

phphtml.net

Задачник PHP

Трепачев Дмитрий Петрович

Минск 2014
проект phphtml.net
серия “Учебники PHP для начинающих”

Задачник по PHP

Трепачев Дмитрий Петрович

Преподаватель курсов, программист, фрилансер

Трепачев Д. П., фрилансер-программист

Книга представляет собой задачник по языку PHP.

<http://phphtml.net>

<http://sitetech.by>

http://vk.com/html_css_php_help

Минск, 2014

Версия задачника: 1.0

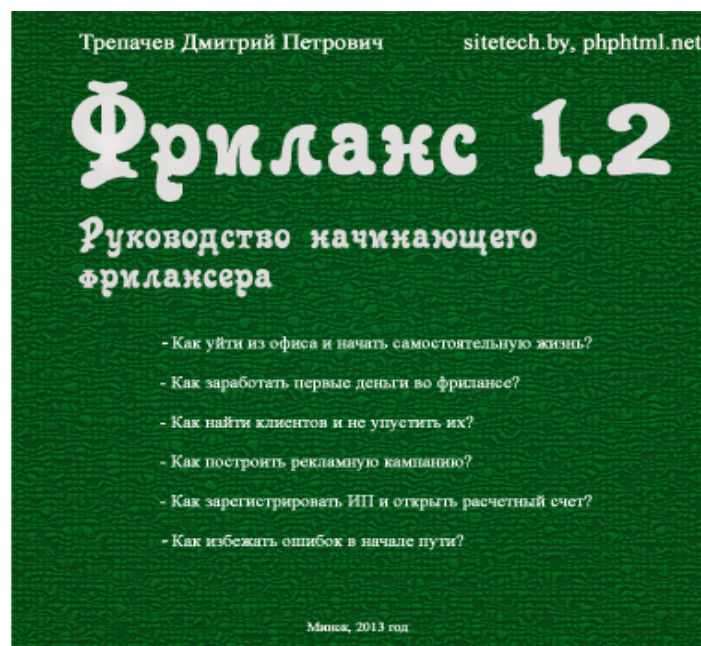
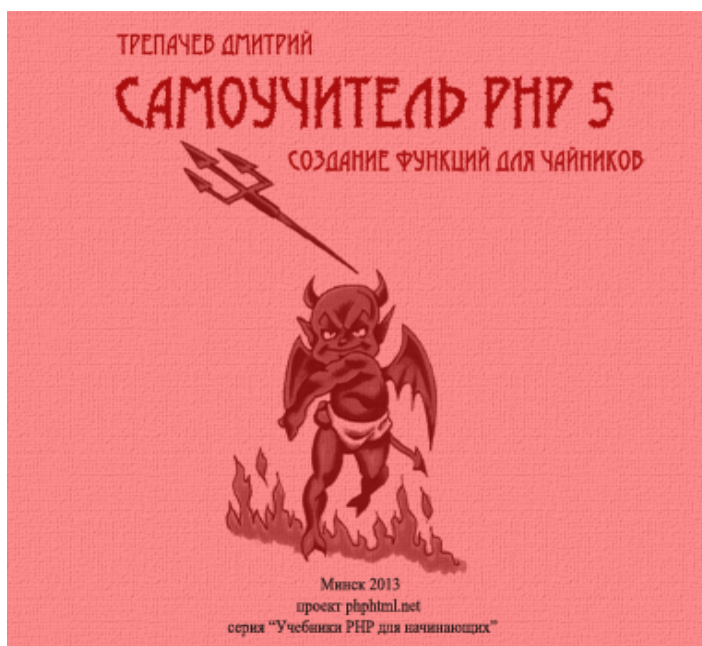
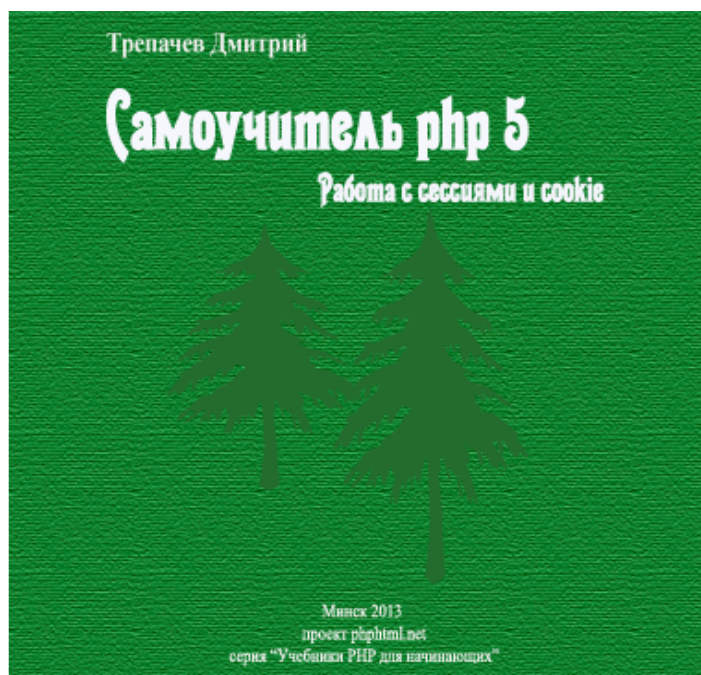
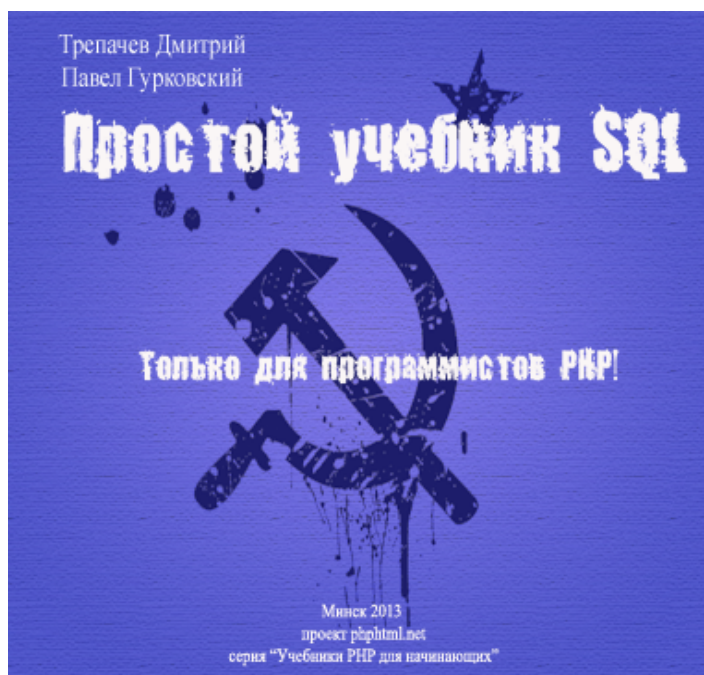
Оставьте свой отзыв!

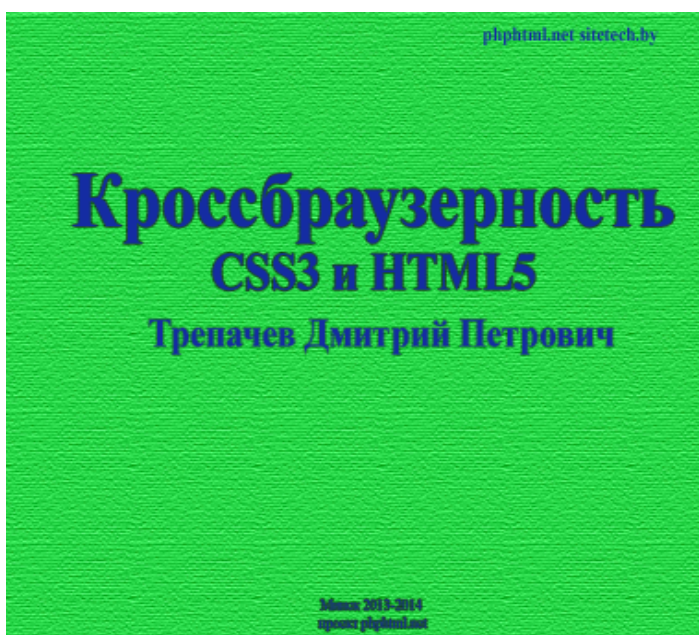
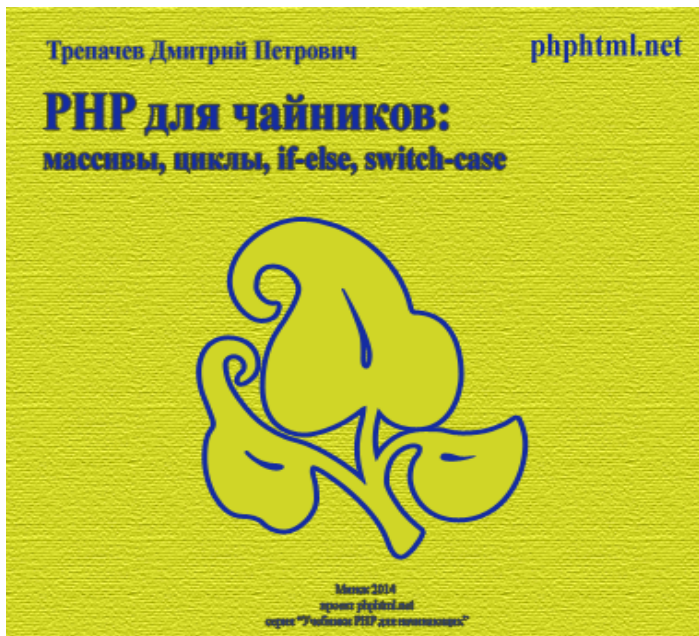
Очень большая просьба тем, кто прочитал книгу — оставьте свой отзыв здесь <http://phphtml.net>. Он для меня очень ценен и помогает понять, что я не зря трачу время, создавая бесплатные руководства и справочники.

