

MÓDULO

PROGRAMAÇÃO PHP

UNIDADE

PHP E JAVASCRIPT/AJAX

ÍNDICE

OBJETIVOS.....	3
INTRODUÇÃO.....	4
1. NOÇÕES BÁSICAS DE JAVASCRIPT	5
2. ENVIAR DADOS DO PHP PARA O JAVASCRIPT.....	9
3. ENVIAR DADOS DO JAVASCRIPT PARA O PHP.....	11
4. PHP, JAVASCRIPT E JSON	14
CONCLUSÃO	19
AUTOAVALIAÇÃO	21
SOLUÇÕES.....	25
PROPOSTAS DE DESENVOLVIMENTO DO ESTUDO.....	26
BIBLIOGRAFIA	27

OBJETIVOS

Com esta unidade didática, pretende-se que desenvolva os seguintes objetivos de aprendizagem:

- Saber como comunicar PHP com JavaScript e JavaScript com PHP.
- Aprender a utilizar PHP e JavaScript para lidar com ficheiros JSON.
- Compreender a interação entre PHP e AJAX.

INTRODUÇÃO

JavaScript é uma linguagem de programação web executada no lado do cliente, ou seja, o servidor serve os dados que são interpretados pelo navegador e é este que executa os códigos do script.

O PHP é uma linguagem de programação do lado do servidor com a qual é possível interagir, o que por vezes torna a programação muito mais fácil.

Nesta unidade, aprenderá a interagir com estas duas linguagens.

1. NOÇÕES BÁSICAS DE JAVASCRIPT

JavaScript é uma linguagem executada e interpretada no cliente, ou seja, o navegador é o responsável por traduzir o código em eventos ou conteúdo.

Pode criar ficheiros .js para conter o código JavaScript e invocá-los nas páginas web através da seguinte linha:

```
<script type="text/javascript" src="ficheiro.js"></script>
```

Ou inserir o código diretamente na página web através das seguintes tags:

```
<script type="text/javascript">  
Código  
</script>
```

Uma singularidade do JavaScript é que, como o PHP, as instruções devem terminar com um ponto e vírgula (;).

Segue-se uma rápida revisão de algumas noções básicas de JavaScript.

Criar variáveis

Em JavaScript, para criar uma variável, basta defini-la com a palavra reservada **var**. Isso pode ser feito em qualquer lugar da página web, tendo em consideração que o escopo (se é uma variável global ou local) das variáveis é tratado de forma semelhante ao PHP, ou seja, uma variável que é criada dentro de uma função só será acessível a partir dessa função, enquanto uma variável criada fora das funções estará acessível a partir de qualquer ponto.

Exemplo de definição de variável:

```
var variavel;
```

Criar funções

Em JavaScript também é possível definir funções, como em PHP, com a palavra reservada **function**, que é seguida dos parâmetros que pretender. O nome deve começar com letras (não números) e pode devolver um valor.

```
<script type="text/javascript">  
function soma(um,dois){  
    return um + dois;  
}  
</script>
```

Ao contrário das outras linguagens de programação, quando estas são executadas no cliente, as funções podem ser chamadas a partir de eventos do navegador, como um clique, uma passagem do cursor sobre uma imagem, etc. Para isso, basta chamar o nome da função e os seus parâmetros a partir dos eventos:

```
<a href="#" onclick="alert(soma(2,7))">soma 2 + 7</a>
```

A função JavaScript **alert** mostrará uma mensagem no ecrã, numa caixa de aviso, com o conteúdo que passar pela função.

Operadores e estruturas de controle

Em JavaScript, os operadores mais comuns são:

- Sinal de mais (+), para a soma.
- Sinal de menos (-), para a subtração.
- Asterisco (*), para a multiplicação.
- Uma barra (/), para a divisão.
- Sinal de igual (=), para corresponder.
- Dois sinais de igual (==), para comparar valores.
- Ponto de exclamação (!), para a negação.
- Sinal de maior (>), "maior que".
- Sinal de menor (<), "menor que".
- Sinal de maior e um sinal de igual (>=), "maior ou igual a".
- Sinal de menor e um sinal de igual (<=), "menor ou igual a".

