

Este curso é de propriedade do aplicativo [SoloLearn](#) e foi traduzido por [Ana Beatriz Augusto](#) usando os recursos [Reverso Context](#) e [Google Tradutor](#).

## **Variáveis `$_SERVER`: Nome do script**

Uma superglobal é uma variável pré definida que é sempre acessível, independente do escopo. Você pode acessar as superglobais do php através de qualquer função, classe ou arquivo.

As variáveis superglobais do php são `$_SERVER`, `$GLOBALS`, `$_REQUEST`, `$_POST`, `$_GET`, `$_FILES`, `$_ENV`, `$_COOKIE`, `$_SESSION`.

`$_SERVER` é uma array que inclui informações como cabeçalhos, caminhos e localizações de scripts. Os itens dentro dessa array são criados pelo servidor web.

`$_SERVER[ 'SCRIPT_NAME' ]` retorna o caminho do script atual:

```
<?php
```

```
echo $_SERVER[ 'SCRIPT_NAME' ];
```

```
//O resultado será "/somefile.php"
```

```
?>
```

# Variáveis \$\_SERVER:

## Nome da máquina

`$_SERVER[ 'HTTP_HOST' ]` retorna host da máquina.

```
<?php
    echo
$_SERVER[ 'HTTP_HOST' ];
    //O resultado será
"localhost"
?>
```

Este método pode ser útil quando você tem várias imagens no seu servidor e precisa transferir o site para outro host. Invés de mudar o caminho de cada imagem, você pode fazer o seguinte:

Crie um arquivo config.php que recebe o caminho das suas imagens:

```
<?php
    $host =
$_SERVER[ 'HTTP_HOST' ];
    $image_path =
$host.' /images/';
?>
```

Use o arquivo config.php nos seus scripts:

```
<?php  
    require 'config.php';  
    echo '  
  <p>Nome: <input type="text" name="name" /></p>  
  <p>Idade: <input type="text" name="age" /></p>  
  <p><input type="submit" name="submit" value="Enviar"  
/></p>  
</form>
```



O atributo `action` especifica que quando o formulário é enviado, os dados são mandados para um arquivo php nomeado `first.php`

Elementos de formulários HTML tem nomes, que serão usados quando for acessado os dados com php.

Agora, quando temos um formulário HTML com o atributo `action` definido para nosso arquivo php, podemos acessar os dados do formulário postado usando a array associativa `$_POST`.

No arquivo `first.php`:

```
<html>
<body>
    Bem vindo <?php echo $_POST["name"]; ?><br />
    Sua idade: <?php echo $_POST["age"]; ?>
</body>
</html>
```

A array superglobal `$_POST` guarda pares de chave e valor. Nos pares, chaves são os nomes que o formulário controla e valores são os dados enviados pelo usuário.

# GET e POST

Os dois métodos de se enviar formulários são GET e POST. Informações enviadas para um formulário via método POST é invisível a outros pois todos os nomes e/ou valores são incorporados dentro do corpo do pedido HTTP. Também, não há limites na quantidade de informação a ser enviada.

Além disso, POST suporta funcionalidades avançadas como suporte para entradas binárias de múltiplas partes enquanto carrega arquivos para o servidor.

Entretanto, não é possível favoritar a página, como os valores enviados não são visíveis.

Informação enviada através de um formulário usando o método GET é visível para todos (todos os nomes das variáveis e valores são mostrados na URL). GET também estabelece limites na quantidade de informação que pode ser mandada - aproximadamente 200 caracteres.

Entretanto, por causa que as variáveis são mostradas na URL, é possível favoritar a página, o que pode ser útil em algumas situações.

Por exemplo:

```
<form action="actionGet.php" method="get">
  Nome: <input type="text" name="name" /><br /><br />
  Idade: <input type="text" name="age" /><br /><br />
  <input type="submit" name="submit" value="Enviar" />
</form>
```

actionGet.php

```
<?php
```

```
echo "Olá " . $_GET['name'] . " . ";
```

```
echo "Você tem " . $_GET['age'] . " anos.";
```

```
?>
```

Agora, o formulário foi enviado para o actionGet.php e você pode ver os dados enviados na URL:



GET NUNCA deve ser usado para mandar senhas ou outras informações sensíveis! Quando se usa POST ou GET, validar os dados através de filtragem e de processamento é importante para proteger seu formulário de hackers!