Скачать больше книг

Все мои книги вы можете скачать на моем сайте

<http://phphtml.net>





Записаться на курсы

Вы можете записаться на групповые и индивидуальные курсы HTML, PHP, JavaScript, jQuery, CSS3+HTML5, web-хакинг и другое.

Смотрите на <http://phphtml.net>.

Еженедельные бесплатные вебинары

Каждую неделю на сайте phphtml.net при поддержке паблика vk.com/html_css_php_help проходят бесплатные вебинары на тематику «Создание сайтов». Подключайтесь!

Оглавление

Основы языка PHP.....	6
Работа с переменными.....	6
Работа со строками.....	6
Обращение к символам строки.....	6
Практика.....	7
Присваивание. Работа с массивами. Типизация переменных.....	8
Работа с присваиванием и декрементами.....	8
Работа с массивами.....	9
Ассоциативные массивы.....	9
Многомерные массивы.....	10
Работа с типами переменных.....	10
Конструкция if-else.....	11
Работа с if-else.....	11
Работа с empty и isset.....	11
Работа с логическими переменными.....	12
Практика.....	12
Работа с OR и AND.....	12
Задачи.....	13
Циклы foreach, while, for.....	14
Работа с foreach.....	14
Работа с ключами.....	14
Практика.....	14
Циклы while и for.....	14
Задачи.....	15
Математические функции PHP.....	17
Работа с %.....	17
Работа с модулем.....	17
Работа со степенью и корнем.....	17
Работа с функциями округления.....	17

Работа с min и max.....	17
Работа с рандомом.....	18
Задачи.....	18
Функции работы со строками.....	20
Работа с регистром символов.....	20
Работа с strlen, substr.....	20
Работа с str_replace.....	20
Работа с explode, implode.....	21
Работа с trim, ltrim, rtrim.....	21
Работа с strip_tags и htmlspecialchars.....	21
Работа с chr и ord.....	21
Работа с substr_count, str_word_count, str_split.....	22
Работа с str_repeat, strrev.....	22
Работа с формами в PHP.....	24
Формы.....	24
GET-запросы и функция include.....	24
Атрибуты value и placeholder.....	25
Массив \$_SERVER.....	25
Работа с датами в PHP.....	26
Timestamp: time и mktime.....	26
Функция date.....	26
Сравнение дат.....	27
На strtotime.....	27
Прибавление и отнимание от дат.....	27
Задачи.....	27
Функции работы с массивами.....	29
На count и range.....	29
На array_sum и array_product.....	29
На array_merge и array_slice.....	29
На array_keys, array_values, array_combine.....	29

На array_flip, array_count_values, array_reverse, array_unique.....	29
На сортировку, shuffle и array_rand.....	30
На array_map и array_walk.....	30
На array_chunk и array_pad.....	30
Работа с функциями в PHP.....	31
Простые функции.....	31
На параметры по умолчанию.....	31
На static и global.....	31
На html.....	31
На строки.....	31
На даты.....	31
На перевод чисел в месяцы и дни.....	32
На строки.....	33
На ссылки.....	33
На списки и таблицы.....	34

ОСНОВЫ ЯЗЫКА PHP

Работа с переменными

1. Создайте переменную **\$a** и присвойте ей значение 3. Выведите значение этой переменной на экран.
2. Создайте переменные **\$a=10** и **\$b=2**. Выведите на экран их сумму, разность, произведение и частное (результат деления).
3. Создайте переменные **\$c=15** и **\$d=2**. Просуммируйте их, а результат присвойте переменной **\$result**. Выведите на экран значение переменной **\$result**.
4. Создайте переменные **\$a=10**, **\$b=2** и **\$c=5**. Выведите на экран их сумму.
5. Создайте переменные **\$a=17** и **\$b=10**. Отнимите от **\$a** переменную **\$b** и результат присвойте переменной **\$c**. Затем создайте переменную **\$d**, присвойте ей значение 7. Сложите переменные **\$c** и **\$d**, а результат запишите в переменную **\$result**. Выведите на экран значение переменной **\$result**.

Работа со строками

6. Создайте переменную **\$text** и присвойте ей значение 'Привет, Мир!'. Выведите значение этой переменной на экран.
7. Создайте переменные **\$text1='Привет, '** и **\$text2='Мир!'**. С помощью этих переменных и операции сложения строк выведите на экран фразу 'Привет, Мир!'.
\$text1 = 'Привет, ' и **\$text2** = 'Мир!'
8. Создайте переменную **\$name** и присвойте ей ваше имя. Выведите на экран фразу 'Привет, %Имя%!'.
\$name = ваше имя
9. Создайте переменную **\$age** и присвойте ей ваш возраст. Выведите на экран 'Мне %Возраст% лет!'.
\$age = ваш возраст

Обращение к символам строки

10. Создайте переменную **\$text** и присвойте ей значение 'Мама мыла раму!'. Выведите символы: ы, м, а, у, ! всеми возможными способами.
11. Создайте переменную **\$game** и присвойте ей значение 'Путешественник'. Обращаясь с этим словом как с отдельными символами составьте различные русские слова.

Практика

12. Напишите скрипт, который считает количество секунд в часе, в сутках, в месяце.
13. Создайте три переменные — час, минута, секунда. С их помощью выведите время начала нашего занятия в формате 'час:минута:секунда'.
14. Создайте переменную, присвойте ей число. Возведите это число в квадрат. Выведите его на экран.
15. Напишите скрипт, который переводит температуру из градусов Цельсия в градусы Фарингейта. Формула: $T_c = \frac{5 \cdot (T_f - 32)}{9}$. Подсказка: для градусов Фарингейта следует ввести переменную и записать в нее температуру, которую нужно перевести.

Присваивание. Работа с массивами.

Типизация переменных.

Работа с присваиванием и декрементами

1. Переделайте этот код так, чтобы в нем использовалась только одна переменная **\$var**. Количество строк кода при этом не должно измениться!

```
$var = 3;  
$var1 = $var + 5;  
$var2 = $var1 * 10;  
echo $var2;
```

2. Переделайте этот код так, чтобы в нем использовались операции **+=**, **.=**, ***=**, **-=**. Количество строк кода при этом не должно измениться!

```
$var = 47;  
$var = $var + 7;  
$var = $var - 18;  
$var = $var * 10;  
echo $var;
```

3. Переделайте этот код так, чтобы в нем использовались операции **+=**, **.=**, ***=**, **-=**. Количество строк кода при этом не должно измениться!

```
$text = 'Я';  
$text = $text . ' хочу' ;  
$text = $text . ' знать' ;  
$text = $text . ' PHP!' ;  
echo $text;
```

4. Переделайте этот код так, чтобы в нем использовались операции **++** и **--**. Количество строк кода при этом не должно измениться!

```
$var = 10;  
$var = $var + 1;  
$var = $var + 1;  
$var = $var - 1;  
echo $var;
```

5. Переделайте этот код так, чтобы в нем использовались операции **+++**, **-**, **+=**, **.=**, ***=**, **-=**. Количество строк кода при этом не должно измениться!

```
$var = 10;  
$var = $var + 7;  
$var = $var + 1;
```



```
$var = $var - 1;  
$var = $var + 12;  
$var = $var * 7;  
$var = $var - 15;  
echo $var;
```

Работа с массивами

6. Создайте массив **\$arr=array('joomla', 'wordpress', 'netcat')**. Выведите значение массива на экран с помощью команды **var_dump()**.
7. С помощью массива **\$arr** из предыдущего номера выведите на экран содержимое первого, второго и третьего элементов.
8. Создайте массив **\$arr=array('html', 'css', 'php')** и с его помощью выведите на экран строку 'php, html, css'.
9. Создайте массив **\$arr** с элементами 2, 5, 3, 9. Умножьте первый элемент массива на второй, а третий элемент на четвертый. Результаты сложите, присвойте переменной **\$result**. Выведите на экран значение этой переменной.
10. Создайте массив **\$arr** с элементами 'a', 'b', 'c', 78 двумя различными способами.
11. Создайте массив **\$arr** с элементами 'Я', 'учу', 'PHP', '!'. С его помощью выведите на экран фразу 'Я учу PHP!'.
12. Создайте массив **\$arr=array('a', 'b', 'c', 'd', 'e')**. С помощью одной переменной **\$var** поменяйте местами элементы 'b' и 'c'.
13. Создайте массив **\$arr=array('a', 'b', 'c', 'd', 'e')**. С помощью одной переменной **\$var** сделайте из него массив **array('e', 'd', 'c', 'b', 'a')**.

Ассоциативные массивы

14. Создайте массив **\$arr**. Выведите значение массива на экран с помощью команды **var_dump()**.

```
$arr = array('text'=>'текст!', 1=>'125', 'key'=>'element');  
var_dump($arr);
```

15. Создайте массив заработных плат **\$arr**. Выведите на экран зарплату Пети и Коли.

```
$arr = array('Коля'=>'1000$', 'Вася'=>'500$', 'Петя'=>'200$');
```

16. Создайте ассоциативный массив дней недели. Ключами в нем должны служить номера дней от начала недели (понедельник — первый и т.д.). Выведите на экран текущий день недели.

17. Создайте массив **\$arr** с ключами 'I', 'study', 'php', 'sign' и соответствующими элементами 'Я', 'учу', 'PHP', '!'. С его помощью выведите на экран фразу 'Я учу PHP!'.

