

# Работа с датами в PHP

Для работы с датами в PHP применяются различные функции. Мы начнем изучение с функции `time`.

## Функция `time`, формат `timestamp`

Функция `time` возвращает разницу в секундах между **1-го января 1970 года и текущим моментом времени**. Такое представление даты называется форматом `timestamp`.

## Зачем нужен `timestamp`?

Время в формате `timestamp` используется для того, чтобы найти разницу между датами в секундах.

К примеру, у меня есть `timestamp` за **1-го января текущего года** и `timestamp` за **текущий момент времени**. Я отниму один от другого и получу разницу между этими моментами **в секундах**. Поделю эту разницу на 60 и получу разницу в минутах (так как в минуте 60 секунд). Поделю еще на 60 - и получу разницу в часах. И так далее.

С помощью функции `time` мы можем получить только текущий момент времени. Чтобы получить `timestamp` за **любую дату** следует использовать функцию `mktime`:

## Функция `mktime`

Функция `mktime` работает аналогично функции `time`, но, в отличие от нее, принимает параметры: `mktime(час, минута, секунда, месяц, день, год)` ( обратите внимание на то, что **месяц** и **день** переставлены местами). Посмотрите примеры работы:

```
<?php
/*
Функция вернет timestamp для 31 января 2020 года,
12 часов, 43 минуты, 59 секунд:
*/
echo mktime(12, 43, 59, 1, 31, 2020);
?>
```

**Параметры можно опускать с конца.** Если мы не напишем год (последний параметр) - то автоматически возьмется текущий год, если не напишем еще и день (предпоследний параметр) - то возьмется текущий день и текущий год. И так далее:

```
<?php
/*
Функция вернет timestamp для 31 января текущего года,
12 часов, 43 минуты, 59 секунд:
*/
echo mktime(12, 43, 59, 1, 31);
?>
```

```

<?php
/*
Функция вернет timestamp для 14 (это номер текущего дня)
января текущего года, 12 часов, 43 минуты, 59 секунд:
*/
echo mktime(12, 43, 59, 1);

?>
<?php
/*
Функция вернет timestamp текущего дня, текущего месяца, текущего
года,
12 часов, 43 минуты, 59 секунд:
*/
echo mktime(12, 43, 59);
?>

```

Решим типичную **задачу на time и mktime**. Пусть нам нужно найти **разницу в секундах** между текущим моментом времени и полуднем (12 часов, 0 минут, 0 секунд) 1-го февраля (февраль - это месяц номер 2) 2000 года:

```

<?php
/*
Функция time() вернет текущий момент времени в формате timestamp,
а mktime - за заданную дату.

Отнимем результаты друг от друга и получим разницу в секундах:
*/
echo time() - mktime(12, 0, 0, 2, 1, 2000);
?>

```

**Полученная разница в секундах** будет выглядеть так: **632269111** (обновите страницу и это число поменяется).

Вооружившись знаниями о том, что такое формат **timestamp** (он нам еще понадобится в дальнейшем), изучим более полезные функции для работы с датами, например, функцию **date**.

## Функция date

Функция **date** выводит текущие дату и время в заданном формате.

**Формат** задается управляющими командами (английскими буквами), при этом можно вставлять любые разделители между ними (дефисы, двоеточие и так далее).

**Команды** (заглавные буквы отличаются от обычных, обратите внимание!):

- **U** – количество секунд, прошедших с 1 января 1970 года (то есть **timestamp**).

- **z** – номер дня от начала года.
- **Y** – год, 4 цифры.
- **y** - год, две цифры.
- **m** – номер месяца (с нулем спереди).
- **n** – номер месяца без нуля впереди.
- **d** – номер дня в месяце, всегда две цифры (то есть первая может быть нулём).
- **j** – номер дня в месяце без предваряющего нуля.
- **w** – день недели (0 - воскресенье, 1 - понедельник и т.д.).
- **h** – часы в 12-часовом формате.
- **H** – часы в 24-часовом формате.
- **i** – минуты.
- **s** – секунды.
- **L** – 1, если високосный год, 0, если не високосный.
- **W** – порядковый номер недели года.
- **t** – количество дней в указанном месяце.

