



IN

DEV FULL STACK - JavaScript

**Aula 07**  
**Cookies**

## DEV FULL STACK - JavaScript

# Aula 07: Cookies

### Objetivos da aula:

1. Entender o funcionamento de cookies
2. Praticar o uso de cookies
3. Criar um pequeno projeto (carrinho de compras)

# Introdução

## O que são **cookies**?

Um cookie HTTP (um cookie web ou cookie de navegador) é um pequeno fragmento de dados que um servidor envia para o navegador do usuário. O navegador pode armazenar estes dados e enviá-los de volta em uma próxima requisição para o mesmo servidor.

## O que fazer com eles?

**Gerenciamento de sessão:** Logins, carrinhos de compra, placar de jogos ou qualquer outra atividade que deva ser guardada por um servidor.

**Personalização:** Preferências de usuário, temas e outras configurações.

**Rastreamento:** Registro e análise do comportamento de um usuário.

# Como se Criam os Cookies?

Há duas formas de criar cookies:

- **Cabeçalhos HTTP** – Ao enviar uma resposta ao navegador (ex: página HTML) o servidor pode incluir no cabeçalho desta resposta um ou mais cookies contendo informações. Estes cookies são manipulados por scripts CGI (PHP, Python,... ) no lado do servidor e por Javascript ou outra linguagem de script válida no lado do navegador (cliente).
- **Javascript** – Usando scripts JS tanto do lado do cliente quanto do lado do servidor.

Nesta aula abordaremos apenas a manipulação Javascript.

# Onde Os Cookies São Guardados?

Isso depende do sistema operacional e do navegador utilizado.

Por exemplo, no Windows 10:

1) Google Chrome

- "C:\Users\Your\_User\_Name\AppData\Local\Google\Chrome\User Data\Default"

2) Mozilla Firefox

- "C:\Users\Your\_User\_Name\AppData\Roaming\Mozilla\Firefox\Profiles"

3) Microsoft Edge

- "C:\Users\Your\_User\_Name\AppData\Local\Microsoft\Edge\User Data\Default"

No passado, os cookies eram armazenados em arquivos tipo texto separados que ficavam na pasta de instalação de cada browser. Hoje, a os navegadores tratam os cookies de formas diferentes, mas isso não é um problema.

**A localização física dos cookies não é importante**, já que eles podem ser manipulados por meio de scripts JS que executam dentro da página que está carregada no navegador.

# Limites

**Cada navegador define seus limites baseado em questões técnicas ou estratégicas.**

Existem recomendações propostas de limites de utilização de cookies, mas não há nenhuma obrigatoriedade imposta aos criadores de browsers.

**Tamanho máximo de um único cookie**

(recomendação: 4096 bytes)

**Número máximo de cookies por host**

(recomendação: 20 cookies)

**Número máximo de todos os cookies do navegador**

(recomendação: 300 cookies)

**Importante: o navegador Chrome só manipula cookies enviados via servidor. Páginas rodando localmente (sem servidor Web) não reconhecem esse recurso.**

# Criando um Cookie em Javascript

Para criar um cookie novo, basta criar uma chave e um valor associado a ela:

```
document.cookie = "usuario=maria";
```

Também é possível definir uma data de expiração para o cookie, o que faz com que o mesmo deixe de ser válido a partir desta data:

```
document.cookie = "usuario=maria; expires=Thu, 18 Dec 2021 12:00:00 UTC";
```

Outra opção disponível é definir o subdiretório ao qual o cookie pertence. Se isso não for especificado, o cookie é válido para o mesmo diretório da página que o criou (e seus subdiretórios):

```
document.cookie = "usuario=maria; expires=Thu, 18 Dec 2021 12:00:00 UTC;  
path=/docs";
```

# ATIVIDADE 1

Vamos construir um pequeno servidor web para testar nossos primeiros cookies.