Многомерные массивы

18. Создайте многомерный массив **\$arr**. С его помощью выведите на экран слова 'joomla', 'drupal', 'зеленый', 'красный'.

```
$arr = array(  
    'cms'=>array('joomla','wordpress','drupal'),  
    'colors'=>array('blue'=>'голубой','red'=>'красный','green'=>'зеленый'));
```

19. Создайте двухмерный массив. Первые два ключа — это 'ru' и 'en'. Первый ключ содержит элемент, являющийся массивом названий дней недели по-русски, а второй — по-английски.

Работа с типами переменных

20. Определите тип следующих переменных с помощью функции `gettype()`.

```
$var = array('joomla','wordpress','drupal');  
$var = 'привет!';  
$var = 23;  
$var = '23';  
$var = true;  
$var = 'true';  
$var = 2.5;  
$var = null;
```

21. Проверьте тип переменных из предыдущего с помощью функций `is_null`, `is_numeric`, `is_integer`, `is_string`, `is_array`, `is_bool`.

Конструкция if-else

Работа с if-else

1. Если переменная **\$a равна нулю**, то выведите 'Верно!', иначе выведите 'Неверно!'.
Проверьте работу скрипта при **\$a**, равном 1, 0, -3.
2. Если переменная **\$a больше нуля**, то выведите 'Верно!', иначе выведите 'Неверно!'.
Проверьте работу скрипта при **\$a**, равном 1, 0, -3.
3. Если переменная **\$a меньше нуля**, то выведите 'Верно!', иначе выведите 'Неверно!'.
Проверьте работу скрипта при **\$a**, равном 1, 0, -3.
4. Если переменная **\$a больше или равна нулю**, то выведите 'Верно!', иначе выведите 'Неверно!'. Проверьте работу скрипта при **\$a**, равном 1, 0, -3.
5. Если переменная **\$a меньше или равна нулю**, то выведите 'Верно!', иначе выведите 'Неверно!'. Проверьте работу скрипта при **\$a**, равном 1, 0, -3.
6. Если переменная **\$a не равна нулю**, то выведите 'Верно!', иначе выведите 'Неверно!'. Проверьте работу скрипта при **\$a**, равном 1, 0, -3.
7. Если переменная **\$a равна 'test'**, то выведите 'Верно!', иначе выведите 'Неверно!'.
Проверьте работу скрипта при **\$a**, равном 'test', 'тест', 3.
8. Если переменная **\$a равна '1' и по значению и по типу**, то выведите 'Верно!', иначе выведите 'Неверно!'. Проверьте работу скрипта при **\$a**, равном '1', 1, 3.

Работа с empty и isset

9. Если переменная **\$a пустая**, то выведите 'Верно!', иначе выведите 'Неверно!'.
Проверьте работу скрипта при **\$a**, равном 1, 0, -3, null, true, "".
10. Если переменная **\$a НЕ пустая**, то выведите 'Верно!', иначе выведите 'Неверно!'.
Проверьте работу скрипта при **\$a**, равном 1, 0, -3, null, true, "".
11. Если переменная **\$a существует**, то выведите 'Верно!', иначе выведите 'Неверно!'.
Проверьте работу скрипта при **\$a**, равном 1, 0, -3, null, true, "".
12. Если переменная **\$a НЕ существует**, то выведите 'Верно!', иначе выведите 'Неверно!'. Проверьте работу скрипта при **\$a**, равном 1, 0, -3, null, true, "".

Работа с логическими переменными

13. Если переменная **\$var** **равна true**, то выведите 'Верно!', иначе выведите 'Неверно!'. Проверьте работу скрипта при **\$var**, равном true, false. Напишите два варианта скрипта — с короткой записью и с длинной.
14. Если переменная **\$test** **не равна true**, то выведите 'Верно!', иначе выведите 'Неверно!'. Проверьте работу скрипта при **\$test**, равном true, false. Напишите два варианта скрипта — с короткой записью и с длинной.

Практика

15. Создайте переменные **\$a** и **\$b**. Просуммируйте их, а результат запишите в переменную **\$result**. Если **\$result** больше 5, то присвойте переменной **\$result** значение 5. Если же она меньше 5-ти — то умножьте ее на 10. Выведите на экран значение переменной **\$result**. Проверьте работу скрипта при **\$a** и **\$b**, равных 2 и 5, 3 и 1.

Работа с OR и AND

16. Если переменная **\$a** **больше нуля и меньше 5-ти**, то выведите 'Верно!', иначе выведите 'Неверно!'. Проверьте работу скрипта при **\$a**, равном 5, 0, -3, 2.
17. Если переменная **\$a** **равна нулю или равна двум**, то поделите ее на 10, иначе прибавьте к ней 7 и выведите ее на экран. Проверьте работу скрипта при **\$a**, равном 5, 0, -3, 2.
18. Если переменная **\$a** **не равна 1 или не равна 3**, то выведите 'Верно!', иначе выведите 'Неверно!'. Проверьте работу скрипта при **\$a**, равном 1, 0, 3, 2.
19. Если переменная **\$a** **равна или меньше 1**, а переменная **\$b** **больше или равна 3**, то выведите сумму этих переменных, иначе выведите 'Неверно!'. Проверьте работу скрипта при **\$a** и **\$b**, равном 1 и 3, 0 и 6, 3 и 5.
20. Если переменная **\$a** **больше нуля и меньше 5-ти**, то увеличьте **\$a** на 1, иначе прибавьте к **\$a** число 5. Выведите новое значение переменной на экран. Проверьте работу скрипта при **\$a**, равном 5, 0, -3, 9.
21. Если переменная **\$a** **больше 2-х и меньше 11-ти**, **или** переменная **\$b** **больше или равна 6-х и меньше 14-ти**, то увеличьте **\$a** на 2, иначе прибавьте к **\$a** число 5.

Выведите новое значение переменной на экран. Проверьте работу скрипта самостоятельно.

Задачи

22. Переменная **\$lang** может принимать два значения: 'ru' и 'en'. Если она имеет значение 'ru', то в переменную **\$arr** запишем массив дней недели на русском языке, а если имеет значение 'en' – то на английском.
23. В переменной **\$day** лежит число от 1 до 31. Определите в какую декаду месяца попадает это число (в первую, вторую или третью).
24. В переменной **\$month** лежит число от 1 до 12. Определите в какую пору года попадает этот месяц (зима, лето, весна, осень).
25. Переменная **\$lang** может принимать два значения: 'ru' и 'en'. Переменная **\$day** принимает значение от одного до 7-ми. Если **\$lang** имеет значение 'ru', то в переменную **\$result** запишем название дня недели на русском языке в соответствии со значением переменной **\$day** (1 — понедельник, 2 — вторник и т.д.). Если же **\$lang** имеет значение 'en' – то поступим аналогично, но день недели будет на английском.
26. Переменная **\$lang** может принимать два значения: 'ru' и 'en'. Переменная **\$month** принимает значение от одного до 12-ти. Если **\$lang** имеет значение 'ru', то в переменную **\$result** запишем название месяца на русском языке в соответствии со значением переменной **\$month** (1 — январь, 2 — февраль и т.д.). Если же **\$lang** имеет значение 'en' – то поступим аналогично, но месяц будет на английском.
27. Если вы вкладываете деньги под проценты в банк, то он начислит вам 10% годовых, если начальная сумма более 1000\$ и 7% годовых, если начальная сумма меньше 1000\$. Найдите сумму, которая будет на счету у вкладчика через 1 год, 2 года... 10 лет. Решите задачу с капитализацией и без.
28. Вы взяли кредит на сумму **\$summ**. Каждый месяц вы выплачиваете банку 10% от начальной суммы кредита и каждый месяц банк начисляет 7% на остаток по кредиту. Найдите за сколько месяцев вы выплатите кредит в 400\$ и сколько составит переплата.

Циклы foreach, while, for

Работа с foreach

1. Дан массив с элементами 'html', 'css', 'php', 'js', 'jq'. С помощью цикла foreach выведите эти слова в столбик.
2. Дан массив с элементами 1, 20, 15, 17, 24, 35. С помощью цикла foreach найдите сумму элементов этого массива. Запишите ее в переменную **\$result**.
3. Дан массив с элементами 26, 17, 136, 12, 79, 15. С помощью цикла foreach найдите сумму квадратов элементов этого массива. Результат запишите в переменную **\$result**.

Работа с ключами

4. Дан массив **\$arr**. С помощью **первого** цикла foreach выведите на экран столбец ключей, с помощью **второго** — столбец элементов.

```
$arr = array('green'=>'зеленый', 'red'=>'красный', 'blue'=>'голубой');
```

5. Дан массив **\$arr** с ключами 'Коля', 'Вася', 'Петя' с элементами '200', '300', '400'. С помощью цикла foreach выведите на экран столбец строк такого формата: 'Коля — зарплата 200 долларов.'
6. Дан массив **\$arr**. С помощью цикла foreach запишите английские названия в массив **\$en**, а русские — в массив **\$ru**.

```
$arr = array('green'=>'зеленый', 'red'=>'красный', 'blue'=>'голубой');  
$en = array('green', 'red', 'blue');  
$ru = array('зеленый', 'красный', 'голубой');
```

Практика

7. Дан массив с элементами 2, 5, 9, 15, 0, 4. С помощью цикла foreach и оператора if выведите на экран столбец тех элементов массива, которые больше 3-х, но меньше 10.
8. Дан массив с элементами 1, 2, 3, 4, 5, 6, 7, 8, 9. С помощью цикла foreach создайте строку '-1-2-3-4-5-6-7-8-9-'.

