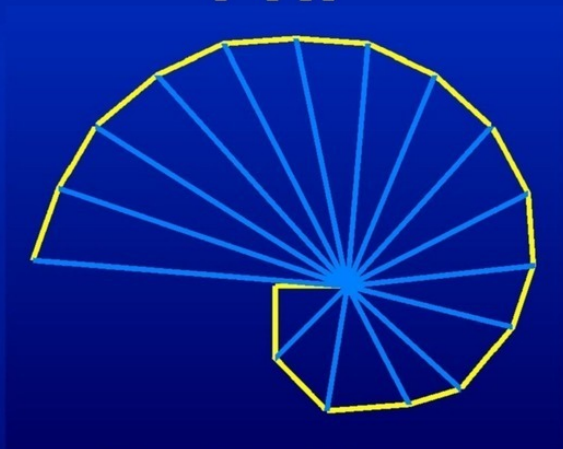


**РИСОВАНИЕ  
ЦВЕТНЫХ  
ИЗОБРАЖЕНИЙ  
С ПОМОЩЬЮ  
PHR**



**Джеймс Уэллс**

**Джеймс Уэллс**

**Рисование цветных  
изображений с помощью РНР.  
Программирование  
для развлечения**

*[http://www.litres.ru/pages/biblio\\_book/?art=68617058](http://www.litres.ru/pages/biblio_book/?art=68617058)  
ISBN 9785005933225*

**Аннотация**

Научитесь программировать! Это увлекательно! Книга познакомит вас с языком программирования на РНР. Даже если вы не собираетесь стать программистом, после чтения книги вы научитесь «рисовать» картины с помощью РНР-кода, даже если вы не художник. Или вы можете использовать примеры из книги в качестве упражнений для вашего мозга. Это интереснее чем разгадывать кроссворды! Купите книгу сейчас, чтобы повысить свою креативность!

# Содержание

Для кого эта книга?	5
Что такое компьютерное программирование?	6
Программное обеспечение, которое вам понадобится	8
Редактирование системных переменных	12
Запуск PHP	17
Тип данных	21
Переменные	22
Массивы	28
If-else оператор	30
Циклы	32
While Цикл	33
Do While Цикл	35
For Цикл	37
Цикл Foreach (Для каждого)	40
Функция	43
Передача переменных по значению или по ссылке	45
Как рисовать изображения	47
Как рисовать линии	50
Конец ознакомительного фрагмента.	60

**Рисование цветных  
изображений  
с помощью РНР  
Программирование  
для развлечения**

**Джеймс Уэллс**

© Джеймс Уэллс, 2022

ISBN 978-5-0059-3322-5

Создано в интеллектуальной издательской системе Ridero

# Для кого эта книга?

Эта книга научит вас создавать красочные изображения с помощью PHP-кода. PHP – это компьютерный язык для веб-разработчиков. Это самый простой язык для изучения, но это очень мощный и популярный компьютерный язык. Многие известные социальные сайты построены на PHP.

Если вы хотите стать веб-разработчиком, эта книга для вас.

Если вы не собираетесь становиться веб-разработчиком или программистом, вы можете использовать эту книгу в качестве умственного упражнения или для создания красочных изображений для искусства или развлечения.

# Что такое компьютерное программирование?

Вы можете представить компьютерное программирование как упаковку. Когда вам нужно упаковать в коробки много разных предметов, вы должны выбрать коробку оптимального размера для каждого предмета. В процессе вы можете перекладывать вещи из одной коробки в другую и перемещать коробки по своему усмотрению.

Вы можете стать хорошим программистом, если сможете переставить гаражные ящики так, чтобы освободилось место для вашей машины.

В этой книге вы изучите компьютерный язык РНР, один из самых простых для программирования.

Когда вы закончите читать эту книгу, вы сможете писать РНР-код для создания фантастических красочных фигур, которые поднимут вам настроение. Вы можете увидеть пример ниже.

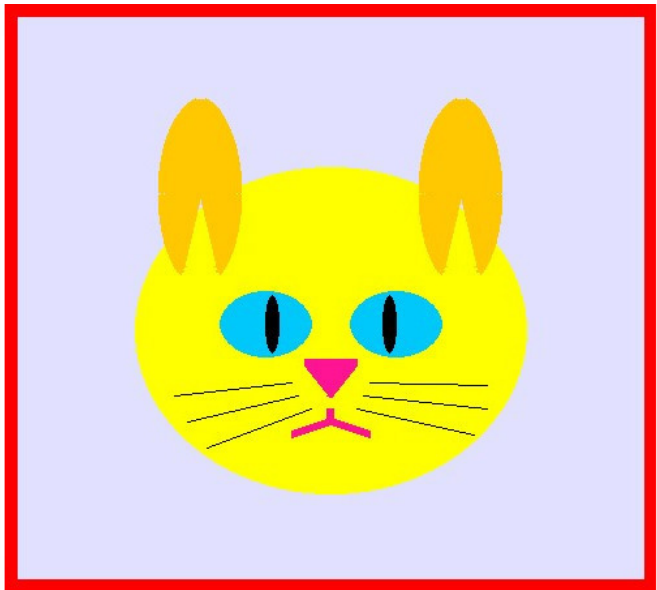


Рис. 1. Картинка кота создана с помощью PHP-кода.

Вы можете скачать все файлы PHP и все цветные изображения отсюда:[http://learn-coding.today/gd/php\\_img.zip](http://learn-coding.today/gd/php_img.zip)

# **Программное обеспечение, которое вам понадобится**

Для программирования на PHP необходимо установить PHP и notepad++. Оба программного обеспечения бесплатны.

Вам понадобится веб-браузер. Я рекомендую Chrome или Firefox. Введите в Google «Скачать Google Chrome для Windows 10 64-бит», и вы найдете ссылку.

Вы можете скачать PHP здесь: <https://www.php.net/downloads.php>.

См. Рис. 2.



