# Cookies de JavaScript

[❮ Anterior](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/js_timing.asp&usg=ALkJrhibNgR1dP2JI8N7DnwYhm6gyOfKWg)[Próximo ❯](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/js_ajax_intro.asp&usg=ALkJrhhhONwjEXkRsLhgti5VXf1M9PoOZA)

Os cookies permitem que você armazene informações do usuário em páginas da web.

## O que são Cookies?

Os cookies são dados, armazenados em pequenos arquivos de texto, no seu computador.

Quando um servidor web enviou uma página da Web para um navegador, a conexão é desligada e o servidor esquece tudo sobre o usuário.

Cookies foram inventados para resolver o problema "como lembrar informações sobre o usuário":

* Quando um usuário visita uma página da Web, seu nome pode ser armazenado em um cookie.
* Da próxima vez que o usuário visitar a página, o cookie "lembra" seu nome.

Os cookies são salvos em pares nome-valor como:

username = John Doe

Quando um navegador solicita uma página da Web de um servidor, os cookies pertencentes à página são adicionados ao pedido. Desta forma, o servidor obtém os dados necessários para "lembrar" informações sobre usuários.

Nenhum dos exemplos abaixo funcionará se o navegador tiver suporte para cookies local desativado.

## Crie um cookie com JavaScript

O JavaScript pode criar, ler e excluir cookies com a propriedade **document.cookie** .

Com JavaScript, um cookie pode ser criado assim:

document.cookie = "username=John Doe";

Você também pode adicionar uma data de validade (em horário UTC). Por padrão, o cookie é excluído quando o navegador está fechado:

document.cookie = "username=John Doe; expires=Thu, 18 Dec 2013 12:00:00 UTC";

Com um parâmetro de caminho, você pode informar ao navegador em que caminho o cookie pertence. Por padrão, o cookie pertence à página atual.

document.cookie = "username=John Doe; expires=Thu, 18 Dec 2013 12:00:00 UTC; path=/";

## Leia um Cookie com JavaScript

Com o JavaScript, os cookies podem ser lidos assim:

var x = document.cookie;

document.cookie retornará todos os cookies em uma string como: cookie1 = value; cookie2 = valor; cookie3 = valor;

## Alterar um cookie com JavaScript

Com o JavaScript, você pode alterar um cookie da mesma forma que você o criou:

document.cookie = "username=John Smith; expires=Thu, 18 Dec 2013 12:00:00 UTC; path=/";

O velho cookie é substituído.

## Exclua um cookie com JavaScript

Eliminar um cookie é muito simples.

Você não precisa especificar um valor de cookie quando você exclui um cookie.

Basta configurar o parâmetro expira para uma data passada:

document.cookie = "username=; expires=Thu, 01 Jan 1970 00:00:00 UTC; path=/;";

Você deve definir o caminho do cookie para garantir que você exclua o cookie certo.

Alguns navegadores não permitem que você exclua um cookie se você não especificar o caminho.

## The Cookie String

A propriedade document.cookie parece uma string de texto normal. Mas não é.

Mesmo se você escrever uma corda de cookie inteira para document.cookie, quando você lê isso novamente, você só pode ver o par nome-valor dele.

Se você definir um novo cookie, os cookies mais antigos não são substituídos. O novo cookie é adicionado ao document.cookie, então se você ler document.cookie novamente, você terá algo como:

cookie1 = valor; cookie2 = valor;

Exibir Todos os Cookies Criar Cookie 1 Criar Cookie 2 Eliminar Cookie 1 Eliminar Cookie 2

Se você quiser encontrar o valor de um cookie especificado, você deve escrever uma função JavaScript que pesquisa o valor do cookie na cadeia de cookies.

## Exemplo de Cookie do JavaScript

No exemplo a seguir, criaremos um cookie que armazena o nome de um visitante.

A primeira vez que um visitante chega à página web, ele será convidado a preencher seu nome. O nome é então armazenado em um cookie.