Циклы while и for

9. Выведите столбец чисел от 1 до 100.
10. Выведите столбец чисел от 11 до 33.

11. Выведите столбец четных чисел в промежутке от нуля до 100.
12. Есть число **\$n** = 1000. Делите его на 2 столько раз, пока результат деления не станет меньше 50. Какое число получится? Посчитайте количество итераций, необходимых для этого (итерации — это количество проходов цикла), и запишите его в переменную **\$num**.

Задачи

13. Дан массив с элементами 4, 2, 5, 19, 13, 0, 10. С помощью цикла `foreach` и оператора `if` выведите на экран столбец тех элементов массива, которые больше 3-х, но меньше 10.
14. Дан массив с элементами 4, 2, 5, 19, 13, 0, 10. С помощью цикла `foreach` и оператора `if` проверьте есть ли в массиве элемент со значением **\$e**, равном 2, 3 или 4. Если есть — выведите на экран 'Есть!', иначе выведите 'Нет!'.
15. Дан массив **\$arr**. С помощью цикла `foreach` и переменной **\$count** подсчитайте количество элементов этого массива. Проверьте работу скрипта на примере массива с элементами 4, 2, 5, 19, 13, 0, 10.
16. Дан массив с элементами 1, 2, 3, 4, 5, 6, 7, 8, 9. С помощью цикла `foreach` и оператора `if` выведите на экран столбец элементов массива, как показано на картинке.

```
1, 2, 3 <br>
4, 5, 6 <br>
7, 8, 9 <br>
```

17. Составьте массив месяцев. С помощью цикла `foreach` выведите все месяцы, а текущий месяц выведите жирным (****). Текущий месяц должен храниться в переменной **\$month**.
18. Составьте массив дней недели. С помощью цикла `foreach` выведите все дни недели, выходные дни следует вывести жирным (****).
19. Составьте массив дней недели. С помощью цикла `foreach` выведите все дни недели, а текущий день выведите курсивом (*<i></i>*). Текущий день должен храниться в переменной **\$day**.
20. Нарисуйте пирамиду, как показано на рисунке, только у вашей пирамиды должно

быть 20 рядов, а не 5.

```
x
xx
xxx
xxxx
xxxxx
```

21. Нарисуйте пирамиду, как показано на рисунке, только у вашей пирамиды должно быть 9 рядов, а не 5.

```
1
22
333
4444
55555
```

22. Нарисуйте пирамиду, как показано на рисунке, воспользовавшись циклом `for` или `while`.

```
xx
xxxx
xxxxxx
xxxxxxxx
xxxxxxxxxx
```

Математические функции PHP

Работа с %

1. Даны переменные $\$a = 10$ и $\$b = 3$. Найдите остаток от деления $\$a$ на $\$b$.
2. Даны переменные $\$a$ и $\$b$. Проверьте, что $\$a$ делится без остатка на $\$b$. Если это так — выведите 'Делится' и результат деления, иначе выведите 'Делится с остатком' и остаток от деления.

Работа с модулем

3. Даны переменные $\$a$ и $\$b$. Найдите модуль разности $\$a$ и $\$b$. Проверьте работу скрипта самостоятельно для различных $\$a$ и $\$b$.
4. Даны переменные $\$a$ и $\$b$. Отнимите от $\$a$ переменную $\$b$ и результат присвойте переменной $\$c$. Сделайте так, чтобы в любом случае в переменную $\$c$ записалось положительное значение. Проверьте работу скрипта при $\$a$ и $\$b$, равных соответственно 3 и 5, 6 и 1.

Работа со степенью и корнем

5. Возведите 2 в 10 степень. Результат запишите в переменную $\$st$.
6. Найдите квадратный корень из 245.
7. Дан массив с элементами 4, 2, 5, 19, 13, 0, 10. Найдите квадратный корень из суммы квадратов его элементов. Для решения воспользуйтесь циклом `foreach`.

Работа с функциями округления

8. Найдите квадратный корень из 379. Результат округлите до целых, до десятых, до сотых.
9. Найдите квадратный корень из 587. Округлите результат в большую и меньшую сторону, запишите результаты округления в ассоциативный массив с ключами 'floor' и 'ceil'.

Работа с min и max

10. Даны числа 4, -2, 5, 19, -130, 0, 10. Найдите минимальное и максимальное числа.

Работа с рандомом

11. Выведите на экран случайное число от 1 до 100.
12. Сделайте так, чтобы каждый раз при обновлении страницы на экран выводилась случайным образом одна из трех картинок.
13. Заполните массив 10-ю случайными числами. (Подсказка: нужно воспользоваться циклами for или while).

Задачи

14. Напишите скрипт, который будет находить корни квадратного уравнения.
15. Напишите скрипт, который будет находить факториал числа. Факториал — это произведение всех целых чисел, меньше данного. Например, $4! = 1*2*3$.
16. Напишите скрипт, который проверяет, является ли данное число простым (простое число — это то, которое делится только на 1 и на само себя).
17. Дан массив **\$arr**. Найдите среднее арифметическое его элементов. Подсказка: количество элементов массива можно подсчитать функцией count. Проверьте задачу на массиве с элементами 12, 15, 20, 25, 59, 79.
18. Напишите скрипт-калькулятор единого налога. Налог считается так: если вы заработали менее, чем ставка **\$stavka** этого налога, умноженная на 30 — то платите ставку. Если же вы заработали более, чем $30* \$stavka$, то платите не только ставку налога, но и 5% от той суммы, на которую вы превысили $30* \$stavka$.
19. Напишите скрипт-калькулятор оплаты за электроэнергию. Сумма считается так: если вы потратили за месяц менее 150 киловатт, то сумма считается по одной ставке **\$summ1** за 1 киловатт, всё, что больше 150 и меньше 300 киловатт считается по ставке **\$summ2**, а все киловатты после 300 — по ставке **\$summ3**.
20. Напишите скрипт, который найдет все простые числа в заданном промежутке. Проверьте его работу на промежутке от 1 до 1000.
21. Напишите скрипт, который реализует алгоритм 'решето Эратосфена' для поиска простых чисел в заданном промежутке. Алгоритм найдите самостоятельно в поисковых системах.
22. Напишите скрипт, который будет разлагать заданное число на простые множители.

23. Напишите скрипт, который будет находить коэффициенты a , b и c квадратного уравнения для данных корней. То есть скрипт является генератором квадратных уравнений. Подсказка: погуглите про дискриминант и теорему Виета.
24. Напишите скрипт, который будет находить тройки целых чисел, удовлетворяющих теореме Пифагора. Решите задачу двумя способами: 'в лоб' и воспользовавшись специальными формулами: <http://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D1%84%D0%B0%D0%B3%D0%BE%D1%80%D0%BE%D0%B2%D0%B0%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0>

Функции работы со строками

Работа с регистром символов

1. Дана строка 'Привет, мир!'. Сделайте из нее строку 'ПРИВЕТ МИР!'.
2. Дана строка 'PHP'. Сделайте из нее строку 'php'.
3. Дана строка 'москва'. Сделайте из нее строку 'Москва'.
4. Дана строка 'МОСКВА'. Сделайте из нее строку 'Москва'.
5. Дана строка 'иванов иван иванович'. Сделайте из нее строку 'Иванов Иван Иванович'.

Работа с strlen, substr

6. Дана строка 'я учу PHP!'. Найдите количество символов в этой строке.
7. Дана строка 'я учу PHP!'. Вырежьте из нее слово 'учу' и слово 'PHP'.
8. Дана переменная \$str, в которой хранится какой-либо текст. Реализуйте обрезание длинного текста по следующему принципу: если количество символов этого текста больше заданного в переменной \$n, то в переменную \$result запишем первые \$n символов строки \$str и добавим в конец троеточие '...'. В противном случае в переменную \$result запишем содержимое переменной \$str.
9. Дана переменная \$password, в которой хранится пароль пользователя. Если количество символов пароля больше 5-ти и меньше 10-ти, то выведите пользователю сообщение о том, что пароль подходит, иначе сообщение о том, что нужно придумать другой пароль.
10. Нарисуйте пирамиду, как показано на рисунке, только у вашей пирамиды должно столько рядов, чтобы последний элемент пирамидки состоял из одного символа. Первый ряд пирамиды должен храниться в переменной \$str (может иметь различное количество символов). Подсказка: воспользуйтесь функциями strlen и substr.

```
xxxxxxxxx
xxxxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
```

Работа с str_replace

11. Дана строка 'Я-учу-PHP!'. Замените все дефисы на тег '
'.

12. Дана строка '31.12.2013'. Замените все точки на дефисы.
13. Дана строка \$str. Замените смайлики ':)', ':(', '^_^', которые встречаются в этой строке на соответствующие картинки ().
14. Дана переменная \$str, в которой хранится строка русского текста. Напишите скрипт, который запишет транслит этого текста в переменную \$translit. Напишите также скрипт, который выполнит обратную операцию.

Работа с explode, implode

15. Дана строка 'я учу PHP!'. С помощью функции explode запишите каждое слово этой строки в отдельный элемент массива.
16. Дан массив с элементами 'html', 'css', 'php', 'js'. С помощью функции implode создайте строку из этих элементов, разделенных запятыми.
17. В переменной \$date лежит дата в формате '2013-12-31'. Преобразуйте эту дату в формат '31.12.2013'.
18. В переменной \$date лежит дата в формате '31.12.2013'. Преобразуйте эту дату в формат '2013-12-31'.