## Current Stable PHP 8.1.7 (Changelog)

---

- [php-8.1.7.tar.gz \(sig\)](#) [19,252Kb]  
sha256: 5f0b422a117633c86d48d028934b8dc0783
- [php-8.1.7.tar.bz2 \(sig\)](#) [15,101Kb]  
sha256: b816753eb005511e695d90945c27093c323
- [php-8.1.7.tar.xz \(sig\)](#) [11,444Kb]  
sha256: f042322f1b5a9f7c2dec84b7086ef67689
- [Windows downloads](#)

Рисунок 2. Скачать PHP.

Если у вас есть ПК с Windows, перейдите к загрузкам Windows на веб-странице загрузки PHP. См. Рис. 3.

## VS16 x64 Thread Safe (2022-Apr-12 18:12:52)

- [Zip](#) [29.33MB]

sha256: 78b724971fe14a4087c32f839586e54188362f892

- [Debug Pack](#) [23.75MB]

sha256: 767aa12df5a83992f63d3223f0e1a9b375c66e1cc

- [Development package \(SDK to develop PHP extensions\)](#) [

sha256: 206bb13b71511bc4bd08b63f4ef79a3a10c18078

Рисунок 3. Пакеты PHP.

Выберите и загрузите пакет Thread Safe в виде файла Zip. Создайте папку php на своем основном жестком диске и распакуйте пакет PHP в эту папку. На моем ПК это C:/php. См. Рис. 4

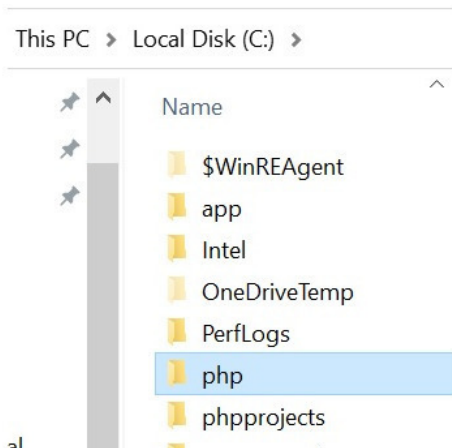


Рисунок 4. Папка PHP на диске C

# Редактирование системных переменных

В левом нижнем углу экрана найдите окно поиска. Введите «Настройки», найдите значок «Настройки» и щелкните по нему. См. Рисунок 5.

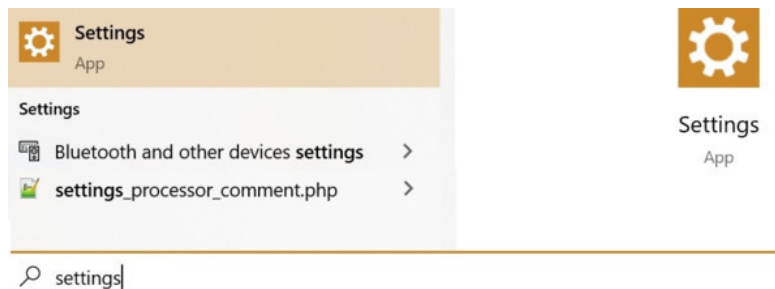


Рис. 5. Найдите значок «Настройки».

В окне «Настройки» введите в поле поиска «Редактировать системные переменные среды». Появится окно свойств системы. См. рисунки 6 и 7.

## Get even more out of Windows

With a few quick selections, you'll be on your way to enjoying the

Let's go!

Skip for now



Edit the system environment variables

edit system



System

Display, sound, notifications,  
power



Devices

Bluetooth

Рисунок 6. Окно настроек

Рис. 7. Окно «Свойства системы».

В окне свойств системы нажмите кнопку «Переменные среды» и откройте окно «Переменные среды». См. Рис. 8.

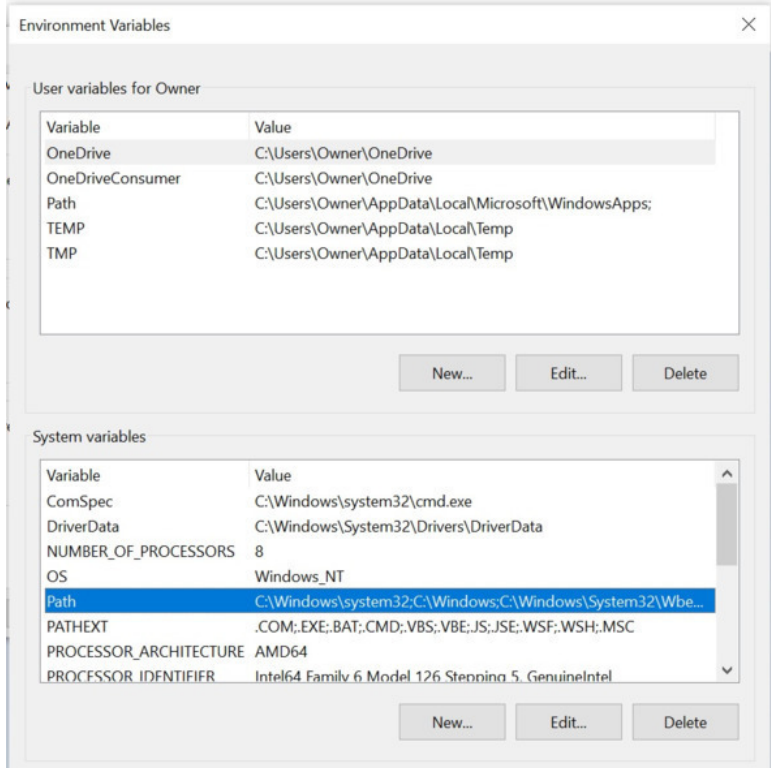


Рисунок 8. Системные переменные среды.

Выберите параметр «Путь» в окне «Переменные среды» и нажмите кнопку «Изменить». Появится окно «Редактировать переменные среды». Нажмите кнопку «Создать» и добавьте свой путь к php (C:\php) в новом поле.

См. Рис. 9.

