

## **Создание таймера с помощью html, css и js.**

Таймеры широко применяются на сайтах разного назначения и решают разнообразные задачи.

### **Промоакции и скидки.**

Многие онлайн-магазины используют таймеры обратного отсчета, чтобы стимулировать покупателей быстрее принять решение о покупке.

### **Проведение мероприятий и событий.**

Онлайн-трансляции, конференции, мастер-классы часто анонсируются заранее, а таймер помогает посетителям сайта отслеживать время начала события.

### **Обратный отсчёт до доставки товара.**

Некоторые сервисы доставки продуктов питания или товаров показывают клиенту точное время, когда заказ прибудет.

Реализуем js-код для отображения простого таймера обратного отсчета, который показывает оставшиеся дни, часы, минуты и секунды до определенной даты.

## **До окончания акции осталось :**

0 дней 9 часов 41 минута 19 секунд

В html-разметке создадим контейнер с 4 абзацами с идентификаторами days, hours, minutes и seconds.

```
<p><span id="days"></span></p>
```

В js-коде для всего документа добавим обработчик событий с параметром DOMContentLoaded. – событие в JavaScript, которое срабатывает, когда html-документ полностью загружен и разобран, то есть DOM-дерево построено, но внешние ресурсы, такие как изображения, стили и фреймы, могут быть ещё не загружены.

```
document.addEventListener("DOMContentLoaded", () => {})
```

Определим переменную для отображения конечной даты.

```
let endDate = new Date("January 8, 2026 00:00:00").getTime();
```

Обновляем таймер каждую секунду.

```
setInterval(() => {  
}, 1000);
```

Определяем разницу между текущей датой и концом периода, вычисляя количество оставшихся миллисекунд.

```
let now = new Date().getTime();  
let timeLeft = endDate - now
```

Преобразуем разницу во времени в число дней, часов, минут и секунд:

Дни – деление общего количества миллисекунд на количество миллисекунд в одном дне ( $1000 \text{ мс} \times 60 \text{ сек} \times 60 \text{ мин} \times 24 \text{ ч}$ ).

Часы – остаток от деления на количество миллисекунд в сутках. Затем делится на количество миллисекунд в часе.

Минуты – остаток от деления на количество миллисекунд в часах. Затем делится на количество миллисекунд в минуте.

Секунды – остаток от деления на количество миллисекунд в минутах.

Затем делится на количество миллисекунд в секунде.

```
let days = Math.floor(timeLeft / (1000*60*60*24));  
let hours = Math.floor((timeLeft % (1000*60*60*24)) / (1000*60*60));  
let minutes = Math.floor((timeLeft % (1000*60*60)) / (1000*60));  
let seconds = Math.floor((timeLeft % (1000*60)) / 1000);
```

Если разница во времени меньше либо равна 0, то выведем сообщение – время вышло.

```
if (timeLeft <= 0) {  
    return document.querySelector(".timer_countdown").textContent = "Время вышло";  
}
```

Функции formatDay(), formatHours(), formatMinutes() и formatSeconds() автоматически определяют правильное склонение для каждой единицы времени («день/дня/дней», «час/часа/часов») исходя из числа дней, часов, минут и секунд.

Если дней ровно ноль или если количество дней попадает в диапазон от 5 до 20 включительно возвращается слово «дней».

Если последняя цифра числа равна 1 (например, 1, 11, 21, 31 и т.д., кроме случаев, когда число заканчивается на 11), возвращаем слово «день».

Если число дней оканчивается на цифру 2, 3 или 4 (например, 2, 3, 4, 22, 23, 24...), возвращает слово «дня».

Во всех остальных ситуациях возвращается слово.

```
function formatDay() {  
    if (days === 0 || (days >= 5 && days <= 20 )) {  
        return "дней";  
    } else if ((days % 10) === 1) {  
        return "день";  
    } else if ([2, 3, 4].includes(days % 10)) {  
        return "дня";  
    } else {  
        return "дней";  
    }  
}
```

Время выводится в соответствующих местах документа.

```
document.getElementById("days").textContent = (days + " " + formatDay());
```