Работа с trim, ltrim, rtrim

19. Дана строка ' php '. Сделайте из нее 3 разных строки с помощью функций класса trim: 'php', ' php', 'php '.

Работа с strip_tags и htmlspecialchars

20. Дана строка 'html, php, js'. Удалите теги из этой строки.
21. Дана строка 'html, php, js'. Выведите ее на экран 'как есть': то есть браузер не должен преобразовать в жирный.

Работа с chr и ord

22. Узнайте код символов 'a', 'b', 'c', пробела.
23. Изучите таблицу ASCII <http://www.asciitable.com/>. Определите границы, в которых располагаются буквы английского алфавита.
24. Выведите на экран символ с кодом 33.
25. Запишите в переменную \$str случайный символ — большую букву латинского

алфавита. Подсказка: с помощью таблицы ASCII определите какие целые числа соответствуют большим буквам латинского алфавита.

26. Запишите в переменную **\$str** случайную строку **\$len** длиной, состоящую из маленьких букв латинского алфавита. Подсказка: воспользуйтесь циклом **for** или **while**.

Работа с **substr_count**, **str_word_count**, **str_split**

27. Дана строка 'Мама мыла раму'. Узнайте количество букв 'а' и 'м', входящих в эту строку.
28. Скопируйте весь текст со страницы <http://www.php.ru/functions/?cat=strings>, запишите его в переменную **\$str**. Подсчитайте количество символов и количество слов в данной строке.
29. Создайте массив гласных букв. С помощью этого массива подсчитайте количество гласных в строке **\$str**. Результат представьте в виде ассоциативного массива, где ключами будут буквы, а элементами их количество.
30. Дана строка '1234567890'. Разбейте ее на массив с элементами '12', '34', '56', '78', '90'.

Работа с **str_repeat**, **strrev**

31. Проверьте, является ли пара слов палиндромом (одинаково читается во всех направлениях, кот-ток, нос-сон).
32. Дан массив **\$arr**. Найдите в нем все пары слов-палиндромов (одинаково читаются во всех направлениях, кот-ток, нос-сон). Результат выводите в виде строка формата 'нос — сон'. Проверьте работу скрипта на массиве **\$arr** из примера. Совет: нужно сделать не один, а два цикла.

```
$arr = array('слон', 'нос', 'ток', 'так', 'кот', 'пир', 'мир', 'сон', 'рим');
```

33. Определите является ли фраза палиндромом. Примеры: 'А роза упала на лапу Азора', 'И куци, и щели. И леци, и жуки', 'Лак, суп, Сара, Сура. Пылала сечь. Шел, плешь чесал, алы паруса распускал'. Обратите внимание на то, что при обратном чтении игнорируются пробелы, запятые, дефисы, тире и большие буквы (подсказка: значит сначала нужно привести строку к стандартному виду — удалить лишние символы, привести все к нижнему регистру).

34. Нарисуйте пирамиду, как показано на рисунке, только у вашей пирамиды должно быть 9 рядов, а не 5. Решите задачу с помощью одного цикла и функции `str_repeat`.

```
x
xx
xxx
xxxx
xxxxx
```

35. Нарисуйте пирамиду, как показано на рисунке, только у вашей пирамиды должно быть 9 рядов, а не 5. Решите задачу с помощью одного цикла и функции `str_repeat`.

```
1
22
333
4444
55555
```


Работа с формами в PHP

Формы

1. Спросите имя пользователя с помощью формы. Результат запишите в переменную **\$name**. Выведите на экран фразу 'Привет, %Имя%'.

```
Привет, Дмитрий, 25 лет.  
Твое сообщение: ...
```
2. Спросите у пользователя имя, возраст, а также попросите его ввести сообщение (textarea). Выведите эти данные на экран в формате, приведенном под данной задачей. Позаботьтесь о том, чтобы пользователь не мог вводить теги и таким образом сломать сайт.

3. Спросите возраст пользователя. Если форма была отправлена и введен возраст, то выведите его на экран, а **форму уберите**. Если же форма не была отправлена (это будет при первом заходе на страницу) — просто покажите ее.
4. Спросите у пользователя логин и пароль (в браузере должен быть звездочками). Сравните их с логином **\$login** и паролем **\$pass**, хранящихся в файле. Если все верно — выведите '*Доступ разрешен!*', в противном случае — '*Доступ запрещен!*'. Сделайте так, чтобы скрипт обрезал концевые пробелы в строках, которые ввел пользователь.

GET-запросы и функция include

5. Отправьте GET-запрос с помощью ссылки следующего вида (см. пример). Запишите номер страницы в переменную **\$page**.

```
'domain.ru/index.php?page=3'
```

6. Отправьте GET-запрос на показ определенной страницы сайта. Запишите адрес страницы в переменную **\$page**. С помощью функции include() сделайте так, чтобы к странице index.php подключалась страница из переменной **\$page**. Создайте 5 файлов с различными названиями и с их помощью проверьте работу скрипта.

```
'domain.ru/index.php?page=test.php'
```

Атрибуты value и placeholder

7. Спросите имя пользователя с помощью формы. Результат запишите в переменную **\$name**. Сделайте так, чтобы после отправки формы значения ее полей не пропадали.
8. Спросите у пользователя имя, а также попросите его ввести сообщение (textarea). Сделайте так, чтобы после отправки формы значения ее полей не пропадали.

Массив \$_SERVER

9. Ознакомьтесь с элементами суперглобального массива **\$_SERVER**.

```
$_SERVER['REQUEST_URI']  
$_SERVER['SERVER_NAME']  
$_SERVER['DOCUMENT_ROOT'].
```

Работа с датами в PHP

Timestamp: time и mktime

1. Выведите текущее время в формате timestamp.
2. Выведите 1 января 2025 года в формате timestamp.
3. Выведите 31 декабря **текущего года** в формате timestamp. Скрипт должен работать независимо от года, в котором он запущен.
4. Найдите количество секунд, прошедших с 13:12:59 15-го марта 2000 года до настоящего момента.
5. Найдите количество секунд **\$sec**, прошедших с 7:23:48 1-го числа текущего месяца до настоящего момента. Найдите количество минут, часов и дней, содержащихся в **\$sec** (не одновременно, а последовательно: сначала минуты, потом часы и т.д.).

Функция date

6. Выведите на экран текущий **год, день, месяц, час, минуту, секунду**.
7. Выведите текущую дату-время в форматах '2013-12-31', '31.12.2013', '31.12.13', '12:59:59'.
8. С помощью функций **mktime** и **date** выведите 13 февраля 2015 года в формате '31.12.2013'.
9. Создайте массив дней недели **\$week**. Выведите на экран название *текущего дня недели* с помощью массива **\$week** и функции date. Узнайте какой день недели был 06.06.2006, в ваш день рождения.
10. Создайте массив месяцев **\$month**. Выведите на экран название *текущего месяца* с помощью массива **\$month** и функции date.
11. Найдите номер текущего дня от начала месяца.
12. Найдите количество дней в текущем месяце. Скрипт должен работать независимо от месяца, в котором он запущен.
13. Сделайте поле ввода, в которое пользователь вводит год (4 цифры), а скрипт определяет високосный ли год.
14. Сделайте форму, которая спрашивает дату в формате '31.12.2013'. С помощью функций **explode** и **mktime** переведите эту дату в формат timestamp. Узнайте день

недели (словом) за введенную дату.

15. Сделайте форму, которая спрашивает дату в формате '2013-12-31'. С помощью функций **explode** и **mktime** переведите эту дату в формат timestamp. Узнайте месяц (словом) за введенную дату.
16. Сделайте форму, которая спрашивает у пользователя дату в формате '2013-12-31T12:59:59'. С помощью функций **explode** и **mktime** переведите эту дату в формат timestamp. Узнайте месяц (словом) за введенную дату. Подсказка: explode следует применить дважды.

Сравнение дат

17. Сделайте форму, которая спрашивает две даты в формате '2013-12-31'. Первую дату запишите в переменную **\$date1**, а вторую в **\$date2**. Сравните, какая из введенных дат больше. Выведите ее на экран.

На strtotime

18. Дана дата в формате '2013-12-31'. С помощью функции strtotime преобразуйте ее в формат '31-12-2013'.
19. Сделайте форму, которая спрашивает дату-время в формате '2013-12-31T12:13:59'. С помощью функции strtotime преобразуйте ее в формат '12:13:59 31.12.2013'.

Прибавление и отнимание от дат

20. Прибавьте к дате 31.12.2013 один день. Результат выведите в том же формате.
21. В переменной **\$date** лежит дата в формате '2013-12-31'. Прибавьте к этой дате 2 дня, 1 месяц и 3 дня, 1 год. Отнимите от этой даты 3 дня.

Задачи

22. Передайте в адресную строку GET-параметр **date**, в котором будет записана дата в формате '2013-12-31'. Преобразуйте ее в формат '31.12' (день.месяц).
23. Найдите количество дней, часов, минут, секунд, прошедших с 5:49:59 1-го числа текущего месяца до настоящего момента.
24. Узнайте сколько дней осталось до Нового Года.
25. Дан GET-параметр date, в который вводится год в формате '2013'. Посчитайте

сколько воскресений в этом году приходится на первое число месяца.