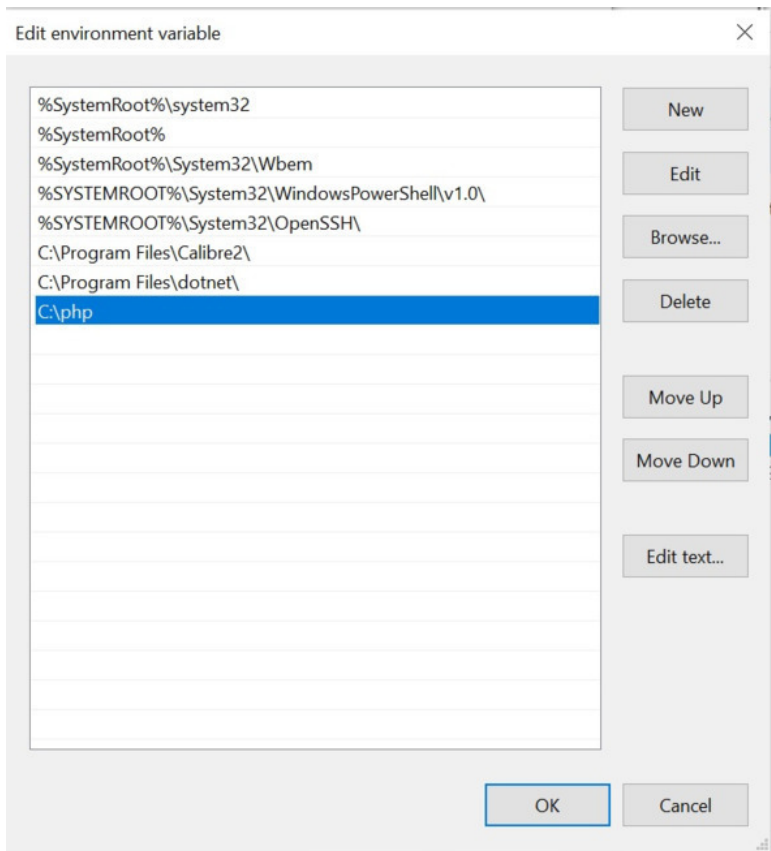


Рисунок 9. Путь к РНР добавлен в системные переменные

Нажмите кнопку «ОК» и сохраните путь РНР.

Теперь создайте папку, в которой вы будете сохранять примеры php-кода, и назовите ее `phpprojects`. Внутри папки `phpprojects` создайте папку `images`. Папка `images` будет использоваться для сохранения красочных изображений, созданных с помощью кода PHP.



# Запуск PHP

В поле поиска в левом нижнем углу экрана введите «CMD». Появится значок окна командной строки. См. рис. 10.

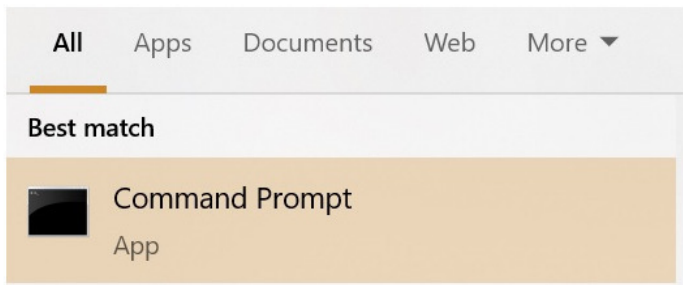


Рис. 10. Значок окна командной строки.

Откройте окно командной строки. Вы увидите следующий текст «C:\Users\Owner>».

Вы должны перейти в папку phpprojects.

Введите «cd..» и нажмите Enter. Вы попадете в папку пользователей.

```
C:\Users\Owner> cd..
```

```
C:\Users>
```

Введите «cd..» и нажмите Enter.

Вы попадете в корень диска C:\.

Введите «cd phpprojects» и нажмите Enter. Вы попадете в папку phpprojects.

Теперь вам нужно ввести команду для запуска PHP.

php -S localhost: 8000

Нажмите Enter.

Command Prompt - php -S localhost:8000

```
Microsoft Windows [Version 10.0.19044.1766]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Owner>cd..

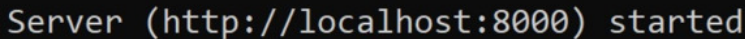
C:\Users>cd..

C:\>cd phpprojects

C:\phpprojects>php -S localhost:8000
[Sun Jul 3 10:01:47 2022] PHP 8.1.5 Development
```

Рис 11. Запуск PHP

PHP запусится. См. рисунки 11 и 12. Рисунок 11. Перейдите в папку phpprojects и запустите PHP.

A black rectangular box with white text that reads "Server (http://localhost:8000) started".

```
Server (http://localhost:8000) started
```

Рисунок 12. PHP-сервер запущен.

Теперь вы можете запустить свой первый PHP-код.

Откройте notepad ++ и напечатайте:

```
<?php  
echo «Computer programming is Fun!»;  
?>
```

Сохраните файл как «index.php».

Откройте браузер, введите URL-адрес: `http://localhost:8000` и нажмите клавишу ввода. Веб-страница с надписью «Компьютерное программирование – это весело!» откроется.

Вы можете создавать более интересные компьютерные программы, но прежде чем делать это, вы должны изучить некоторые базовые вещи.

Как и в математике, в компьютерном программировании используются переменные. В математике переменные – это абстрактные символы, существующие только в нашем вооб-

ражении. Переменные в программировании – это «живые существа», которые живут в памяти компьютера.

Например, если учитель алгебры пишет  $X = 5$ ,  $X$  – абстрактный символ. Если вы напишете  $X = 5$  в

компьютерном коде, вы создадите в памяти компьютера коробочку содержащую 5.

# Тип данных

Данные, которые вы помещаете в поля переменных, могут быть разного типа. Числовой тип данных включает целые числа, называемые `integer`, или числа с десятичной запятой, называемые `float` или `double`. Например, 7 – `integer`, а 7,5 – число с плавающей запятой, `float`.

Строковый тип данных представляет собой последовательность любых буквенных или специальных символов или цифр, заключенных в одинарные или двойные кавычки. Например, номер телефона «7276781209» – это строка, хотя она содержит только цифры.

Тип данных `Boolean` включает `true` или `false`. Вы не включаете его в кавычки. Например,

`$x = true;` или `$y = false;`

Если вы напишете `$x = «true»`, тогда `$x` будет содержать строку, а не логическое значение.