Quanto às strings, também são concatenadas usando o sinal de mais (+):

```
var texto = 'isto é um ' + 'curso';
```

As estruturas de controle mais utilizadas são:

- **if:** funciona da mesma forma que o if no PHP e a sua utilização é semelhante:

```
if (2 == 2){  
  alert('iguais')  
}
```

- **for:** funciona de maneira semelhante ao do PHP:

```
for (i=0;i<=4;i++){  
  alert(i);  
}
```

- **while:** o uso é o mesmo do que em PHP:

```
while(x<6){  
  alert(x);  
  x = x + 1;  
}
```

2. ENVIAR DADOS DO PHP PARA O JAVASCRIPT

Uma das situações mais comuns é querer utilizar alguns dados gerados ou retirados do PHP no código JavaScript. Para isso, basta inserir o código PHP no JavaScript, mas, como se trata de uma linguagem de servidor, a página apresentada não carrega o código PHP. O resultado é, porém, transparente para os navegadores que só veem o resultado. O exemplo mais simples é mostrar o valor de um dado criado em PHP através de um alert de JavaScript:

```
<?php
$var1 = 6;
$var2 = 4;
?>
<script type="text/javascript">
function soma(){
alert('<?echo "Valor da soma " . ($var1 + $var2)?>');
}
</script>
```

Como observa no exemplo, pode inserir valores dentro do JavaScript, mas também pode inserir qualquer outro tipo de código.

O exemplo a seguir mostra um **alert** de JavaScript, mas o conteúdo depende de uma variável da linguagem do servidor:

```
function mostrar(){  
    alert('<?php if ($v == "s") { echo "ok" }else{ echo "ko" } ?>');  
}
```

Desta forma, poderia passar qualquer valor que pretendesse para o JavaScript a partir do PHP, já que o JavaScript só vê o resultado das operações do PHP.

Outra opção que o PHP permite é escolher qual o código JavaScript que será executado, ou seja, sendo uma linguagem de servidor, pode criar uma condição no PHP que faça aparecer ou não um código JavaScript. Por exemplo:

```
<?  
$variavel = "sim";  
if($variavel = "sim"){  
    echo "(<script>alert('Teste')</script>");  
}  
?>
```

O código acima só será executado se a variável tiver o valor "sim". Ao observar o código que chega ao navegador, perceberia que nada realmente o atingiu.

3. ENVIAR DADOS DO JAVASCRIPT PARA O PHP

A única maneira de enviar dados do JavaScript para o PHP é utilizando o AJAX. O JavaScript permite abrir um canal assíncrono com o servidor, ou seja, permite fazer a chamada a uma página no servidor sem que o utilizador perceba. Nesta chamada, feita a partir de JavaScript, é possível incluir valores JavaScript para a URL a partir da página PHP e, assim, enviar parâmetros de uma linguagem para outra.

Este exemplo divide-se em dois ficheiros: o primeiro, exemplo.php, contém as funções JavaScript para fazer a chamada AJAX para um PHP com variáveis JavaScript; o segundo, soma.php, recolhe essas variáveis e faz a soma.

Exemplo 01

■ exemplo.php:

```
<html>
<head></head>
<body>
<script type="text/javascript">
/*função para criar o objeto AJAX que permite a conexão*/
function getHTTPObject(){
if (window.ActiveXObject)
return new ActiveXObject("Microsoft.XMLHTTP");
```

```
else if (window.XMLHttpRequest)
return new XMLHttpRequest();
else {
alert("Não suportado");
return null;
}
}
function passarPhp(){
var valor1 = 5;
var valor2 = 6;
httpObject = getHTTPObject();
if (httpObject != null) {
/*abrir a conexão AJAX com o Php enviando as variáveis*/
httpObject.open("GET", "soma.php?v1=" + valor1 + "&v2=" +valor2,
true);
httpObject.onreadystatechange = function(){
if(httpObject.readyState == 4){
/*quando terminar de carregar soma.php mostra o conteúdo criado */
alert(httpObject.responseText);
}
}
httpObject.send(null);
}
}
</script>
<a href="#" onclick="passarPhp()">envio de teste</a>
/* Este link gera o evento que chama o AJAX */
</body>
</html>
```

