

# Programação para Internet 2

# Cookies

# Cookies

O que é um Cookie? Um biscoito? Quando um usuário acessa um site normalmente são armazenadas algumas informações. O usuário deve estar ciente de toda informação pessoal que é armazenada!

Você provavelmente já acessou um site de compras. Você navegou pelos produtos, procurando por algo específico. Vamos supor que o produto desejado era uma TV. Naquele dia você não ia comprar a TV, então saiu do site sem fazer nenhum pedido. No outro dia, você visitou o site novamente e para sua surpresa, o site já sabia quem era você e te ofereceu uma TV!

Como ele sabia quem era você?

# Cookies

Muito provavelmente, o site utilizou cookies. Um cookie é um dado que é armazenado no **cliente**. Esse dado é uma string. Qualquer site pode armazenar cookies no cliente, desde que no seu navegador esteja habilitada a opção de cookies.

Os cookies são armazenados pelo navegador, portanto, um cookie de um navegador não é compartilhado com outro navegador.

Uma outra característica do cookie é que somente o site que criou o cookie poderá acessá-lo. Então não tem como um site ficar lendo os cookies dos outros sites. O navegador que controla essa segurança. O cookie é restrito por **domínio**.

# Cookies

Como você é um aluno inteligente, já deve ter percebido que se o cookie é armazenado no cliente, então o cliente pode alterá-lo da forma que quiser. Este é um motivo pelo qual deve se tomar cuidado quando se utiliza cookies. As informações que estão no cliente podem ser alteradas de qualquer forma e o servidor deve sempre estar preparado para possíveis fraudes.

Mas será que o cookie é necessário? Não poderíamos guardar as informações no HTTP?

# Cookies

**O protocolo HTTP não armazena estado.** Isso significa que cada requisição é uma requisição sem nenhuma ligação com a anterior. Por este motivo é que os cookies são importantes. Veremos mais adiante que existem outras formas de armazenar dados além do cookie.

Agora que vocês já sabem um pouco sobre cookies, como poderíamos utilizá-los?

# Cookies

Podemos pensar em muitas utilidades. A seguir estão alguns exemplos:

- ⇒ Autenticação: para saber se o usuário está logado no sistema.
- ⇒ Carrinho de compras: o usuário pode criar um carrinho sem logar no sistema.
- ⇒ Anúncios: para não repetir anúncios.
- ⇒ Personalização: conforme as pesquisas do usuário no site, as propagandas seriam mais direcionadas.

# Cookies

Vamos aprender mais um pouco sobre os cookies. Primeiramente eles não servem para armazenar grandes quantidades de dados. Os navegadores mais antigos aceitam somente 20 cookies por domínio. Os mais novos aceitam 50.

Outra limitação é que um cookie não pode ter mais que 4KB. Este tamanho leva em consideração a chave, o valor e outros parâmetros. Também não é bom ter chaves muito grandes, pois podem dar problemas.

Vamos pensar o seguinte: se os cookies ficam armazenados no cliente e o HTTP não guarda estado, como que o servidor sabe os valores dos cookies?

# Cookies

Em toda requisição do cliente, o navegador verifica se existe algum cookie válido daquele domínio e o envia no cabeçalho do HTTP (Cookie). Isso significa que se tiverem muitos cookies grandes, o cliente consumirá muita banda de upload e o servidor de download.

E como que o servidor define um cookie? Ele é definido no servidor e enviado no cabeçalho HTTP de resposta (SetCookie). O servidor não precisa enviar os cookies em todas as respostas. Ele só envia quando for definir ou alterar um cookie.

Para acessar os cookies no PHP, basta utilizar a variável superglobal **\$\_COOKIE**. Ela é um array e a chave é o nome do cookie.

# Cookies

Para criar um cookie em PHP, usamos a função **setcookie()**. Ela possui 7 parâmetros, sendo 1 obrigatório e 6 opcionais. O obrigatório é o nome do cookie. Os outros parâmetros são:

- `valor = ""`: valor do cookie.
- `expiração = 0`: timestamp do unix. Por padrão o cookie é deletado após fechar o navegador.
- `caminho = ""`: caminho do domínio. Por padrão é a pasta atual.
- `domínio = ""`: domínio do cookie. Por padrão o domínio é o atual.
- `seguro = false`: enviado somente com HTTPS.
- `somente http = false`: enviado somente em requisição HTTP.

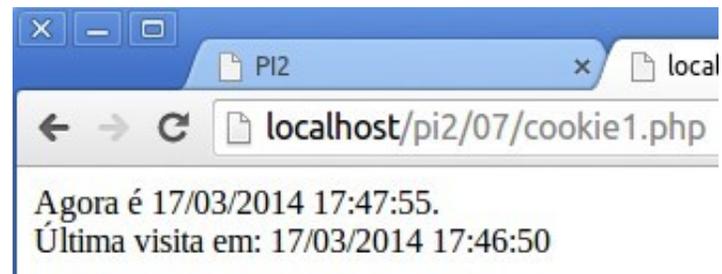
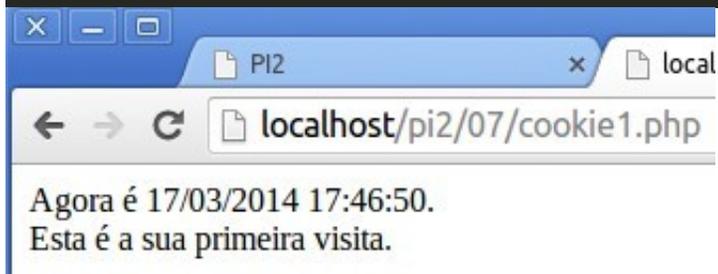
# Cookies