Тип данных `Null` включает в себя `null`, который значит «ничто». Например, `$x = null;`

Другими типами данных являются массив, объект и ресурс. Мы поговорим о них позже.

# Переменные

В PHP имена переменных чувствительны к регистру и должны начинаться со знака доллара и буквенного символа. Переменные. \$date и \$Date – это две разные переменные. Пример переменных: \$x = 5;

или

```
$street_address=«1090 Oak Street»;
```

или

```
$street_address=«1090 Oak Street»;
```

Вы можете конкатенировать (объединить) две или более строки с помощью точки.

Используйте команду echo для вывода значения переменной на веб-страницу или в консоль.

Код PHP должен быть включен в теги '<?php' и»? >».

Строка кода в PHP должна быть закрыта точкой с запятой.

«//» используется для комментария к коду.

Файл: address.php

```
<?php
```

```
$street_address=«1090 Oak Street»;
```

```
$city = «Brooklyn»;
```

```
state=«NY»;
```

```
$zip=«11276»;
```

```
$address=$street.»». $city.»». $state.»». $zip;  
echo $address;  
?>  
Output: 1090 Oak Street, Brooklyn, NY, 11276
```

Давайте объявим две переменные и отобразим их значения на веб-странице.

```
Файл: declare.php  
<?php  
// объявить две переменные  
$a = 7;  
$b = 4;  
// Показать значения переменных  
echo «int a =». $a. "<br>»;  
echo «int b =». $b. "<br> <br>»;  
?>  
Output:  
int a = 7  
int b = 4
```

Мы можем выполнять арифметические операции.  
Давайте создадим PHP-файл «arithmetic.php».

```
<?php  
$a = 7;  
$b = 4;  
// Показать значения переменных  
echo «int a =». $a. "<br>»;
```

```
echo «int b =». $b. "<br> <br>»;  
// арифметические операции  
$sum = $a + $b; //сложение  
$subtraction=$a - $b; //Вычитание  
$product=$a * $b; //умножение  
$division=$b / $a; //деление  
$modulo=$a % $b; // остаток от деления  
// Показать результат  
echo «сумма = a + b <br>»;  
echo» сумма =». $sum. "<br>»;  
echo» вычитание = b - a <br>»;  
echo» вычитание =». $subtraction. "<br>»;  
echo “ <br> умножение = a * b <br>»;  
echo» умножение =». $product.» <br>»;  
echo "<br> деление = b / a <br>»;  
echo» деление =». $division. "<br>»;  
echo «округление =" .round ($division,2). "<br>»;  
echo "<br> остаток от деления = a % b <br>»;  
echo «modulo =». $modulo. "<br> <br>»;  
?>
```

Файл arithmetics.php выведет следующее.

```
int a=4  
int b=7  
сумма = a + b  
сумма = 11  
умножение = a * b  
умножение = 28
```



деление =  $a / b$

деление = 0.57142857142857

округление = 0.57

остаток от деления =  $a \% b$

modulo = 4

Функция `round` возвращает число, с определенным числом десятичных знаков.

В коде вы установили количество знаков после запятой равным 2, и функция вернула 0,57. Если бы вы написали:

```
round($division,4),
```

функция вернула бы 0,5714.

Модуль вычисляет остаток от деления. Например,

$10 \% 3 = 1$ , потому что  $3 * 3 = 9$  и  $10 - 9 = 1$ .

В вашем примере  $7 \% 4 = 3$ , потому что  $4 * 1 = 4$  и  $7 - 4 = 3$ .

Теперь давайте проведем несколько экспериментов. Что произойдет, если мы будем выполнять арифметические операции с переменными, содержащими не числа и строки?

Файл: `string_arithmetics.php`

```
<?php
```

```
$c = «6»;
```

```
$d = «5»;
```

```
echo «str c =». $c. "<br> <br>»;
```

```
echo «str d =». $d.» <br> <br>»;  
echo «sum = c + d; (сумма двух строк) <br> <br>»;  
//сложить две строки как числа  
$sum=$c + $d  
echo «sum=». $sum.» <br> <br>»;  
?>
```

The output is:

```
str c =6  
str d =5  
sum = c + d; (сумма двух строк)  
sum=11
```

PHP обрабатывает строки как числа.  $6 + 5 = 11$ . Теперь давайте попробуем использовать числа как строки. Создайте файл concatenate.php.

```
<?php  
$a = 7;  
$b = 4;  
$c = «6»;  
$d = «5»;  
echo «int a =». $a. " <br> <br>»;  
echo «int b =». $b. " <br> <br>»;  
echo «str c =». $c. " <br> <br>»;  
echo «str d =». $d. " <br> <br>»;  
echo» соединить число и строку <br> <br>»;  
$str1=$a.$d;  
echo «str1=». $str1.» <br>";//output str=45  
echo «соединить два числа как строки <br>»;
```

```
$str2=$a.$b;  
echo «srt2=». $str2; // output str=47  
?>
```

Output:

int a =7

int b =4

str c =6

str d =5

соединить число и строку

srt1=75

соединить два числа как строки

srt2=74

PHP обрабатывает числа как строки и соединяет их как строки. В большинстве случаев PHP изменяет тип переменных по мере необходимости.

# Массивы

Представьте себе контейнер для таблеток с семью отделениями. Он может содержать таблетки или быть пустым.



## Массив

Массив имеет такую же структуру. Количество его отсеков может варьироваться. В держателе для таблеток вы начинаете считать отсеки с единицы.

В массиве подсчет начинается с нуля. Первая секция имеет индекс 0. Вторая секция имеет индекс 1 и так далее.

Вы можете объявить массив следующим образом:

```
$container = array ();
```

В этом случае массив пуст. Вы можете создать не пустой массив.

```
$fruit = array («Apple», «Plum», «Orange», «Banana»,  
Mango», «Lemon», «Pear»);
```

Вы можете получить доступ к каждому элементу массива, используя индекс массива (ключ).

```
echo $fruit [0]; // output Apple  
echo $fruit [1]; // output Orange  
echo $fruit [2]; // output Banana
```

В ассоциативном массиве в качестве ключа используется текстовая строка. Давайте создадим ассоциативный массив цветов HTML. Цвета в HTML кодируются шестнадцатеричными числами. Первые две цифры соответствуют красному цвету, вторые две – зеленому, а третьи две – синему.