Na próxima vez que o visitante chegar na mesma página, ele receberá uma mensagem de boas-vindas.

Para o exemplo, criaremos 3 funções de JavaScript:

1. Uma função para definir um valor de cookie
2. Uma função para obter um valor de cookie
3. Uma função para verificar um valor de cookie

## Uma função para definir um cookie

Primeiro, criamos uma função que armazena o nome do visitante em uma variável de cookie:

### Exemplo

function setCookie(cname, cvalue, exdays) {  
    var d = new Date();  
    d.setTime(d.getTime() + (exdays\*24\*60\*60\*1000));  
    var expires = "expires="+ d.toUTCString();  
    document.cookie = cname + "=" + cvalue + ";" + expires + ";path=/";  
}

**Exemplo explicado:**

Os parâmetros da função acima são o nome do cookie (cname), o valor do cookie (cvalue) e o número de dias até que o cookie expire (exdays).

A função define um cookie juntando o cookiename, o valor do cookie e a string expirar.

## Uma função para obter um cookie

Em seguida, criamos uma função que retorna o valor de um cookie especificado:

### Exemplo

function getCookie(cname) {  
    var name = cname + "=";  
    var decodedCookie = decodeURIComponent(document.cookie);  
    var ca = decodedCookie.split(';');  
    for(var i = 0; i <ca.length; i++) {  
        var c = ca[i];  
        while (c.charAt(0) == ' ') {  
            c = c.substring(1);  
        }  
        if (c.indexOf(name) == 0) {  
            return c.substring(name.length, c.length);  
        }  
    }  
    return "";  
}

**Função explicada:**

Pegue o parâmetro cookiename como (cname).

Crie uma variável (nome) com o texto para procurar (cname + "=").

Decode a cadeia de cookies, para lidar com cookies com caracteres especiais, por exemplo '$'

Dividir document.cookie em ponto e vírgula em uma matriz chamada ca (ca = decodedCookie.split (';')).

Loop através da matriz ca (i = 0; i <ca.length; i ++) e leia cada valor c = ca [i]).

Se o cookie for encontrado (c.indexOf (name) == 0), devolva o valor do cookie (c.substring (name.length, c.length).

Se o cookie não for encontrado, volte "".

## Uma função para verificar um cookie

Por último, criamos a função que verifica se um cookie está configurado.

Se o cookie estiver configurado, ele exibirá uma saudação.

Se o cookie não estiver configurado, ele exibirá uma caixa de solicitação, solicitando o nome do usuário e armazenando o cookie de nome de usuário por 365 dias, chamando a função setCookie:

### Exemplo

function checkCookie() {  
    var username = getCookie("username");  
    if (username != "") {  
        alert("Welcome again " + username);  
    } else {  
        username = prompt("Please enter your name:", "");  
        if (username != "" && username != null) {  
            setCookie("username", username, 365);  
        }  
    }  
}

## Todos juntos agora

### Exemplo

function setCookie(cname, cvalue, exdays) {  
    var d = new Date();  
    d.setTime(d.getTime() + (exdays \* 24 \* 60 \* 60 \* 1000));  
    var expires = "expires="+d.toUTCString();  
    document.cookie = cname + "=" + cvalue + ";" + expires + ";path=/";  
}  
  
function getCookie(cname) {  
    var name = cname + "=";  
    var ca = document.cookie.split(';');  
    for(var i = 0; i < ca.length; i++) {  
        var c = ca[i];  
        while (c.charAt(0) == ' ') {  
            c = c.substring(1);  
        }  
        if (c.indexOf(name) == 0) {  
            return c.substring(name.length, c.length);  
        }  
    }  
    return "";  
}  
  
function checkCookie() {  
    var user = getCookie("username");  
    if (user != "") {  
        alert("Welcome again " + user);  
    } else {  
        user = prompt("Please enter your name:", "");  
        if (user != "" && user != null) {  
            setCookie("username", user, 365);  
        }  
    }  
}