### **Примеры работы с date:**

```
<?php
```

```
//Все примеры показаны для даты 01.06.2013 в 12.23.59, понедельник
```

```
echo date('Y'); //вернет '2013'
```

```
echo date('y'); //вернет '13'
```

```
echo date('m'); //вернет '06' - номер месяца
```

```
echo date('d'); //вернет '01' - номер дня в месяце
```

```
echo date('j'); //вернет '1' - номер дня в месяце (без нуля спереди)
```

```
echo date('w'); //вернет '1' - понедельник
```

```
echo date('H'); //вернет '12' - часы
```

```
echo date('i'); //вернет '23' - минуты
```

```
echo date('s'); //вернет '59' - секунды
```

```
echo date('d-m-Y'); //вернет '01-06-2013'

echo date('d.m.Y'); //вернет '01.06.2013'

echo date('H:i:s d.m.Y'); //вернет '12:23:59 01.06.2013'

?>
```

## Второй параметр функции date

Функция **date** имеет второй **необязательный параметр**, который принимает момент времени в формате timestamp. Если передать этот параметр, то функция **date** отформатирует не текущий момент времени, а тот, который передан вторым параметром. Этот **timestamp** можно получить, к примеру, через **mkttime** (но не обязательно):

```
<?php

echo date('d-m-Y', mkttime(0, 0, 0, 12, 29, 13)); //вернет '29-12-2013'

?>
```

Этим можно воспользоваться, чтобы узнать номер дня недели за определенную дату - просто передадим ее вторым параметром с помощью функции **mkttime**, а первым параметром поставим управляющий символ '**w**':

```
<?php

//Узнаем какой день недели был 29-12-2013:

echo date('w', mkttime(0, 0, 0, 12, 29, 13)); //вернет '0' - воскресенье

?>
```

## Функция strtotime

Следующая полезная функция, которую мы разберем, называется **strtotime**.

Функция **strtotime** - это аналог функции **mkttime** (тоже возвращает timestamp), только в отличие от нее принимает дату в **более** свободном формате.

К примеру, я могу передать ей строку '**2025-12-31**' и функция сама разберет, где тут год, где месяц, а где день, и вернет эту дату в формате timestamp.

Что можно делать еще: можно написать так - **strtotime('now')** - и мы получим текущий момент времени, или так - **strtotime('next Monday')** - и мы получим следующий понедельник (Monday по-английски 'понедельник').

Все форматы можно посмотреть [тут](#).

Примеры работы:

```
<?php

echo strtotime("now");

echo strtotime("10 September 2000");
```

```
echo strtotime("+1 day");

echo strtotime("+1 week");

echo strtotime("+1 week 2 days 4 hours 2 seconds");

echo strtotime("next Thursday");

echo strtotime("last Monday");

?>
```

Следующий код вернет дату предыдущего понедельника:

```
<?php

echo date('d-m-Y', strtotime("last Monday"));

?>
```

## Как добавить или отнять дату

Чтобы отнять или прибавить к дате определенный промежуток времени, следует пользоваться комбинацией из трех функций: [date\\_create](#) - она подготавливает дату к работе (дата должна быть в формате год-месяц-день) создавая так называемый *объект дата*, с которым производятся дальнейшие манипуляции, [date\\_modify](#) - она прибавляет или отнимает, [date\\_format](#) - она выводит получившуюся дату в заданном формате. Чтобы вникнуть в работу с этими функциями изучите примеры ниже.

### Пример 1

Давайте создадим объект с датой за 2025 год, 12 месяц, 31 день, затем прибавим к ней 1 день и выведем в формате 'день.месяц.год'

```
<?php

$date = date_create('2025-12-31');

date_modify($date, '1 day');

echo date_format($date, 'd.m.Y');

?>
```

Результат выполнения кода:

01.01.2026

### Пример 2

Давайте создадим объект с датой за 2025 год, 12 месяц, 31 день, затем прибавим к ней 3 дня и выведем в формате 'день.месяц.год'

```
<?php

$date = date_create('2025-12-31');

date_modify($date, '3 days');
```

```
echo date_format($date, 'd.m.Y');  
?>
```

Результат выполнения кода:

03.01.2026

## Пример 3

Давайте создадим объект с датой за 2025 год, 12 месяц, 31 день, затем прибавим к ней 3 дня и 1 месяц и выведем в формате 'день.месяц.год'

```
<?php  
    $date = date_create('2025-12-31');  
    date_modify($date, '3 days 1 month');  
    echo date_format($date, 'd.m.Y');  
?>
```