Например, #FF0000 – код красного цвета, #00FF00 – код зеленого цвета, а #0000FF – код синего цвета.

```
$colors = array («red» => «#FF0000», «orange»  
=> «#FFA500», «yellow»=> «#FFFF00», green =>  
«#008000 <>»);
```

```
echo $colors [«orange»]; // output #FFA500  
echo $colors [«yellow»]; // output #FFFF00  
echo $colors [«green»]; // output #008000
```

# If-else оператор

Оператор if-else часто используется в компьютерном коде.

Например, если идет дождь, мы остаемся дома. В противном случае мы пойдем на пляж.

В компьютерном коде вы можете написать то же следующим образом:

```
if ($weather == «raining»)  
$plan= «we stay at home»;  
else  
$plan= «we go to the beach»;
```

Оператор if-else может быть более сложным:

```
if ($weather==«Hurricane»)  
$plan= «we go to a shelter»;  
elseif ($weather == «raining»)  
$plan= «we stay at home»;  
elseif ($weather==«Cloudy»)  
$plan= «we go shopping»;  
else  
$plan= «we go to the beach»;
```

Условие может быть более сложным. || символ означает

«или».

```
if (($grade==«A») || ($grade==«B») || ($grade==«C»))  
$test_result = 'passed';  
else  
$test_result = 'failed';
```

Когда используется «или» для нескольких условий, результат будет истинным, если хотя бы одно условие истинно. Символ && означает «и».

```
if (($weather==«sunny») && ($season==«summer»))  
$plan= «we go to the beach»;  
elseif (($weather==«sunny») && ($season==«winter»))  
$plan= «we go skiing»;
```

Когда используется «И» для нескольких условий, результат будет истинным, если истинно каждое условие.

Вы должны включить объединенные условия в скобки ().

Обратите внимание, что вы используете один знак равенства при присвоении значения переменной и двойной знак равенства при сравнении двух переменных.

Для сравнения переменных вы также можете использовать знаки больше ">», меньше "<», больше или равно ">=», меньше или равно "<=" или не равно "!=».

# Циклы

Что такое цикл? Представьте, что ваши дети играют ночью, когда им пора ложиться спать.

Вы говорите им: «Дети, пора спать!».

Они продолжают играть.

Вы говорите им: «Дети, пора спать!».

Они продолжают играть.

И вы продолжаете говорить им одно и то же, пока они не пойдут спать.

В компьютерном программировании это называется циклом.



# While Цикл

```
While ($kids!= «go to sleep»)  
{  
$say= «Let's go to sleep»;  
}
```

Посмотрите на пример цикла while ниже. \$a++ означает увеличение значения \$a на единицу. Это то же самое что

```
$a = $a +1;
```

Файл: while\_loop.php

```
<?p  
$a=0;  
while ($a <5)  
{  
$a++;  
echo 'a =». $a;  
echo»,»,»;  
}  
?>
```

Output: a =1, a =2, a =3, a =4, a =5,

Значение переменной \$a в цикле будет увеличиваться на 1 при каждом выполнении цикла, пока \$a не достигнет 5.

Затем цикл остановится. При использовании цикла `while` вы должны указать условия, при которых цикл останавливается. В противном случае он будет выполняться всегда и приведет к крушению компьютера.

# Do While Цикл

В цикле while сначала проверяется условие. Если условие истинно, код выполняется. В цикле Do While условие проверяется после выполнения кода.

```
do {  
    $say= «Let's go to sleep»;  
} while ($kids!= «go to sleep»);  
Файл: do_while.php  
<?php  
$b=0;  
do {  
    $b++;  
    echo 'b =». $b;  
    echo»,»,»;  
}  
while ($b <5)  
?>  
output: b =1, b =2, b =3, b =4, b =5,
```

Вы можете увидеть разницу между циклами while и do while, если присвоить переменной значение, не удовлетворяющее условию.

```
$a=7;  
while ($a <5)
```

```
{  
$a++;  
echo «a =». $a;  
echo»,»,»;  
}
```

Семь больше пяти. Поскольку условие проверяется перед выполнением кода, цикл `while` не будет выполняться и не будет иметь вывода.

```
$b=7;  
do {  
$b++;  
echo 'b =». $b;  
echo»,»,»;  
}  
while ($b <5)
```

Поскольку условие проверяется после выполнения кода, цикл `do while` выводит `b = 8`.

Семерка будет увеличена на единицу, и выполнение остановится.

# For Цикл

For Цикл имеет следующую структуру.

```
for ($i =0; $i <3; $i++) {  
    echo «I =». $i. "<br>»;  
}
```

The output:

I = 0

I = 1

I = 2

Цикл for удобен для перебора массива. Используя упомянутый ранее массив фруктов, мы можем отобразить все фрукты.

Вы можете использовать функцию count для возврата количества элементов в массиве.

Файл: fruit.php

```
<?php
```

```
$fruit = array («Apple», «Plum», «Orange», «Banana»,  
Mango», «Lemon», «Pear»);  
for ($i=0; $i <count ($fruit); $i++) {  
    echo $fruit [$i];  
    echo»,»;  
}  
?>
```

Output: Apple, Plum, Orange, Banana, Mango, Lemon, Pear,

Вы можете вывести все ключи массива или ключ элемента

по значению.

Файл: return\_key.php

```
<?php
```

```
$fruit = array («Apple», «Plum», «Orange», «Banana»,»  
Mango», «Lemon», «Pear»);
```

```
echo “ <br> <br>»;
```

```
print_r (array_keys ($fruit));
```

```
echo “ <br> <br>»;
```

```
print_r (array_keys ($fruit, «Orange»));
```

```
?>
```

Output:

```
Array ([0] => 0 [1] => 1 [2] => 2 [3] => 3 [4] => 4  
[5] => 5 [6] => 6) //выводит все ключи
```

```
Array ([0] => 2) //выводит ключ только одного элемента
```

Функция print\_r печатает массив.