- 1) Crie um pasta onde ficarão todas as páginas do servidor (ex-> **c:\MeuSite**)
- 2) Abra a janela de comandos do Windows (cmd), vá para a pasta que você criou e digite o comando:

```
python -m http.server
```

- 3) Copie para dentro da pasta que você criou o arquivo chamado **"Aula07\_exemplo01.htm"** que o professor vai fornecer.
- 4) Abra seu navegador no endereço <http://localhost:8000>
- 5) Clique no link **Aula07\_exemplo01.html**

**Lembre-se: isto é uma simulação onde sua máquina física está desempenhando dois papéis: (1) uma máquina cliente usando o navegador e (2) uma máquina servidora rodando o servidor web. Em situações reais isso geralmente funciona em máquinas distintas.**



# Lendo um Cookie Já Criado

Nos navegadores modernos, podemos usar o JS para ler os cookies existentes (desde que tenham sido criados pelo mesmo navegador), porém eles estão todos guardados em uma mesma string (**`document.cookie`**).

Isso obriga o programador a aplicar alguma lógica no tratamento desta string para recuperar cada chave e valor de cookie em separado.

Veja um exemplo simples de recuperação de cookies na página **`Aula07_exemplo02.html`**, que foram criados no exemplo 1.



# Alterando um Cookie

Para alterar o valor de um cookie basta repetir a mesma string de criação, alterando os valores desejados para a mesma chave.

Ou seja: a alteração do valor da chave acontece com o programador “escrevendo por cima” dos valores anteriores da chave.

Veja o exemplo:

**[Aula07\\_exemplo03.html](#)**



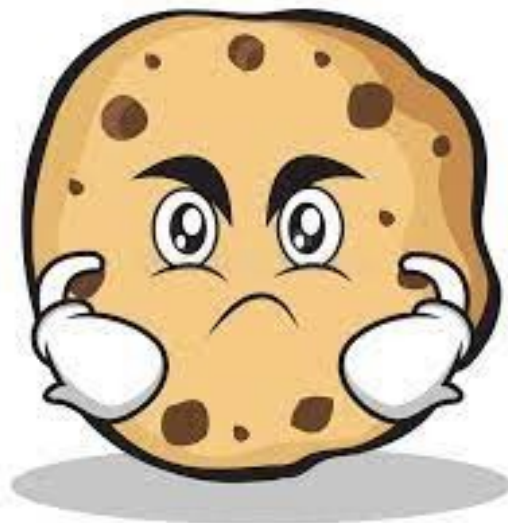
# Apagando um Cookie

Para eliminar um cookie basta alterar o valor da chave para vazio e alterar sua data de expiração, colocando uma data no passado. Isso fará o cookie perder a validade.

Se apenas alterarmos o valor da chave para vazio sem alterar a data de expiração, o cookie continuará existindo (embora não carregue nenhum valor).

Veja o exemplo:

**[Aula07\\_exemplo04.html](#)**



# ATIVIDADE 2



Construa uma página que, ao carregar, solicita (via prompt) o nome do usuário e, em seguida, exibe uma mensagem de boas vindas.

A página deve verificar se existe um cookie com a seguinte estrutura: **primeiro=nome\_primeiro\_visitante**.

Se o cookie não existir, ele deve ser criado usando o nome informado e exibir uma mensagem na página informando ao usuário que ele é o primeiro a visitar a página.

Caso o cookie já exista, a página dá boas vindas ao novo visitante, mas informa o nome de quem foi o primeiro a acessá-la.

Dica → para saber se uma determinada chave existe nos cookies, podemos usar a função **indexOf("chave=")** que foi usada no exemplo 2 (a função retorna -1 se não encontrar a chave)

OBS: Lembre-se de apagar os cookies do exercício anterior antes de iniciar seus testes (ou levar a existência deles em consideração em sua lógica).

# Projeto:

## Protótipo de um Carrinho de Compras

Sua empresa foi contratada para criar o **protótipo** de um carrinho de compras para vendas online para a loja de revistas em quadrinhos **HQTop**.

O protótipo deve ter duas páginas:

- (1) Um mostruário de revistas em quadrinhos à venda contendo um botão ao lado de cada revista para colocá-la no carrinho de compras.
- (2) Uma página que exhibe o conteúdo do carrinho de compras e permite que o usuário finalize a compra, retire itens do carrinho ou esvazie o carrinho completamente.

**OBS:** O objetivo deste exercício **não é construir um carrinho de compras totalmente funcional**, e sim praticar a manipulação de cookies.

# Referências

**Aqui estão alguns links interessantes para leitura complementar:**

**Mais informações sobre Cookies:**

**<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Cookies>**