Vamos por a mão na massa. A seguir um exemplo de um cookie que guarda sempre a data da última visita.

```
<?
header('Content-Type: text/html; charset=utf-8');

if (isset($_COOKIE['ultima_visita'])) {
    $mensagem = "Última visita em: $_COOKIE[ultima_visita]";
} else {
    $mensagem = "Esta é a sua primeira visita.";
}

$agora = date('d/m/Y H:i:s');
setcookie('ultima_visita', $agora);
echo "Agora é $agora.<br />$mensagem";
```



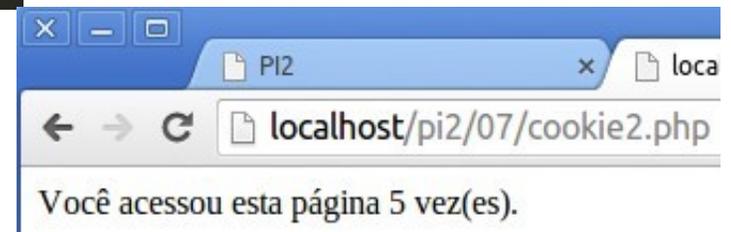
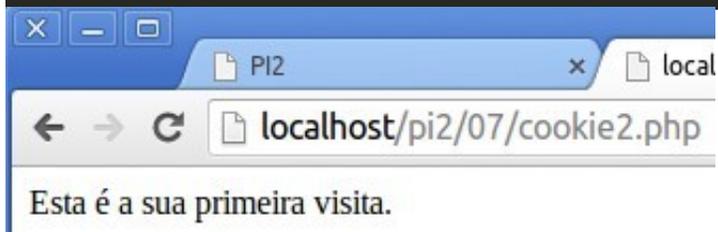
# Cookies

Vamos definir para o cookie expirar em 1 minuto. Podemos criar um timestamp do Unix (int) utilizando a função **time()**. Ela retorna o tempo atual. O timestamp Unix conta os segundos a partir de 01/01/1970 00:00:00.

```
<?
header('Content-Type: text/html; charset=utf-8');

if (isset($_COOKIE['contador'])) {
    $mensagem = "Você acessou esta página $_COOKIE[contador] vez(es).";
    $contador = $_COOKIE['contador'] + 1;
} else {
    $mensagem = "Esta é a sua primeira visita.";
    $contador = 1;
}

setcookie('contador', $contador, time()+60);
echo $mensagem;
```



# Cookies

Como eu faço para deletar um cookie? Neste caso, basta definir a data expiração para o passado.

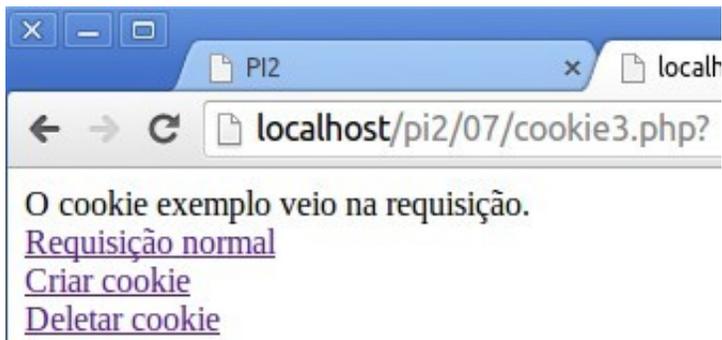
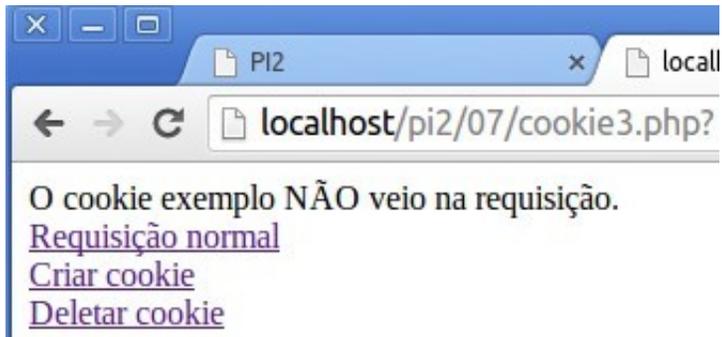
```
<?
header('Content-Type: text/html; charset=utf-8');

if (isset($_COOKIE['exemplo'])) {
    $mensagem = "O cookie exemplo veio na requisição.";
} else {
    $mensagem = "O cookie exemplo NÃO veio na requisição.";
}

if ($_GET['acao'] == 'criar') {
    setcookie('exemplo', 'exemplo', time()+3600); //1 hora
    $mensagem .= "<br />O cookie foi criado.";
} elseif ($_GET['acao'] == 'deletar') {
    setcookie('exemplo', 'exemplo', time()-3600);
    $mensagem .= "<br />O cookie foi deletado.";
}

echo "$mensagem<br/>";
?>
<a href="#">Requisição normal</a><br />
<a href="?acao=criar">Criar cookie</a><br />
<a href="?acao=deletar">Deletar cookie</a>
```

# Cookies



# Referências

NIEDERAUER, Juliano. **Desenvolvendo websites com PHP**. 2. ed. São Paulo: Novatec, 2011. 301 p. ISBN 9788575222348.