Файл: colors\_key.php

```
<?php
```

```
$colors=array («red»=> «#FF0000», «orange»=>  
«#FFA500»);
```

```
echo “ <br> <br>»;
```

```
print_r (array_keys ($colors));
```

```
echo “ <br> <br>»;  
print_r (array_keys ($colors, «#FF0000»));  
?>
```

Output:

Array ([0] => red [1] => orange)

Array ([0] => red)

# Цикл Foreach (Для каждого)

Иногда цикл foreach более удобен, потому что вам не нужно знать, сколько элементов в массиве. Вот пример цикла foreach с нашим массивом фруктов.

```
foreach ($fruit as $value) {  
    echo $value;  
    echo»,»,;  
}
```

Output: Apple, Plum, Orange, Banana, Mango, Lemon, Pear,

Первый параметр в цикле foreach должен быть именем массива. В нашем случае это \$fruit. Второй параметр может иметь любое имя. Тот же результат вы получите с помощью следующего кода.

```
foreach ($fruit as $anything) {  
    echo $anything;  
    echo»,»,;  
}
```

```
}
```

Output: Apple, Plum, Orange, Banana, Mango, Lemon, Pear,

```
foreach ($colors as $color) {  
    echo $color;  
    echo»,»,;  
}
```

Output: #FF0000, #FFA500, #FFFF00, #008000,



Вы можете отображать в цикле `foreach` не только значения массива, но и ключи массива.

```
foreach ($colors as $key => $color) {  
    echo $key;  
    echo "=>»;  
    echo $color;  
    echo "<br>»;
```

Output:

```
red=> #FF0000  
orange=> #FFA500  
yellow=> #FFFF00  
green=> #008000
```

Вы должны понимать, что использование точных слов для таких параметров, как `$key` и `$color`, необязательно.

Вы можете использовать любое слово для каждого. Существенным является только местоположение параметров.

```
foreach ($colors as $x => $y) {  
    echo $x;  
    echo "=>»;  
    echo $y;  
    echo "<br>»;
```

Output:

```
red=> #FF0000  
orange=> #FFA500
```

yellow=> #FFFF00

green=> #008000

# Функция

Вы можете представить себе функцию как кофе машину. Вы добавляете туда немного воды и немного молотого кофе, и получаете восхитительный напиток, который поможет вам изучить программирование. То же самое с функцией. Вы передаете что-то функции, и она возвращает что-то вам. PHP имеет встроенные функции и пользовательские функции.

Давайте рассмотрим несколько примеров пользовательских функций – функций, созданных пользователем. Имя функции должно начинаться с буквы или знака подчеркивания. Оно не может начинаться с цифры. Давайте создадим функцию, которая конвертирует фунты в килограммы.

```
function convert_lb_to_kg ($lb) {  
    $kg = $lb * 0.454;  
    return $kg;  
}
```

Функция может иметь несколько параметров

```
function calculate_speed ($miles, $hours) {  
    $msg= «»;  
    If ($hours ==0)  
    $msg= " Количество часов должно быть больше  
    нуля!»;
```

**else**

**{**

**\$speed = \$miles/\$hours;**

**\$msg= «Скорость». \$speed.» миль в час.»;**

**}**

**return \$msg;**

**}**

# Передача переменных по значению или по ссылке

Создадим функцию, которая ничего не возвращает, но меняет свой параметр.

Вы должны использовать знак «&» перед параметром, чтобы передать его по ссылке.

```
<?php  
function convert_lb_to_kg (&$w) {  
    $w = $w * 0.454;  
}
```

```
$a=100;  
convert_lb_to_kg ($a);  
echo “ 100lb =». $a.» kg <br>»;  
?>
```

Output: 100lb = 45.4 kg

Если вы не используете знак &, вы передаете копию своей переменной в функцию, и когда копия изменяется, исходная переменная не изменяется.

```
<?php
```

```
function convert_lb_to_kg ($w) {
```

```
$w = $w * 0.454;
```

```
}
```

```
$a=100;
```

```
convert_lb_to_kg ($a);
```

```
echo " 100lb =». $a.» kg <br>»;
```

```
?>
```

Output: 100lb = 100 kg

# Как рисовать изображения

Теперь мы изучили основы и можем приступить к рисованию изображений.

Вернемся к школьной математике. Когда вы рисовали графики, горизонтальная ось ( $x$ ) называлась абсциссой, а вертикальная ось ( $y$ ) называлась ординатой. Чтобы нарисовать линию, мы должны указать координаты  $X$  и  $Y$  двух точек. В компьютерном программировании ширина и высота окна измеряются в пикселях. Точка в левом верхнем углу окна имеет координаты  $X=0$  и  $Y=0$ .

Если мы сдвинем точку на 200 пикселей вправо, не меняя ее положения по вертикали, координаты станут  $X=200$  и  $Y=0$ .

Если мы переместим точку на 300 пикселей вниз, координаты точки станут  $X = 200$   $Y = 300$ .

Нарисуйте еще одну точку с координатами  $X = 800$  и  $Y = 600$  и соедините точки прямой линией. См. Рис. 13.

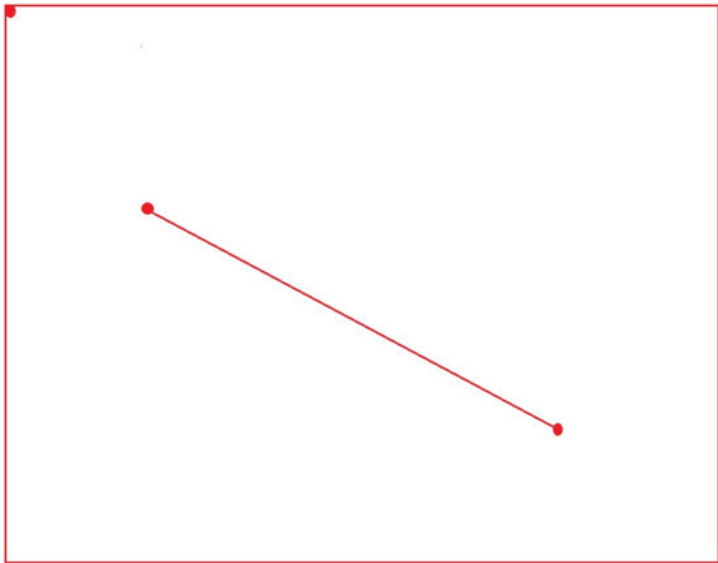


Рис. 13. Прямая.