Результат выполнения кода:

03.02.2026

## Пример 4

Давайте создадим объект с датой за 2025 год, 1 месяц, 1 день, затем отнимем от нее 1 день и выведем в формате 'день.месяц.год'

```
<?php  
    $date = date_create('2025-01-01');  
    date_modify($date, '-1 day');  
    echo date_format($date, 'd.m.Y');  
?>
```

Результат выполнения кода:

31.12.2024

# Задачи на даты в PHP

## Примеры решения задач

### Задача

**Задача.** Выведите 23 сентября 2031 года, 12:58:59 в формате timestamp.

**Решение:** воспользуемся [функцией mktime](#) (сентябрь - 9-ый месяц):

```

<?php

echo mktime(12, 58, 59, 9, 23, 2031);

/*
Напоминаю, что месяц и день идут в неправильном порядке:
поэтому '9, 23,' а не '23, 9'.

*/
?>

```

Можно также воспользоваться [функцией strtotime](#), если представить нужную дату в формате **2031-09-23 12:58:59**:

```

<?php

echo strtotime('2031-09-23 12:58:59');

?>

```

## Задача

**Задача.** Найдите разницу между **1 сентября 2010 года, 7:25:59** и текущим моментом времени в секундах.

**Решение:** текущий момент времени в формате **timestamp** получим с помощью функции [time](#), а **timestamp** для заданной даты - с помощью функции [mktime](#). Отнимем одно число от второго и получим искомую разницу:

```

<?php

echo time() - mktime(7, 25, 59, 9, 1, 2010);

?>

```

## Задача . Функция date

**Задача.** Выведите текущую дату-время в формате '**2025.12.31 12:59:59**'.

**Решение:** воспользуемся функцией [date](#), передав ей управляющие команды в таком порядке: год (команда **Y**), потом точку как символ, потом месяц (команда **m**), опять точку, день (команда **d**), час (команда **H**), двоеточие, минуту (команда **i**), секунду (команда **s**). Получится такая строка: '**Y.m.d H:i:s**'. Подставим ее в функцию **date**:

```

<?php

echo date('Y.m.d H:i:s');

?>

```

## Задача . Функция date

**Задача.** Выведите 1-го сентября текущего года в формате '**2020.09.01**'.

**Решение:** для начала с помощью функции [mktime](#) преобразуем 1-го сентября текущего года в формат **timestamp**. Мы это делаем для того, чтобы подставить найденное число вторым параметром в функцию [date](#) (а первым параметром для **date** мы укажем формат вывода).

Так как требуется текущий год, то последний параметр (год) для **mktime** мы не указываем - тогда возьмется текущий год:

```
<?php  
    //Выведем timestamp 1-го сентября текущего года:  
    echo mktime(0, 0, 0, 9, 1);  
?>
```

Ну, а теперь подставим найденный **timestamp** в функцию **date**:

```
<?php  
    echo date('Y.m.d', mktime(0, 0, 0, 9, 1));  
?>
```

## Задача . Функция date. Вывод дня недели словом

**Задача.** Узнайте, какой день недели (словом) был 1 сентября 2010 года.

**Решение:** решение аналогично предыдущей задаче, только формат вывода для функции **date** мы сделаем в таком виде: '**w**'. В этом случае **date** вернет нам число, соответствующее дню недели за заданную дату (0 - воскресенье, 1 - понедельник и так далее):

```
<?php  
    //День недели числом за нужную дату:  
    echo date('w', mktime(0, 0, 0, 9, 1, 2010));  
?>
```

Кстати, если бы мы хотели узнать, какой день недели сегодня - мы бы просто не передавали второй параметр функции **date** (тогда бы взялся текущий момент времени и, соответственно, вывелась бы '**w**' за текущий день).