26. Найдите сколько воскресений будет в определенном году **\$year**.
27. Сделайте форму с одним полем ввода, в которое пользователь вводит год. Найдите все **пятницы 13-е** в этом году. Результат выведите в виде списка дат.
28. Дана дата в формате '2013-12-31'. Определите знак зодиака, к которому относится данная дата.
29. Дан GET-параметр date, в который вводится год в формате '2013'. Узнайте, какой это будет год по восточному календарю.
30. Узнайте какой день недели был 100 дней назад.
31. Сделайте массив праздников. Если пользователь заходит на сайт в праздник — поздравьте его с этим праздником.
32. Рассчитайте дату пасхи в заданном году. Алгоритм ее расчета найдите в поисковых системах.
33. Сделайте сайт-гороскоп: пользователь заходит на сайт и вводит дату своего рождения. Сайт определяет его знак зодиака и выдает гороскоп на день захода пользователя на сайт. Гороскопы храните в многомерном массиве, на неделю вперед.

Функции работы с массивами

На count и range

1. Создайте массив, заполненный числами от 1 до 100.
2. Дан массив **\$arr**. Подсчитайте количество элементов этого массива.
3. Дан массив **\$arr**. Выведите на экран первый и последний элемент данного массива.

На array_sum и array_product

4. Дан массив **\$arr**. С помощью функций `array_sum` и `count` найдите среднее арифметическое элементов данного массива.
5. Создайте массив, заполненный числами от 1 до 300. Найдите произведение элементов данного массива.

На array_merge и array_slice

6. Даны два массива: первый с элементами '1', '2', '3', второй с элементами 'a', 'b', 'c'. Сделайте из них массив с элементами '1', '2', '3', 'a', 'b', 'c'.
7. Даны массив с элементами '1', '2', '3', '4', '5'. С помощью функции `array_slice` создайте массив **\$b** с элементами '2', '3', '4'.

На array_keys, array_values, array_combine

8. Дан массив `'green'=>'зеленый', 'blue'=>'голубой', 'red'=>'красный'`. Запишите в массив **\$key** английские названия цветов, а в **\$values** – русские.
9. Даны два массива: `'green', 'blue', 'red'` и `"зеленый", "голубой", "красный"`. Создайте с их помощью массив `'green'=>'зеленый', 'blue'=>'голубой', 'red'=>'красный'`. Используйте функцию `array_combine`.

На array_flip, array_count_values, array_reverse, array_unique

10. Дан массив `'green'=>'зеленый', 'blue'=>'голубой', 'red'=>'красный'`. Поменяйте местами ключи и значения.
11. Дан массив с элементами '1', '2', '3', '4', '5'. Создайте массив с элементами '5', '4', '3', '2', '1'.
12. Дан массив с элементами 'a', 'b', 'c', 'b', 'a'. Удалите из него повторяющиеся

элементы.

13. Дан массив с элементами 'a', 'b', 'c', 'b', 'a'. Подсчитайте сколько раз встречается каждая из букв.

На сортировку, shuffle и array_rand

14. Дан массив '3'=>'a', '1'=>'c', '2'=>'e', '4'=>'b'. Попробуйте на нем различные типы сортировок.
15. Дан массив \$arr. Перемешайте его элементы в случайном порядке.
16. Дан массив с элементами '1', '2', '3', '4', '5'. Выведите на экран случайный элемент данного массива.

На array_map и array_walk

17. Дан массив с элементами '1', '2', '3', '4', '5'. Создайте новый массив, в котором будут лежать квадратные корни данных элементов.
18. Дан массив с элементами 'php', '<i>html</i>'. Создайте новый массив, в котором из элементов будут удалены теги.
19. Дан массив с элементами ' a ', ' b ', ' c '. Создайте новый массив, в котором будут данные элементы без концевых пробелов.

На array_chunk и array_pad

20. Дан массив с элементами '1', '2', '3'. Сделайте из него массив с элементами '1', '2', '3', '0', '0', '0'.
21. Создайте массив, заполненный целыми числами от 1 до 20. С помощью функции array_chunk разбейте этот массив на 5 подмассивов ('1', '2', '3', '4'; '5', '6', '7', '8' и т.д.).

Работа с функциями в PHP

Простые функции

1. Сделайте функцию, которая возвращает квадрат числа. Число передается параметром.
2. Сделайте функцию, которая возвращает сумму двух чисел.
3. Сделайте функцию, которая отнимает от первого числа второе и делит на третье.
4. Сделайте функцию, которая принимает 3 параметра: день, месяц и год, а возвращает дату в формате 30-01-2013.

На параметры по умолчанию

5. Сделайте функцию, которая будет возводить число в куб. По умолчанию функция должна возводить в куб число 3.

На static и global

6. Сделайте функцию, которая будет подсчитывать количество раз, которое ее вызвали.

На html

7. Сделайте функцию, которая подключает картинку. Параметром функция должна принимать путь к картинке, а возвращать — тег ``.
8. Сделайте функцию, которая подключает CSS на сайт.

На строки

9. Сделайте функцию, которая принимает параметром строку, а возвращает первые N символов этой строки (N – это второй параметр). По умолчанию функция должна добавлять троеточие в конце возвращаемой строки, а третий параметр должен регулировать это поведение (true – добавить троеточие, false – не добавить). Функция должна корректно работать с кириллицей.
10. Сделайте функцию, которая принимает строку на русском языке, а возвращает ее **транслит**. Сделайте функцию, которая выполняет обратную операцию.
11. Сделайте функцию, которая возвращает множественное или единственное число существительного. Пример: 1 яблоко, 2 (3, 4) яблока, 5 яблок. Функция первым

параметром принимает число, а следующие 3 параметра — форма для единственного числа, для чисел два, три, четыре и для чисел, больших четырех.

На даты

12. Сделайте функцию, которая принимает 3 параметра: день, месяц и год, а возвращает дату в различных форматах (30.01.2013, 30-01-2013, 2013-01-30). Формат задается четвертым параметром.
13. Сделайте функцию, которая берет дату в формате sql (2013-01-31) и делает из нее нормальный формат (30.01.2013, 30-01-2013, 2013-01-30).
14. Сделайте функцию, которая принимает дату в нормальном формате (31.12.2013), а возвращает в sql формате. Совет: воспользуйтесь функцией explode.
15. Сделайте функцию, которая принимает два параметра — месяц и год, а возвращает последнее число месяца (подсказка: покопайтесь в функции date, есть готовая реализация).
16. Сделайте функцию, которая принимает день и месяц, а возвращает какой в эту дату **праздник**. Массив праздников храните прямо в функции.
17. Сделайте функцию, которая находит разницу между двумя датами: количество лет, месяцев, дней, часов, минут, секунд. Результат должен выводиться в виде массива с ключами **y, m, d, h, i, s**.
18. Сделайте функцию, которая принимает параметром число от 1 до 12, а возвращает название месяца на русском языке.
19. Сделайте функцию, которая принимает параметром число от **1 до 7**, а возвращает день недели на русском языке.
20. Сделайте функцию, которая возвращает текущую дату в формате '31 января, пятница'. Вынесите код, который преобразует число в название месяца по-русски в родительном падеже **в отдельную функцию**. Аналогично поступите с кодом, который преобразует число в день недели по-русски.
21. Сделайте функцию, которая принимает параметром название дня недели на английском языке, а возвращает число от 1 до 12.
22. Сделайте функцию, которая принимает параметром число, а возвращает день недели на определенном языке. Язык должен указываться вторым параметром ('ru'

— русский, 'en' — англ., 'ge' — немецкий). По умолчанию функция должна возвращать русское название.

На ссылки

23. Сделайте функцию, которая делает ссылку. Она должна принимать два параметра: адрес ссылки и текст ссылки и возвращать тег

```
<a href='адрес_ссылки'>текст_ссылки</a>
```

24. Сделайте функцию, которая делает ссылку в случае, если дата, заданная в параметре больше или равна текущей дате. Иначе функция должна вернуть обычный текст.

Такая задача часто нужна, если вы реализуете автоматическое добавление страниц на сайте (они появляются в заданную дату и в эту же дату должны появиться и ссылки на эти страницы!).

25. Сделайте функцию, которая делает ссылку с определенным классом: первый класс она добавляет в случае, если адрес ссылки совпадает с текущей страницей, а второй класс - когда не совпадает. **Совет: адрес страницы лежит в `$_SERVER['REQUEST_URI']`.**

Функция должна вернуть примерно следующее:

```
<a href='адрес_ссылки' class='класс'>текст_ссылки</a>.
```

Такая задача нужна очень часто, практически на каждом сайте — для выделения ссылки в меню, на которой мы находимся сейчас.

26. Сделайте функцию, которая преобразует число, переданное ей в параметрах, в слово. Вторым параметром она принимает массив такого рода: `array('1'=>'Работает!', '0'=>'Пусто!')`. Если функции передали число 1 — она вернет слово 'Работает!', а если 0 — 'Пусто!' (в соответствии с массивом). Массив, естественно должен быть любым.

Такая задача часто возникает при обработке данных из БД. В БД мы можем хранить данные в виде нулей и единиц, либо коротких слов, а пользователю хотелось бы раскрыть смысл этих нулей и единиц.

На списки и таблицы

27. Сделайте функцию, которая выводит список (`ul` или `ol` — это должно регулироваться параметром). Пункты списка должны передаваться в массиве.

Пример: передадим этой функции массив **array('один', 'два', 'три')** и то, что мы хотим список **ol**.

Она должна вернуть:

```
<ol><li> один </li><li> два </li><li> три </li></ol>
```

28. Сделайте функцию, которая выводит список, в котором каждый N-ый пункт будет иметь заданный класс.

Пусть N равно 2, тогда каждый второй элемент должен иметь заданный класс.

Пример для массива **array('один', 'два', 'три', 'четыре')** и класса **last**:

```
<ol><li> один </li><li class='last'> два </li><li> три </li><li class='last'> четыре </li></ol>
```

29. Сделайте функцию, которая принимает двухмерный массив, а выводит html-таблицу.