Теперь вы можете нарисовать линию на веб-странице с помощью PHP-кода.

Но прежде мы должны внести изменения в файл `php.ini` в парке PHP. Откройте файл `php.ini` в `notepad++` и найдите строку «`;extension=gd`». Удалите точку с запятой перед `extension=gd`, `extension=fileinfo` и `extension=gmp`.

Сохраните файл.

```
;extension=bz2
```

```
;extension=curl
```



```
;extension=ffi
;extension=ftp
extension=fileinfo
extension=gd
;extension=gettext
extension=gmp
extension=intl
```

Найдите error reporting, удалите точку с запятой перед E\_ALL и удалите весь текст после E\_ALL на той же строке.

```
;
; Error Level Constants:
E_ALL
; E_ERROR           - fatal run-time errors
; E_RECOVERABLE_ERROR - almost fatal run-time errors
; E_WARNING         - run-time warnings (non-fatal errors)
; E_PARSE           - compile-time parse errors
; E_NOTICE          - run-time notices (these are warnings which often result in a notice on a subsequent page)
```

Рис. 14. Error reporting

Вы можете посмотреть это видео, чтобы узнать больше о конфигурации PHP.

<https://youtu.be/LVEhccXXnOo>

# Как рисовать линии

Чтобы создать и отобразить линию или любую фигуру, нам нужно создать пустое черное изображение, используя встроенную функцию PHP **imagecreatetruecolor**. Эта функция принимает два параметра: ширину и высоту изображения.

Затем нам нужно заказать цвет с помощью функции **imagecolorallocate**. Эта функция принимает изображение, созданное функцией **imagecreatetruecolor**, и цвет в формате RGB. RGB означает красный (Red), зеленый (Green) и синий (Blue).

Каждый RGB цвет может иметь значение от 0 до 255. Например, RGB (255, 0, 0) красный,

RGB (255, 255, 0) – желтый, RGB (0, 255, 0) – зеленый, RGB (0, 0, 255) – синий.

Вы можете выбрать значения RGB цветов на этом веб-сайте: [https://www.w3schools.com/colors/colors\\_converter.asp](https://www.w3schools.com/colors/colors_converter.asp).

После заказа цвета мы можем нарисовать изображение. Чтобы нарисовать линию, мы должны использовать функцию **imageline**, которая принимает шесть параметров: изображение, созданное функцией **imagecreatetruecolor**, координаты X и Y двух точек и цвет, созданный функцией **imagecolorallocate**.

Чтобы отобразить строку на веб-странице, мы должны ис-

ПОЛЬЗОВАТЬ ЗАГОЛОВОК:

**header («Content-type: image/jpeg»);**

**imagejpeg (\$img);** функция выводит изображение в браузер.

Функция **imagedestroy** стирает изображение и освобождает память.

Файл: **draw\_line.php**. (См. рис. 15.)

```
<?php
```

```
// Создать пустое изображение
```

```
$img=imagecreatetruecolor (900, 700);
```

```
//заказать цвет (заказываем цвет)
```

```
$yellow = imagecolorallocate ($img, 255, 255, 0);
```

```
// draw a line
```

```
imageline (
```

```
$img, //GdImage
```

```
200, //x1,
```

```
300, //y1,
```

```
800, //x2,
```

```
600, //y2,
```

```
$yellow //color
```

```
);
```

```
header («Content-type: image/jpeg»);
```

```
//показать изображение
```

```
imagejpeg ($img);
```

```
imagedestroy ($img);
```

```
?>
```



Рисунок 15. Нарисуйте линию. Draw line.

Мы можем изменить толщину линии, добавив одну строку кода. Функция `imagesthickness` будет контролировать толщину линии в пикселях.

Файл: `draw_thick_line.php` (см. рис. 16).

```
<?php
```

```
//Создать пустое изображение
```

```
$img=imagecreatetruecolor (900, 700);
```

```
// set the line thickness
```

```
imagesthickness ($img, 7);
```

```
//заказать цвет (заказываем цвет)
```

```
$yellow = imagecolorallocate ($img, 255, 255, 0);  
// начертить линию  
imageline (  
$img, //GdImage  
200, //x1,  
300, //y1,  
800, //x2,  
600, //y2,  
$yellow //color  
  
);  
header («Content-type: image/jpeg»);  
//вывести изображение  
imagejpeg ($img);  
imagedestroy ($img);  
?>
```

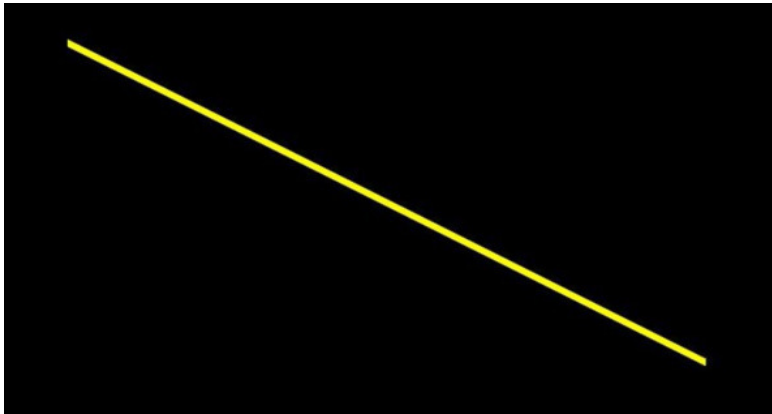


Рис. 16. Толстая линия.