Продолжим решать нашу задачу: мы вывели номер дня недели, а по задаче его следует вывести словом. Для этого составим массив дней недели **\$week** и с его помощью выведем то, что нам нужно. Вот этот массив:

```
<?php  
    //Массив дней недели:  
    $week = ['вс', 'пн', 'вт', 'ср', 'чт', 'пт', 'сб'];  
  
    //Выведем с его помощью, к примеру, понедельник:  
    echo $week[1];  
  
    //А теперь вторник:  
    echo $week[2];  
?>
```

Совместим теперь то, что вернет нам **date**, с нашим массивом **\$week**:

```
<?php  
    //День недели цифрой за нужную дату:
```

```

$day = date('w', mktime(0, 0, 0, 9, 1, 2010));

//Массив дней недели:
$week = ['вс', 'пн', 'вт', 'ср', 'чт', 'пт', 'сб'];

//День недели словом:
echo $week[$day];

?>

```

## Задача . Преобразование форматов

**Задача.** Данна дата в формате '31-12-2025'. С помощью функций [mktime](#) и [explode](#) переведите эту дату в формат **timestamp**.

**Решение:** разобьем строку '31-12-2025' функцией **explode** в массив **\$arr**:

```

<?php

$arr = explode('31-12-2025'); //получим ['31', '12', '2025']

?>

```

В элементе **\$arr[0]** будет лежать день, в элементе **\$arr[1]** - месяц, в элементе **\$arr[2]** - год. Подставим эти данные в функцию **mktime** (напоминаю, что она принимает данные в формате '...месяц-день-год', не '...день-месяц-год')

```

<?php

$arr = explode('31-12-2025');

echo mktime(0, 0, 0, $arr[1], $arr[0], $arr[2]);

?>

```

## Задачи для решения

### Timestamp: time и mktime

Для решения задач данного блока вам понадобятся следующие функции: [time](#), [mktime](#).

1. Выведите текущее время в формате **timestamp**.
2. Выведите 1 марта 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 времени.
5. Найдите количество целых часов, прошедших с **7:23:48 текущего дня** до настоящего момента времени.

## Функция date

Для решения задач данного блока вам понадобятся следующие функции: [date](#).

6. Выведите на экран текущий год, месяц, день, час, минуту, секунду.
7. Выведите текущую дату-время в форматах '**2025-12-31', '31.12.2025', '31.12.13', '12:59:59'**'.
8. С помощью функций [mktime](#) и [date](#) выведите 12 февраля 2025 года в формате '**12.02.2025**'.
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 количество дней в текущем месяце. Скрипт должен работать независимо от месяца, в котором он запущен.
12. Сделайте поле ввода, в которое пользователь вводит год (4 цифры), а скрипт определяет **високосный** ли год.
13. Сделайте форму, которая спрашивает дату в формате '**31.12.2025**'. С помощью функций [mktime](#) и [explode](#) переведите эту дату в формат **timestamp**. Узнайте день недели (словом) за введенную дату.
14. Сделайте форму, которая спрашивает дату в формате '**2025-12-31**'. С помощью функций [mktime](#) и [explode](#) переведите эту дату в формат **timestamp**. Узнайте месяц (словом) за введенную дату.

## Сравнение дат

15. Сделайте форму, которая спрашивает две даты в формате '**2025-12-31**'. Первую дату запишите в переменную **\$date1**, а вторую в **\$date2**. Сравните, какая из введенных дат больше. Выведите ее на экран.

## На strtotime

Для решения задач данного блока вам понадобятся следующие функции: [strtotime](#).

16. Данна дата в формате '**2025-12-31**'. С помощью функции [strtotime](#) и функции [date](#) преобразуйте ее в формат '**31-12-2025**'.
17. Сделайте форму, которая спрашивает дату-время в формате '**2025-12-31T12:13:59**'. С помощью функции [strtotime](#) и функции [date](#) преобразуйте ее в формат '**12:13:59 31.12.2025**'.

## Прибавление и отнимание дат

Для решения задач данного блока вам понадобятся следующие функции: [date\\_create](#), [date\\_modify](#), [date\\_format](#).

18. В переменной **\$date** лежит дата в формате '**2025-12-31**'. Прибавьте к этой дате 2 дня, 1 месяц и 3 дня, 1 год. Отнимите от этой даты 3 дня.

## **Задачи**

19. Узнайте сколько дней осталось до Нового Года. Скрипт должен работать в любом году.
20. Сделайте форму с одним полем ввода, в которое пользователь вводит год. Найдите все пятницы 13-е в этом году. Результат выведите в виде массива дат.
21. Узнайте какой день недели был 100 дней назад.