Пример: для массива

```
array('1'=>array('1','2','3'),'2'=>array('4','5','6'),'3'=>array('7','8','9'), ),
```

должна получится такая таблица:

30. 1	31. 2	32. 3
33. 4	34. 5	35. 6
36. 7	37. 8	38. 9

39. Сделайте функцию, аналогичную предыдущей, которая принимает **одномерный** массив. Функция сама должна разбить одномерный массив на двухмерные и добавить недостающие элементы в последнем подмассиве (пустые строки или). Воспользуйтесь функциями **array_chunk** и **array_pad**.

40. Сделайте функцию, аналогичную предыдущей, которая четному ряду таблицы добавляет класс **odd**, а нечетному — класс **even**.

Продвинутая работа с формами

Checkbox

1. Спросите у пользователя имя с помощью формы. Сделайте чекбокс: если он отмечен, то поприветствуйте пользователя, если не отмечен — попрощайтесь с пользователем.
2. Спросите у пользователя, какие из языков он знает: html, css, php, javascript. Выведите на экран те языки, которые знает пользователь. Если пользователь не отметил ни один язык — выведите на экран сообщение об этом.

Radio

3. Спросите у пользователя знает ли он PHP с помощью двух radio-кнопок. Выведите результат на экран. Сделайте так, чтобы по умолчанию один из вариантов был уже отмечен.
4. Спросите у пользователя его возраст с помощью нескольких radio-кнопок. Варианты ответа сделайте такими: менее 20 лет, 20-25, 26-30, более 30.

Select и multi-select

5. Спросите у пользователя его возраст с помощью select. Варианты ответа сделайте такими: менее 20 лет, 20-25, 26-30, более 30.
6. Спросите у пользователя с помощью мультиселекта, какие из языков он знает: html, css, php, javascript. Выведите на экран те языки, которые знает пользователь.

Задачи

7. Сделайте функцию, которая создает инпут type text или password. Функция должна иметь следующие параметры: type, name, value, placeholder.
8. Модифицируйте функцию из предыдущей задачи так, чтобы она сохраняла значение инпута после отправки.
9. Сделайте функцию, которая создает **textarea**. Функция должна иметь следующие параметры: name, value, placeholder.
10. Модифицируйте функцию из предыдущей задачи так, чтобы она сохраняла значение **textarea** после отправки.

11. Сделайте функцию, которая создает чекбокс. Если чекбокс не отмечен — функция должна отправлять 0 (то есть нужно сделать hidden инпут), если отмечен — 1.
12. Дан чекбокс. Сделайте так, чтобы после отправки формы его значение сохранялось (отмечен или не отмечен).
13. Напишите функцию, которая создает чекбокс и сохраняет его значение после отправки.
14. Сделайте функцию, которая будет создавать **селект**. Функция должна принимать многомерный массив такого типа:

```
$arr = array(  
    0=>array('value'=>'php', 'text'=>'Язык PHP'),  
    0=>array('value'=>'html', 'text'=>'Язык HTML'),  
)
```

15. Модифицируйте предыдущую функцию так, чтобы она сохраняла выбранный элемент после отправки.
16. Сделайте 3 селекта, которые позволяют выбирать дату: селект с днями от 1 до 31, селект с русскими названиями месяцев, селект с годами. Реализуйте функцию, которая будет создавать подобный селект. Функция должна принимать параметром диапазон годов, за который следует показывать дату.
17. Модифицируйте предыдущую задачу так, чтобы по умолчанию была выбрана текущая дата и селекты сохраняли выбранные значения после отправки.
18. Сделайте тест с различными вариантами ответов, из которых пользователь может выбрать только один. После отправки теста скрипт должен проверять правильность ответов на вопросы и выводить результаты. Также он должен подсчитывать процент правильно решенных заданий. Придумайте 5-10 вопросов для этого теста.

Работа с базой данных SQL

Все задачи будут по данной таблице workers (если не сказано иное).

id	name	age	salary
1	Дима	23	400
2	Петя	25	500
3	Вася	23	500
4	Коля	30	1000
5	Иван	27	500
6	Кирилл	28	1000

SELECT

1. Выбрать работника с id = 3. Узнать его имя.
2. Выбрать работников с зарплатой 1000\$.
3. Выбрать работников в возрасте 23 года.
4. Выбрать работников с зарплатой более 400\$.
5. Выбрать работников с зарплатой равной или большей 500\$.
6. Выбрать работников с зарплатой **НЕ** равной 500\$.
7. Выбрать работников с зарплатой равной или меньшей 900\$.
8. Узнайте зарплату и возраст Васи.

OR и AND

1. Выбрать работников в возрасте от 25 (не включительно) до 28 лет (включительно).
2. Выбрать работника Петю.
3. Выбрать работников Петю и Васю.
4. Выбрать всех, кроме работника Петя.
5. Выбрать всех работников в возрасте 27 лет или с зарплатой 1000\$.
6. Выбрать всех работников в возрасте от 23 лет (включительно) до 27 лет (не включительно) или с зарплатой 1000\$.
7. Выбрать всех работников в возрасте от 23 лет до 27 лет или с зарплатой от 400\$ до 1000\$.
8. Выбрать всех работников в возрасте 27 лет или с зарплатой не равной 400\$.

INSERT

1. Добавьте нового работника Никиту, 26 лет, зарплата 300\$. Воспользуйтесь первым синтаксисом.
2. Добавьте нового работника Светлану с зарплатой 1200\$. Воспользуйтесь вторым синтаксисом.

DELETE

1. Удалить работника с id=7.
2. Удалить Колю.
3. Удалить всех работников, у которых возраст 23 года.

Задачи

1. Передайте через GET-параметр id работника, которого следует вывести на экран. Выведите его имя, возраст и зарплату.
2. Выведите на экран таблицу всех работников (см. задачу №4, только без колонки 'удаление').
3. Сделайте форму добавления нового работника.
4. Сделайте в таблице всех работников ссылку 'удалить'. По нажатию на эту ссылку из БД должна удаляться запись с этим id (его следует передавать через GET-параметр del_id).

id	имя	возраст	зарплата	удаление
1	Дима	23	400	удалить
2	Петя	25	500	удалить
3	Вася	23	500	удалить
4	Коля	30	1000	удалить
5	Иван	27	500	удалить
6	Кирилл	28	1000	удалить

Верните таблицу workers в исходное состояние.

id	name	age	salary
1	Дима	23	400

2	Петя	25	500
3	Вася	23	500
4	Коля	30	1000
5	Иван	27	500
6	Кирилл	28	1000

Update

1. Поставьте Васе зарплату в 200\$.
2. Работнику с id=4 поставьте возраст 35 лет.
3. Всем, у кого зарплата 500\$ сделайте ее 700\$.
4. Работникам с id больше 2 и меньше 5 включительно поставьте возраст 23.
5. Поменяйте Васю на Женю и прибавьте ему зарплату до 900\$.

На LIMIT

1. Из таблицы **workers** достать первые 6 записей.
2. Из таблицы **workers** достать записи со вторую по седьмую.

На ORDER BY

1. Из таблицы **workers** достать всех работников и отсортировать их по **возрастанию** зарплаты.
2. Из таблицы **workers** достать всех работников и отсортировать их по **убыванию** зарплаты.
3. Из таблицы **workers** достать работников со второго по шестого и отсортировать их по возрастанию возраста.
4. В таблице **workers** подсчитать всех работников (команда COUNT).
5. В таблице **workers** подсчитать всех работников с зарплатой 300\$.

На LIKE

1. Создайте таблицу pages.

id	Athor (автор)	Article (статья)
1	Петров	В своей статье рассказывает о машинах .
2	Иванов	Написал статью об инфляции.
3	Сидоров	Придумал новый химический

		элемент.
4	Осокина	Также писала о машинах .
5	Ветров	Написал статью о том, как разрабатывать элементы дизайна.

2. В таблице **pages** найти строки, в которых **фамилия** автора заканчивается на «ов».
3. В таблице **pages** найти строки, в которых есть слово «элемент» (искать только по колонке article).
4. В таблице **workers** найти строки, в которых **возраст** работника начинается с числа 3, а далее идет только одна цифра.
5. В таблице **workers** найти строки, в которых **имя** работника заканчивается на «я».

Работа с сессиями и cookie (куки)

Работа с сессиями

1. Сделайте две страницы: index.php и test.php. При заходе на index спросите с помощью формы страну пользователя, запишите ее в **сессию**. При заходе на test.php выведите страну пользователя.
2. Запишите в сессию время захода пользователя на сайт. При обновлении страницы выводите сколько секунд назад пользователь зашел на сайт.
3. Спросите у пользователя email с помощью формы. Затем сделайте так, чтобы в другой форме (имя, фамилия, пароль, email) при ее открытии поле email было автоматически заполнено.
4. Сделайте счетчик обновления страницы пользователем. Данные храните в сессии. Скрипт должен выводить на экран количество обновлений. При первом заходе на страницу он должен вывести сообщение о том, что вы еще не обновляли страницу.
5. Сделайте две страницы: index.php и form.php. При заходе на index спросите с помощью формы **город** и **возраст** пользователя. На form.php сделайте форму с полями 'Имя', 'Возраст', 'Город'. При заходе на form.php сделайте так, чтобы поля 'Возраст' и 'Город' уже были заполнены.
6. Добавьте в предыдущую задачу страницу logout.php. При заходе на нее разрушайте сессию пользователя.
7. Предложите пользователю выбрать любимый цвет с помощью выпадающего списка (красный, зеленый, синий). Затем сделайте небольшую витрину с товарами трех цветов. При заходе пользователя на эту витрину он должен увидеть товары своего любимого цвета.
8. С помощью витрины, созданной в предыдущей задаче, реализуйте корзину. Корзина должна хранить все товары, выбранные пользователем и суммировать их стоимости.
9. Реализуйте **тест** по принципу 'одна страница сайта — одна задача'. Запомните результаты ответов пользователя в сессию.