Чтобы попрактиковаться, пожалуйста, создайте линии с разными направлениями, цветами и толщиной. Попробуйте использовать функцию **imagedashedline** вместо функции **imagine**. **imagedashedline** принимает те же параметры, но рисует прерванную линию. ( — — — — — — — — — )

Нарисуем солнце на голубом небе. Сначала нарисуем желтый круг. См. рис. 16. Чтобы нарисовать круг, мы будем использовать функцию **imagefilledellipse**. Эта функция принимает шесть параметров, GD-изображение, координаты X и Y центра эллипса, ширину, высоту и цвет. См. исходный код в файле `yellow_circle1.php`.

File: `yellow_circle1.php`

`<?php`

```
$height=«2000»;  
$width=«2000»;  
// Создать пустое изображение  
$img=imagecreatetruecolor ($width, $height);  
//заказать цвет (заказываем цвет)  
$yellow = imagecolorallocate ($img, 255, 255, 0);  
imagefilledellipse ($img, $width/2, $height/2, $width/2,  
$height/2, $yellow);  
//вывести изображение в файл  
imagejpeg($img,'images/yellow_circle1.jpg',100);  
header («Content-type: image/jpeg»);  
//показать изображение  
imagejpeg ($img);  
imagedestroy ($img);  
?>
```

Теперь давайте добавим синий цвет фона. Закажите голубой цвет:

```
$lightblue = imagecolorallocate ($img, 0, 195, 255);
```

Затем нарисуйте синий прямоугольник, используя функцию `imagefilledrectangle`. Эта функция принимает шесть параметров: изображение GD, координаты X1 и Y1 начальной точки (левый верхний угол) и координаты X2 и Y2 конечной точки (правый нижний угол) и цвет. Вот строки кода, которые мы добавили в файл `yellow_circle2.php`. Они выделены.

**imagefilledrectangle (\$img, 0, 0, \$width, \$height, \$lightblue);**

Затем мы можем нарисовать желтый круг с помощью функции `imagefilledellipse` и получить изображение, как на рисунке 17.

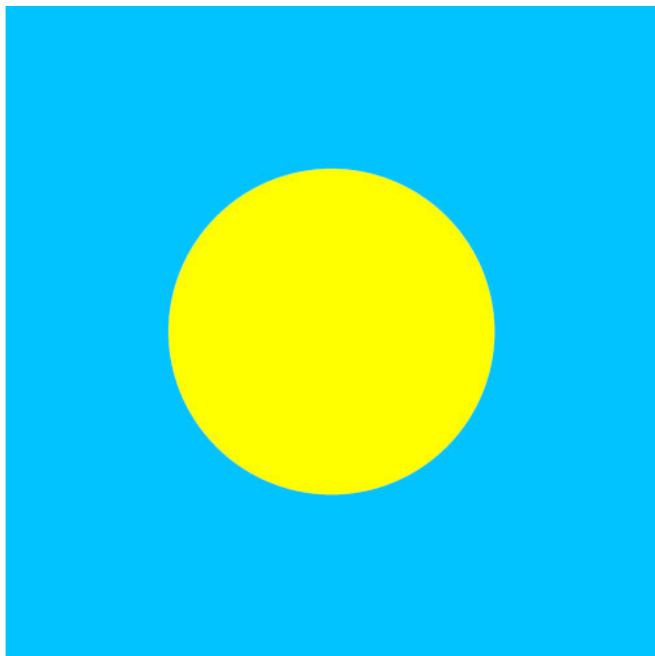


Рисунок 17 Желтый круг на голубом фоне.

Файл: `yellow_circle2.php`



```
<?php
$height=«2000»;
$width=«2000»;
//Создать пустое изображение
$img=imagecreatetruecolor ($width, $height);
//заказать цвет
$lightblue = imagecolorallocate ($img, 0, 195, 255);
$yellow = imagecolorallocate ($img, 255, 255, 0);
//рисуем прямоугольник
imagefilledrectangle ($img, 0, 0, $width, $height,
$lightblue);
imagefilledellipse ($img, $width/2, $height/2, $width/2,
$height/2, $yellow);
//output the image in the file
imagejpeg($img,'images/yellow_circle2.jpg',100);
header («Content-type: image/jpeg»);
//показать изображение
imagejpeg ($img);
imagedestroy ($img);
?>
```

Теперь нарисуем солнце, как на рисунке 18.

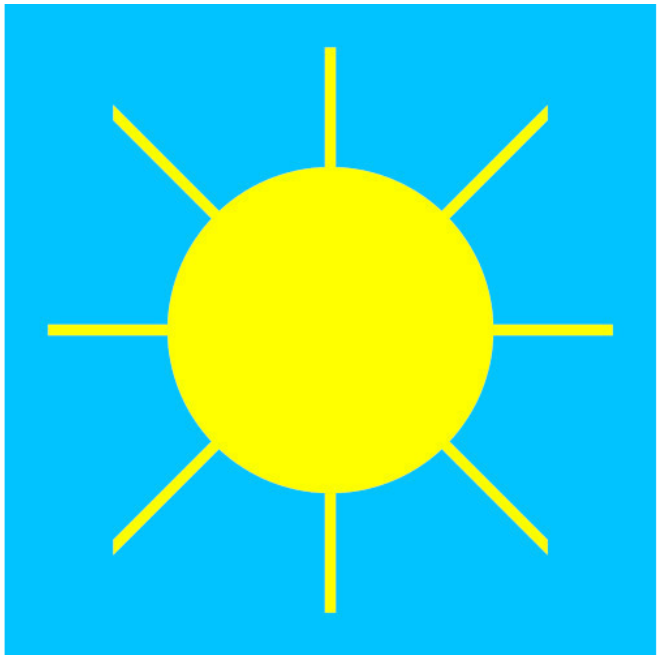


Рисунок 18. Солнце.

Файл: sun3.php

```
<?php  
$height=«3000»;  
$width=«3000»;  
//Создать пустое изображение  
$img=imagecreatetruecolor ($width, $height);
```

**//заказать цвет**

**\$lightblue = imagecolorallocate (\$img, 0, 195, 255);**

**\$yellow = imagecolorallocate (\$img, 255, 255, 0);**

# Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.