- soma.php:

```
<?php  
echo $_GET['v1'] + $_GET['v2'];  
?>
```

4. PHP, JAVASCRIPT E JSON

O JSON é um sistema de gestão de dados em ficheiros simples estruturados. A estrutura destes ficheiros assemelha-se a uma árvore cujos ramos estão pendurados e, por sua vez, outros ramos estão neles pendurados. Segue-se um exemplo de um ficheiro com valores JSON:

```
var JSONstring =
{
  "nome": "teste",
  "email": "email@testes.com",
  "passatempo":
  [
    {
      "passatempo": "desporto",
      "marcado": "true"
    },
    {
      "passatempo": "ler",
      "marcado": "true"
    },
    {
      "passatempo": "ouvir música",
      "marcado": "false"
    }
  ]
}
```



```
}  
]  
};
```

É muito utilizado em JavaScript para o envio de informações, pois é muito rápido de criar e de ler.

Em PHP, existem duas funções, chamadas **json_decode** e **json_encode**, que decodificam uma string JSON num array, e que codificam um array numa string JSON, respetivamente. Graças ao AJAX, é possível passar grandes quantidades de informações entre PHP e JavaScript, e vice-versa.

Exemplo 02

Neste exemplo, aprenderá a passar grandes quantidades de dados de JavaScript para PHP utilizando o JSON, bem como devolver outras quantidades de dados de PHP para JavaScript utilizando JSON e AJAX.

Serão utilizados dois ficheiros: o exemplo2.php irá recolher os dados de um formulário JavaScript e convertê-los numa string JSON. Esta string será enviada para o segundo ficheiro, chamado compilador.php, que, por sua vez, irá devolver os dados no formato JSON para o exemplo2.php, com AJAX.

■ exemplo2.php:

```
<html>  
<head></head>  
<body>  
<script>  
// criar o objeto AJAX para carregar ficheiros do servidor.  
function getHTTPObject(){  
if (window.ActiveXObject)  
return new ActiveXObject("Microsoft.XMLHTTP");  
else if (window.XMLHttpRequest)  
return new XMLHttpRequest();
```

```
else {
    alert("Não suportado");
    return null;
}

function validate(){
    //criar uma string JSON a partir dos dados do formulário
    var p = document.forms['pessoal'];
    var JSONObject = new Object;
    JSONObject.firstname = p['nome'].value;
    JSONObject.email = p['email'].value;
    JSONObject.passatempo = new Array;
    for(var i=0; i<2; i++){
        JSONObject.passatempo[i] = new Object;
        JSONObject.passatempo[i].passatempo = p['passatempo'][i].value;
        JSONObject.passatempo[i].marcado = p['passatempo'][i].checked;
    }
    JSONstring = JSON.stringify(JSONObject);
    runAjax(JSONstring);
}

var request;
function runAjax(JSONstring){
    request = getHTTPObject();
    //chamar sendData quando o ficheiro AJAX é carregado
    request.onreadystatechange = sendData;
    request.open("GET", "compilador.php?json="+JSONstring,
    true);
    request.send(null);
}

function sendData(){
    if(request.readyState == 4){
        var JSONtext = request.responseText;
```

```
//esta função JavaScript converte uma string em JSON
var JSONObject = JSON.parse(JSONtext);
var msg = "Erros: "+JSONObject.errorsNum+
"\n- "+JSONObject.error[0]+
"\n- "+JSONObject.error[1];
//mostrar os dados depois de carregá-los como JSON
alert(msg);
}
}
</script>
</head>
<body>
<form name="pessoal" action="" method="POST">
nome <input type="text" name="nome"/><br/>
email <input type="text" name="email"/><br/>
passatempo
<input type="checkbox" name="passatempo" value="desporto"/>
desporto
<input type="checkbox" name="passatempo" value="ler"/> ler
<input type="button" name="valid" value="Validar"
onclick="validate()"/>
</form>
</body>
</html>
```

■ compilador.php:

```
<?php
$decoded = json_decode($_GET['json']);
// tratar os dados json inserindo-os na base de dados ou
comparando-os com algum valor, criando uma resposta num array
$json = array();
$json['errorsNum'] = 2;
$json['error'] = array();
$json['error'][] = 'email invalido!';
$json['error'][] = 'Bom passatempo!';
//codificar o array no formato JSON
$encoded = json_encode($json);
//escrever para devolvê-lo
echo $encoded;
?>
```

CONCLUSÃO

A integração do JavaScript com linguagens de servidor confere poder de carregamento às páginas da web (sem precisar de as atualizar) e facilita a interação com os utilizadores.

