calculate some aggregate statistics (e.g., the total sum of all product prices).

10. Complex JSON Operations

Read JSON data from a file, perform complex operations like joining data from multiple sources, and display the result (e.g., merging product details with user reviews).

11. Write a PHP script to decode a JSON string.

12. Write a PHP script to decode large integers.
Sample integer :
{"number": 123456789012345678901234567890}

13. Write a PHP script to get JSON representation of a value from an array.

14. Write a PHP function to display JSON decode errors.

PHP Searching and Sorting

1. Write a PHP program to sort a list of elements using Quick sort.
Quick sort is a comparison sort, meaning that it can sort items of any type for which a "less-than" relation (formally, a total order) is defined.

2. Write a PHP program to sort a list of elements using Heap sort.
In computer science, heapsort (invented by J. W. J. Williams in 1964) is a comparison-based sorting algorithm. Heapsort can be thought of as an improved selection sort: like that algorithm, it divides its input into a sorted and an unsorted region, and it interactively shrinks the unsorted region by extracting the largest element and moving that to the sorted region. The improvement consists of the use of a heap data structure rather than a linear-time search to find the maximum. Although somewhat slower in practice on most machines than a well-implemented quicksort, it has the advantage of a more favorable worst-case O(n log n) runtime. Heapsort is an in-place algorithm, but it is not a stable sort.
A run of the heapsort algorithm sorting an array of randomly permuted values. In the first stage of the algorithm, the array elements are reordered to satisfy the heap property. Before the actual sorting takes place, the heap tree structure is shown briefly for illustration.

3. Write a PHP program to sort a list of elements using Insertion sort.
Insertion sort is a simple sorting algorithm that builds the final sorted array (or list) one item at a time. It is much less efficient on large lists than more advanced algorithms such as quicksort, heapsort, or merge sort.

4. Write a PHP program to sort a list of elements using Selection sort.
The selection sort improves on the bubble sort by making only one exchange for every pass through the list.

5. Write a PHP program to sort a list of elements using Shell sort.
According to Wikipedia "Shell sort or Shell's method, is an in-place comparison sort. It can be seen as either a generalization of sorting by exchange (bubble sort) or sorting by insertion (insertion sort). The method starts by sorting pairs of elements far apart from each other, then progressively reducing the gap between elements to be compared. Starting with far apart elements can move some out-of-place elements into position faster than a simple nearest neighbor exchange."

6. Write a PHP program to sort a list of elements using Bubble sort.
According to Wikipedia "Bubble sort, sometimes referred to as sinking sort, is a simple sorting algorithm that repeatedly steps through the list to be sorted, compares each pair of adjacent items and swaps them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted. The algorithm, which is a comparison sort, is named for the way smaller elements "bubble" to the top of the list. Although the algorithm is simple, it is too slow and impractical for most problems even when compared to insertion sort. It can be practical if the input is usually in sort order but may occasionally have some out-of-order elements nearly in position."

7. Write a PHP program to sort a list of elements using Cocktail sort.
Cocktail shaker sort (also known as bidirectional bubble sort, cocktail sort, shaker sort, ripple sort, shuffle sort, or shuttle sort) is a variation of bubble sort that is both a stable sorting algorithm and a comparison sort. The algorithm differs from a bubble sort in that it sorts in both directions on each pass through the list. This sorting algorithm is only marginally more difficult to implement than a bubble sort and solves the problem of turtles in bubble sorts. It provides only marginal performance improvements, and does not improve asymptotic performance; like the bubble sort, it is not of practical interest, though it finds some use in education.

8. Write a PHP program to sort a list of elements using Comb sort.

The Comb Sort is a variant of the Bubble Sort. Like the Shell sort, the Comb Sort increases the gap used in comparisons and exchanges. Some implementations use the insertion sort once the gap is less than a certain amount. The basic idea is to eliminate turtles, or small values near the end of the list since in a bubble sort these slow the sorting down tremendously. Rabbits, large values around the beginning of the list, do not pose a problem in bubble sort.

In bubble sort, when any two elements are compared, they always have a gap of 1. The basic idea of comb sort is that the gap can be much more than 1.