Работа со cookie

10. Сделайте две страницы: `index.php` и `test.php`. При заходе на `index` спросите с помощью формы страну пользователя, запишите ее в **куки** с именем `country`. При заходе на `test.php` выведите страну пользователя.
11. Удалите куку с именем `country`.
12. Установите куку с именем **age**. Запишите туда случайное число от 10 до 70 (с помощью `mt_rand`). Сделайте так, чтобы эта кука установилась на 1 час, на 3 часа, на 1 день, на год, на 10 лет, до конца текущего дня, до конца текущего года.

Задачи

13. Напишите оболочку над cookie. Оболочка должна представлять собой набор функций: сохранение куки, удаление куки, редактирование куки.
14. Сделайте на сайте 5 картинок с товарами. Реализуйте корзину. Под каждой картинкой должна быть ссылка 'положить в корзину'. По нажатию на эту ссылку этот товар должен занестись в корзину (сессия), также должна увеличиться общая сумма, которую должен заплатить пользователь (сумма также должна быть указана под картинками с товарами).
15. Сделайте счетчик посещения сайта посетителем. Каждый раз, заходя на сайт, он должен видеть надпись: 'Вы посетили наш сайт % раз!'.
16. Покажите пользователю баннер с кнопкой 'Не показывать больше!'. Если он нажмет на эту кнопку — не показывайте ему баннер в течении месяца.
17. Запомните дату последнего посещения сайта пользователем. При заходе на сайт напишите ему, сколько дней он не был на вашем сайте.
18. Спросите дату рождения пользователя. При следующем заходе на сайт напишите сколько дней осталось до его дня рождения. Если сегодня день рождения пользователя — поздравьте его!
19. Реализуйте выбор дизайна сайта пользователем. Сделайте несколько дизайнов сайта. Пользователь может выбрать один из дизайнов с помощью выпадающего списка. Этот выбор будет сохранен в куки и пользователь, заходя на сайт, всегда будет видеть один и тот же дизайн.

20. Реализуйте игру 'Казино' с помощью куки и сессий. Суть игры такая: пользователю дается сумма в 100\$, в казино выпадает цифра от одного до 6 (кубик). Игрок может поставить на одну, две или три цифры. На одну цифру — ставка умножается на 6, на две — на 3, на три — на 2. В случае выигрыша сумма записывается в кошелек игрока, в случае проигрыша — отнимается. Кошелек храните в куках, чтобы игрок мог продолжить игру через какое-то время.

Объектно-ориентирование программирование

Работа с классами

1. Сделайте класс **Power**, который возводит число в нужную степень (не используя стандартную функцию `pow`). Класс должен иметь следующие методы: 2, 3, 4, 5 степени.
2. Сделайте класс-калькулятор. Класс должен иметь следующие методы: сложение, вычитание, умножение, деление. Каждый метод должен принимать параметром два числа.
3. Сделайте класс **Sqrt**, который извлекает из числа корень нужной степени (используя `pow`: `pow(4, 0.5)` – это квадратный корень). Класс должен иметь следующие методы: корни 2, 3, 4, 5 степени.
4. Сделайте класс-оболочку над `$_SERVER`. Класс должен иметь следующие методы: `REQUEST_URI`, `IP`, `DOCUMENT_ROOT`, `HTTP_HOST`, `HTTP_USER_AGENT`, `QUERY_STRING`.

Private и public

5. Напишите класс-сумматор. Класс должен иметь следующие методы: сумма двух чисел, сумма квадратов двух чисел, сумма кубов двух чисел и так далее до 5-той степени. Класс должен иметь `private` методы возведения в степень (без `pow`) и `public` методы суммирования.
6. Напишите класс-оболочку над `cookie`. Оболочка должна представлять собой набор функций: сохранение куки, удаление куки, редактирование куки, считывание куки. По умолчанию кука должна устанавливаться на 1 год.
7. Сделайте класс-геометрический-калькулятор. Класс должен иметь следующие методы: площадь квадрата, площадь прямоугольника, объем куба, объем параллелепипеда, длина окружности через радиус, площадь круга, объем шара. Класс должен иметь `private` свойство — число π .

__construct

8. Сделайте класс-валидатор `$_REQUEST`, `$_GET`, `$_POST`. Класс должен иметь private методы обрезание пробелов **del_gaps** и удаление тегов **del_tags** (оболочку над стандартными) и public метод `protect`, который получает массив, обрабатывает его `del_gaps` и `del_tags` и возвращает обработанный массив.
Класс должен иметь public свойства `request`, `post`, `get`, в которых хранятся соответствующие обработанные массивы. Обработка должна происходить при инициализации класса (то есть в `__construct`).
9. Напишите оболочку над сессиями. Оболочка должна представлять собой набор функций: сохранение в сессию, считывание из сессии, удаление всех переменных сессии. Сессия должна стартовать в функции `__construct`.

На наследование и использование других классов

10. Напишите класс-сумматор. Класс должен иметь следующие методы: сумма двух чисел, сумма квадратов двух чисел, сумма кубов двух чисел и так далее до 5-той степени. Класс должен наследовать методы возведения в степень от класса **Power** и иметь public методы суммирования.
11. Напишите класс-сумматор. Класс должен иметь следующие методы: сумма двух чисел, сумма квадратов двух чисел, сумма кубов двух чисел и так далее до 5-той степени. Класс должен брать методы возведения в степень от **внешнего** класса **Power** (и использовать их через `$this->power->метод`) и иметь public методы суммирования.
12. Напишите класс-сумматор. Класс должен иметь следующие методы: сумма двух чисел, сумма **квадратных корней** двух чисел, сумма **кубических корней** двух чисел и так далее до 5-той степени. Класс должен наследовать методы взятия корня от класса **Sqrt** и иметь public методы суммирования.
13. Напишите класс-сумматор. Класс должен иметь следующие методы: сумма двух чисел, сумма **квадратных корней** двух чисел, сумма **кубических корней** двух чисел и так далее до 5-той степени. Класс должен брать методы взятия корня от **внешнего** класса **Power** и иметь public методы суммирования.

Задачи

14. Сделайте класс-оболочку над БД. Методы класса продумайте самостоятельно.
15. Сделайте класс, который находит сумму квадратов всех элементов массива и извлекает квадратный корень, сумму кубов и извлекает кубический корень и так далее до 5-той степени. Класс должен наследовать от класса **Sqrt**, для возведения в степень использовать стандартную функцию `pow`.
16. Сделайте класс для работы с датами. Класс должен уметь находить разницу между двумя датами, принимать дату в `sql`-формате, а возвращать в заданном, принимать дату в формате '31.12.2013', а возвращать в заданном. Также класс должен определять текущий день недели и месяц (словом, по-русски) и иметь для этого соответствующие методы.

Класс должен иметь `public` свойство **today**, в котором хранится текущая дата (заполняется в `__construct`). Класс должен иметь `public` свойство **weekday**, в котором хранится текущий день недели (по-русски). Класс должен иметь `public` свойство **month**, в котором хранится текущий месяц (по-русски).

Класс должен иметь и использовать `private` метод, который принимает количество секунд `$num`, а возвращает массив, в котором содержится количество лет, месяцев, дней, часов, минут, секунд в `$num`.

Добавьте несколько методов на свой вкус.

17. Сделайте класс, который подключает внешние файлы к `html` странице, такие как `CSS` и `JS`. Класс должен иметь методы `getCss` и `getJs`.
18. Сделайте класс, который добавляет тег мета на сайт (кодировка, дескрипцион, кейвордс).
19. Сделайте класс, который создает формы. Класс должен иметь следующие методы: создание инпута, создание `textarea`, создание селекта, создание чекбокса (вместе со скрытым), создание `radio`.
20. Сделайте класс, который принимает массив, а возвращает `html`-таблицу. Класс также должен принимать количество столбцов в этой таблице и сам разбивать массив нужным образом.

Функции работы с папками и файлами

Данный раздел не полон, будет дописан в следующем издании.

file_put_contents, file_get_contents, rename, copy, filesize, unset, file_exists

1. Создайте файл 'test.txt' и запишите в него фразу 'Привет, мир!'.
2. Считайте данные из файла 'test.txt' и выведите их на экран.
3. Переименуйте файл 'test.txt' в 'mir.txt'.
4. Создайте копию файла 'mir.txt' и назовите ее 'world.txt'.
5. Определите размер файла 'world.txt'. Выведите его на экран.
6. Удалите файл 'world.txt'.
7. Проверьте существование файлов 'world.txt' и 'mir.txt'.

Работа с file, PHP_EOL

1. Создайте массив с элементами 'Иванов', 'Петров', 'Сидоров'. Запишите содержимое этого массива в файл 'file.txt' так, чтобы все элементы располагались на разных строках. Воспользуйтесь константой PHP_EOL.
2. Считайте содержимое файла 'file.txt' с помощью функции file. Запишите результат в массив \$file.

Загрузка файлов на сервер

На scandir

1. Создайте папку с названием 'folder'.
2. Выведите на экран название всех файлов из папки.
3. Выведите на экран название всех файлов с расширением txt из папки.
4. Дана папка с CSS файлами. Считайте название этих файлов и подключите эти CSS файлы к сайту.

fopen, fread, fwrite

Задачи