Esta unidade é especialmente importante do ponto de vista do utilizador, pois é bastante positivo não ter de esperar por um novo carregamento sempre que se clica numa nova opção de um website, o que torna toda a experiência mais rápida e, por sua vez, satisfatória.

AUTOAVALIAÇÃO

1. Qual é a diferença entre PHP e JavaScript?

- a) O JavaScript é executado no navegador e o PHP no servidor.
- b) O PHP consome mais recursos.
- c) O JavaScript é mais poderoso que o PHP.
- d) O PHP é executado no navegador e o JavaScript no servidor.

2. Qual das seguintes afirmações está incorreta?

- a) Os dados JavaScript e PHP não podem ser partilhados diretamente.
- b) O JavaScript é executado no servidor e fornece os dados ao navegador.
- c) O PHP não pode exibir dados sem solicitá-los ao servidor.
- d) Num website, é possível referir-se a uma variável PHP no JavaScript.

3. Qual é o símbolo para concatenar strings em JavaScript?

- a) O “e” comercial (&), como no PHP.
- b) O sinal de mais (+), como no PHP.
- c) O sinal de mais (+), ao contrário do (&) do PHP.
- d) Não é possível concatenar strings em JavaScript.

4. Complete a seguinte frase: Para enviar dados de uma variável PHP para o código JavaScript...

- a) ... deverá passar a variável por uma base de dados.
- b) ... deverá utilizar o AJAX.
- c) ... basta inserir o código PHP dentro do código JavaScript.
- d) ... basta inserir o código JavaScript no PHP.

5. O que é que é possível inserir em JavaScript?

- a) Apenas variáveis PHP.
- b) Apenas constantes PHP.
- c) Qualquer código PHP.
- d) Apenas funções PHP.

6. Qual é a única opção para integrar o código JavaScript em PHP?

- a) Fazer referência às variáveis.
- b) Criar variáveis em ficheiros externos para ler a partir deles.
- c) Chamar as variáveis JavaScript da mesma forma que as do ASP ou PHP.
- d) Usar chamadas assíncronas para ficheiros PHP no servidor.

7. O que é JSON?

- a) Uma linguagem de programação.
- b) Uma API do JavaScript.
- c) Uma API do PHP.
- d) Um sistema de gestão de dados em ficheiros simples e estruturados.

8. Qual é a função PHP utilizada para codificar uma string para JSON?
- a) json_encode.
 - b) encode_json.
 - c) json_codify.
 - d) codify_json.
9. Qual é a função JavaScript utilizada para converter uma string em JSON?
- a) JSON.
 - b) JSON.parse.
 - c) JSON.encode.
 - d) JSON.codify.
10. Qual é o objeto AJAX que permite abrir uma conexão com o servidor?
- a) XMLconection.
 - b) XMLHttpRequest.
 - c) XMLHttpRequest.
 - d) XMLHttpRequest.

SOLUÇÕES

1.	a	2.	b	3.	c	4.	c	5.	c
6.	d	7.	d	8.	a	9.	b	10.	c

PROPOSTAS DE DESENVOLVIMENTO DO ESTUDO

Visite a seguinte página web para saber mais sobre JSON:

- <https://www.json.org/json-pt.html>

BIBLIOGRAFIA

- Cabezas, L. M. (2010), *Php 5*. Madrid: Anaya Multimedia.
- The PHP Group (2001), "SimpleXML". Disponível em:
https://www.php.net/manual/pt_BR/book.simplexml.php
Consultado a 29 de março de 2021.