# \$\_SESSION

Usando uma sessão, você pode armazenar informações em variáveis, para serem usadas em múltiplas páginas.

Informação não é armazenada no computador do usuário, como acontece com os cookies.

Por padrão, variáveis de sessão duram até o usuário fechar o navegador.

Começando uma sessão php:

Uma sessão é iniciada usando a função `session_start()`.

Use a global php `$_SESSION` para definir variáveis de sessão.

```
<?php
    // Iniciando a sessão
    session_start();
    $_SESSION['color'] =
"red";
    $_SESSION['name'] =
"John";
?>
```

Agora, as variáveis de sessão `color` e `name` são acessíveis em múltiplas páginas, durante a sessão inteira.

A função `session_start()` deve ser a primeira coisa em seu documento. Antes de qualquer tag HTML.

Outra página pode ser criada que pode acessar as variáveis de sessão que definimos na página anterior:

```
<?php
```

```
// Inicia a sessão
```

```
session_start();
```

```
?>
```

...

```
<body>
```

```
<?php
```

```
echo "Seu nome é " . $_SESSION[ 'name' ];
```

```
// Mostra "Seu nome é John"
```

```
?>
```

```
</body>
```

...



Suas variáveis de sessão permanecem disponíveis na superglobal `$_SESSION` até você fechar sua sessão.

Todas as variáveis de sessão globais podem ser removidas manualmente usando `session_unset()`. Você também pode destruir a sessão com `session_destroy()`.

# \$\_COOKIE

Cookies são frequentemente usados para identificar o usuário. Um cookie é um pequeno arquivo que o servidor incorpora no computador do usuário. Cada vez que o mesmo computador pede uma página através de um navegador, ele mandará o cookie, também. Com PHP, você consegue criar e recuperar valores de cookies.

Crie cookies usando a função `setcookie()`:

```
setcookie(nome, valor, expira, caminho, dominio,  
seguro, apenas_http);
```

nome: Especifica o nome do cookie.

valor: Especifica o valor do cookie.

expira: Especifica (em segundos) quando o cookie expirará. O valor:  $\text{time()}+86400*30$ , define que o cookie expirará em 30 dias. Se este parâmetro for omitido ou definido para 0, o cookie irá expirar no final da sessão (quando o navegador se fechar). O padrão é 0.

caminho: Especifica o caminho do servidor do cookie. Se definido como "/", o cookie será disponível dentro do domínio inteiro. Se definido como "/php/", o cookie será disponível apenas dentro do diretório php e em todos os subdiretórios do php. O valor padrão é o atual diretório onde o cookie está sendo definido.

**dominio:** Especifica o nome do domínio do cookie. Para tornar o cookie disponível em todos os subdomínios de exemplo.com, defina o domínio como "exemplo.com".

**seguro:** Especifica se o cookie deve, ou não, apenas ser transmitido através de uma conexão segura de HTTPS. True indica que o cookie apenas será definido se uma conexão segura existir. O padrão é False.

**apenas\_http:** Se true, o cookie será acessível apenas através do protocolo HTTP (o cookie não será acessível para linguagens de script). Usar o apenas\_http ajuda a reduzir roubos de identidade usando ataques XSS. O padrão é False.

O parâmetro nome é o único que é requerido. Todos os outros parâmetros são opcionais.

O seguinte exemplo cria um cookie chamado "user" com o valor "John". O cookie irá expirar depois de 30 dias, que é escrito como  $86,400 * 30$ , no qual 86,400 segundos = um dia. O '/' significa que o cookie está disponível ao longo de todo site.

Nós, então, pegamos o valor do cookie "user" (usando a variável global `$_COOKIE`). Nós também usamos a função `isset()` para descobrir se o cookie está definido →

```
<?php
$value = "John";
setcookie("user", $value,
time() + (86400 * 30),
'/');
if(isset($_COOKIE['user']))
{
    echo "Value is: ".
$_COOKIE['user'];
}
//Mostra "Valor é: John"
?>
```

A função `setcookie()` deve aparecer ANTES da tag `html`.  
O valor do cookie é automaticamente codificado quando o cookie é enviado e é automaticamente decodificado quando é recebido. No entanto, NUNCA armazene informação sensível nos cookies.

Este curso é de propriedade do aplicativo [SoloLearn](#) e foi traduzido por [Ana Beatriz Augusto](#) usando os recursos [Reverso Context](#) e [Google Tradutor](#).