9. Write a PHP program to sort a list of elements using Gnome sort.

Gnome sort is a sorting algorithm originally proposed by Dr. Hamid Sarbazi-Azad (Professor of Computer Engineering at Sharif University of Technology) in 2000 and called "stupid sort" (not to be confused with bogosort), and then later on described by Dick Grune and named "gnome sort".

The algorithm always finds the first place where two adjacent elements are in the wrong order and swaps them. It takes advantage of the fact that performing a swap can introduce a new out-of-order adjacent pair only next to the two swapped elements.

10. Write a PHP program to sort a list of elements using Bucket sort.

11. Write a PHP program to sort a list of elements using Counting sort.
According to Wikipedia "In computer science, counting sort is an algorithm for sorting a collection of objects according to keys that are small integers; that is, it is an integer sorting algorithm. It operates by counting the number of objects that have each distinct key value, and using arithmetic on those counts to determine the positions of each key value in the output sequence. Its running time is linear in the number of items and the difference between the maximum and minimum key values, so it is only suitable for direct use in situations where the variation in keys is not significantly greater than the number of items. However, it is often used as a subroutine in another sorting algorithm, radix sort, that can handle larger keys more efficiently".
The algorithm loops over the items, computing a histogram of the number of times each key occurs within the input collection. It then performs a prefix sum computation (a second loop, over the range of possible keys) to determine, for each key, the starting position in the output array of the items having that key. Finally, it loops over the items again, moving each item into its sorted position in the output array.

12. Write a PHP program to sort a list of elements using Radix sort.

13. Write a PHP program to sort a list of elements using Bead sort.

14. Write a PHP program to sort a list of elements using Bogo sort.
In computer science, bogosort is a particularly ineffective sorting algorithm based on the generate and test paradigm. The algorithm successively generates permutations of its input until it finds one that is sorted. It is not useful for sorting but may be used for educational purposes, to contrast it with other more realistic algorithms.

15. Write a PHP program to sort a list of elements using Strand sort.
This is a way of sorting numbers by extracting shorter sequences of already sorted numbers from an unsorted list.

16. Write a PHP program to sort a list of elements using Patience sort.

Patience sorting is a sorting algorithm inspired by and named after, the card game patience. A variant of the algorithm efficiently computes the length of a longest increasing subsequence in a given array.

The algorithm's name derives from a simplified variant of the patience card game. This game begins with a shuffled deck of cards. These cards are dealt one by one into a sequence of piles on the table, according to the following rules.

Initially, there are no piles. The first card dealt forms a new pile consisting of the single card.

Each subsequent card is placed on the leftmost existing pile whose top card has a value greater than or equal the new card's value, or to the right of all of the existing piles, thus forming a new pile.

When there are no more cards remaining to deal, the game ends.

This card game is turned into a two-phase sorting algorithm, as follows. Given an array of n elements from some totally ordered domain, consider this array as a collection of cards and simulate the patience sorting game. When the game is over, recover the sorted sequence by repeatedly picking off the minimum visible card; in order words, perform an p-way merge of the p piles, each of which is internally sorted.

17. Write a PHP program to sort a list of elements using Merge sort.

According to Wikipedia "Merge sort (also commonly spelled mergesort) is an O (n log n) comparison-based sorting algorithm. Most implementations produce a stable sort, which means that the implementation preserves the input order of equal elements in the sorted output."

PHP File Handling

1. Write a PHP program to read and display the contents of a text file.

2. Write a PHP function that takes a file path as input and checks if the file exists.

3. Write a PHP script to count the number of lines in a text file.

4. Write a PHP function to write a string to a file.

5. Write a PHP program that reads a CSV file and displays the data in a tabular format.

6. Write a PHP script to rename a file.

7. Write a PHP function that copies a file from one location to another.

8. Write a PHP program that deletes a specific file from the server.

9. Write a PHP function that checks the file size and displays it in human-readable format (KB, MB, GB).

10. Write a PHP program that checks if a file is writable and displays appropriate messages.

11. Write a PHP program that appends a string to an existing file.

12. Write a PHP script to check the file extension and display appropriate messages for different file types.

13. Write a PHP function to check if a directory exists and create it if it doesn't.

14. Write a PHP program that searches for a specific word in a text file and displays the line number(s) where it occurs.

15. Write a PHP program that reads and displays the contents of a binary file.

16. Write a PHP program that reads an XML file and extracts specific data from it.

17. Write a PHP function that reads a JSON file and returns the decoded data as an associative array.

18. Write a PHP function to calculate the MD5 checksum of a file.

PHP Exception Handling

1. Write a PHP program that demonstrates the basic usage of try-catch blocks to handle exceptions.

2. Write a PHP program that creates a custom exception class in PHP and throw an instance of it within a try block.

3. Write a PHP program that implements a PHP function that divides two numbers but throws an exception if the denominator is zero.

4. Write a PHP script that uses try-catch blocks to handle different types of exceptions and display appropriate error messages.

5. Write a PHP program that reads data from a file and throws a custom exception if the file does not exist.

6. Write a PHP function that takes a string as input and throws an exception if the string is empty.

7. Write a PHP program that implements multiple catch blocks to handle different types of exceptions.

8. Write a PHP program that simulates a database connection and throws an exception if it fails.

9. Write a PHP script that implements the use of the finally block in exception handling.

10. Write a PHP program that uses a try-catch block to handle a specific exception and performs specific actions based on the exception type.

PHP Form

1. Write a PHP script to create an HTML form that collects a user's name, email, and age. Process the form data and display it on a new page.

2. Create an HTML form for user registration with fields for username, password, email, and confirm password. Write PHP code to validate that the passwords match and that the email is in a valid format.

3. Design an HTML form that allows users to upload files. Write PHP code to handle the file uploads, validate file types, and save the uploaded files to a directory.

4. Create a contact form with fields for name, email, and message. Implement PHP validation to check if all fields are filled out and use PHP to send an email with the submitted information.

5. Build an HTML form that includes checkboxes and radio buttons for user preferences. Use PHP to process the selected options and display the results.

6. Design a form with a dropdown menu that allows users to select their favorite color. Use PHP to process the selected option and display a corresponding message.

7. Integrate a CAPTCHA challenge into an HTML form to prevent automated submissions. Write PHP code to validate the CAPTCHA input and process the form if it's correct.

8. Create an HTML form with dynamic input fields (e.g., "Add More" button to add more input fields). Write PHP code to handle the dynamic input fields and process the form data.

9. Create a login form with fields for username and password. Implement PHP code to verify the user's login credentials and display a message based on the outcome.

10. Design a registration form with fields for username, password, and email. Write PHP code to validate the form data and store user information in a MySQL database.

PHP Cookies and Sessions

1. Write a PHP script to set a cookie named "username" with the value and an expiration time of one hour.

2. Write a PHP script to retrieve and display the value of the cookie named "username".

3. Write a PHP script to delete a cookie named "username".

4. Write a PHP script to set a session variable named "userid" with the value 10020.

5. Write a PHP script to retrieve and display the value of the session variable "userid".

6. Write a PHP script to destroy a session and unset all session variables.

7. Write a PHP script to set a secure cookie that can only be transmitted over an encrypted connection.

8. Write a PHP script to check if a cookie named "visited" exists. If it does, display a welcome message; otherwise,
display a default message.

9.Write a PHP script to store an array of user preferences in a session variable.

10. Write a PHP script to retrieve and display user preferences stored in the session variable.

11. Write a PHP script to set a session timeout after 30 minutes of inactivity.

12. Write a PHP script to display the number of active sessions on the server.

13. Write a PHP script to limit the maximum number of concurrent sessions for a user to 3.

14. Write a PHP script to regenerate the session ID to prevent session fixation attacks.

15. Write a PHP script to display the last time the session was accessed by the user.

16. Write a PHP script to set a cookie and a session variable with the same name. Display their values to compare.

PHP Class and Object

1. Write a PHP class 'Rectangle' that has properties for length and width. Implement methods to calculate the rectangle's area and perimeter.

2. Write a PHP class called 'Circle' that has a radius property. Implement methods to calculate the circle's area and circumference.

3. Write a PHP class called 'Shape' with an abstract method 'calculateArea()'. Create two subclasses, 'Triangle' and 'Rectangle', that implement the 'calculateArea()' method.

4. Write a PHP interface called 'Resizable' with a method 'resize()'. Implement the 'Resizable' interface in a class called 'Square' and add functionality to resize the square.

5. Write a PHP class called 'Vehicle' with properties like 'brand', 'model', and 'year'. Implement a method to display the vehicle details.

6. Write a PHP a class hierarchy for a library system, including classes like 'LibraryItem', 'Book', 'DVD', etc. Implement appropriate properties and methods for each class.

7. Write a PHP class called 'Student' with properties like 'name', 'age', and 'grade'. Implement a method to display student information.

8. Write a PHP a class called "BankAccount" with properties like "accountNumber" and "balance". Implement methods to deposit and withdraw money from the account.

9. Write a PHP abstract class called 'Animal' with abstract methods like 'eat()' and 'makeSound()'. Create subclasses like 'Dog', 'Cat', and 'Bird' that implement these methods.

10. Write a PHP class called 'Person' with properties like 'name' and 'age'. Implement the '__toString()' magic method to display person information.

11. Write a class called 'Employee' that extends the 'Person' class and adds properties like 'salary' and 'position'. Implement methods to display employee details.

12. Write a class called 'Product' with properties like 'name' and 'price'. Implement the 'Comparable' interface to compare products based on their prices.

13. Write a class called 'Logger' with a static property called 'logCount' that keeps track of the number of log messages. Implement a static method to log a message.

14. Write a class called 'Math' with static methods like 'add()', 'subtract()', and 'multiply()'. Use these methods to perform mathematical calculations.

15. Write a PHP class called 'File' with properties like 'name' and 'size'. Implement a static method to calculate the total size of multiple files.

16. Write a PHP class called 'Calculator' that has a private property called 'result'. Implement methods to perform basic arithmetic operations like addition and subtraction.

17. Write a PHP class called 'ShoppingCart' with properties like 'items' and 'total'. Implement methods to add items to the cart and calculate the total cost.

18. Write a PHP class called 'Logger' that uses the singleton design pattern to ensure only one instance of the class can be created.

19. Write a class called 'Validation' with static methods to validate email addresses, passwords, and other common input fields.

PHP and Database

1. User Registration and Login System:
  - Develop a PHP script that allows users to register by providing a username, email, and password. Store user information in a MySQL database.
  - Create a login page where users can enter their credentials to log in. Verify login information against the database.

2. Product Inventory Management:
  - Create a PHP script for adding, updating, and deleting products in an online store. Store product details in a MySQL database.
  - Implement a search feature that allows users to find products by name or category.

3. Content Management System (CMS):
  - Design a basic CMS using PHP and MySQL to manage articles. Users should be able to add, edit, and delete articles.
  - Implement a search functionality for finding articles based on keywords.

4. User Reviews and Ratings:
  - Build a system that allows users to submit product reviews and ratings. Store review details and ratings in a MySQL database.
  - Create a page that displays an average rating for each product based on user reviews.

5. E-commerce Shopping Cart:
  - Develop a shopping cart system that allows users to add and remove items from their cart. Store cart data in a MySQL database.
  - Implement a checkout process for users to complete their orders.

6. User Profile Management:
  - Design a PHP script for users to update their profiles. Users can change their username, email, and password.
  - Ensure that the updated user data is stored correctly in the database.

7. Blog System:

- Create a blog system where users can write and publish blog posts. Store blog posts and metadata in a MySQL database.
- Include a comment system that allows users to leave comments on blog posts.

8. Event Registration System:

- Develop a system for event registration. Users should be able to sign up for events, and event details are stored in a MySQL database.
- Create an admin panel to manage event registrations.

9. Membership Subscription System:

- Build a membership subscription system that stores user subscription information in a MySQL database.
- Implement payment processing and user access control based on subscription status.

10. Survey and Feedback System:

- Create a system for conducting surveys and collecting user feedback. Store survey questions and responses in a MySQL database.
- Develop a reporting feature that displays survey results.