

# MÓDULO

# JAVASCRIPT/AJAX

UNIDADE

## INTRODUÇÃO AO JAVASCRIPT



## ÍNDICE

---

<b>OBJETIVOS.....</b>	<b>3</b>
<b>INTRODUÇÃO.....</b>	<b>4</b>
<b>1. O QUE É O JAVASCRIPT?.....</b>	<b>5</b>
1.1. DEFINIÇÃO.....	5
1.2. HISTÓRIA.....	5
1.3. PARA QUE SERVE O JAVASCRIPT?.....	6
<b>2. NOMENCLATURA E COMENTÁRIOS.....</b>	<b>7</b>
2.1. NOMENCLATURA.....	7
2.2. COMENTÁRIOS .....	10
<b>3. CONCEITOS BÁSICOS.....</b>	<b>12</b>
3.1. ESPAÇOS EM BRANCO.....	12
3.2. USO DE PLICAS .....	12
3.3. MODOS DE EXECUTAR JAVASCRIPT .....	13
3.4. CARACTERES ESPECIAIS .....	16
3.5. LINHAS DE CÓDIGO .....	16
<b>CONCLUSÃO.....</b>	<b>17</b>
<b>AUTOAVALIAÇÃO.....</b>	<b>19</b>
<b>SOLUÇÕES.....</b>	<b>23</b>
<b>PROPOSTAS DE DESENVOLVIMENTO DO ESTUDO.....</b>	<b>24</b>
<b>BIBLIOGRAFIA .....</b>	<b>25</b>



## OBJETIVOS

---

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

- Conhecer brevemente a origem e história da linguagem JavaScript.
- Saber descrever a linguagem de programação JavaScript.
- Perceber a ferramenta e a utilidade do JavaScript.
- Aprender alguns conceitos básicos e essenciais.

## INTRODUÇÃO

---

Nesta unidade didática, aprenderá um pouco sobre a história do JavaScript, conhecerá as ferramentas necessárias para criar o código e verá também como e para que podemos usar a linguagem JavaScript. Nesta unidade didática abordaremos ainda um pouco da lógica da programação JavaScript e alguns eventos importantes e úteis.

# 1. O QUE É O JAVASCRIPT?

---

## 1.1. DEFINIÇÃO

JavaScript é uma linguagem de programação para websites, executada no lado do cliente, isto é, se estiver numa página web que usa código JavaScript, o navegador, juntamente com o website, faz download do código, e é o próprio navegador que executa esse código JavaScript.

O JavaScript é usado em muitos websites e, uma vez online, através do navegador é possível ver o seu código-fonte e verificar o código JavaScript utilizado.

## 1.2. HISTÓRIA

Esta linguagem foi desenvolvida por Brendan Eich, da empresa Netscape, com o nome original de Mocha, tendo sido, mais tarde, renomeada para JavaScript.

A alteração do nome coincidiu com a data em que a Netscape lançou uma nova versão do navegador com suporte Java, em dezembro de 1995. Esta alteração causou confusão e sugeriu que o JavaScript era uma extensão da linguagem Java (que não é) – acredita-se que foi uma estratégia da empresa para obter prestígio.

A Microsoft criou um dialeto paralelo ao JavaScript para o seu navegador, Internet Explorer, que chamou de JScript para evitar problemas legais. Este foi adotado pela primeira vez no IE 3.0 em 1996, incluindo compatibilidade com o efeito 2000. O JavaScript e o Jscript eram semelhantes em muitos aspetos, até na forma de desenvolver o código, mas também tinham muitas diferenças.

Para evitar estas incompatibilidades, o W3C (World Wide Web Consortium) encarregou-se de criar uma nova linguagem comum a todos os navegadores.

### 1.3. PARA QUE SERVE O JAVASCRIPT?

Graças ao facto de ser uma linguagem executada no lado do cliente e não no lado do servidor, o JavaScript permite criar aplicações web interativas que respondem rapidamente à interação do utilizador.

No entanto, o facto de o código ser executado no lado do cliente tem também uma série de desvantagens, tais como a necessidade de recarregar a página para aceder aos dados (resolvido com AJAX) ou o facto de o código ser visível (esta questão deve ser levada em consideração uma vez que não é possível implementar um sistema de segurança baseado em senhas com JavaScript, por exemplo).

Esta linguagem será muito útil para interagir com o utilizador, por exemplo, em formulários, no desenvolvimento de aplicações matemáticas, etc., e de uma forma mais poderosa usando o melhor da plataforma AJAX.



## 2. NOMENCLATURA E COMENTÁRIOS

---

### 2.1. NOMENCLATURA

O primeiro dado que deverá ter em mente é que o JavaScript é uma linguagem simples, ou seja, tal como o HTML, não precisa de nenhuma ferramenta específica para a criação de código. Poderá utilizar o Notepad++ ou qualquer outro editor de texto simples, embora aconselhemos a utilização do Sublime Text 3 e do Visual Studio Code.

Após aprender um pouco mais sobre as origens da linguagem, irá conhecer a sua nomenclatura. O JavaScript está integrado no código HTML e, como verá mais tarde, poderá criar arquivos externos, embora neste caso a nomenclatura não se altere.

Para criar um script dentro do código HTML, será entre:

```
<script> ... código ... </script>
```

Ao aceder a um website, o navegador passa e interpreta o script e, se não houver erros, continua a ler o resto da página web, mas, se der erro, podem dar-se duas situações:

- Poderá ser um loop infinito (que não pode ser encerrado), o que faz com que a página nunca carregue.
- Poderá ser um erro comum, sendo que a página continuará a carregar, embora o script nunca funcione.

No exemplo anterior, foram definidas as tags para inserir o código. As tags utilizadas são genéricas para qualquer linguagem de script. O mais correto é indicar a linguagem que irá utilizar, deste modo:

```
<script language="JavaScript"> ... código ... </script>
```

Este método é conhecido como programação embutida e tem este nome porque o código é criado dentro do código HTML. Existe outra maneira de criar código JavaScript, que é utilizar ficheiros .js externos. Neste método é criado um ficheiro .js, que irá conter as funções e todo o código JavaScript, e no HTML é inserida uma linha dentro do "cabeçalho" do documento que faz a chamada a este ficheiro. Uma vez feita a chamada, as funções podem ser invocadas e o navegador carregará o código como se estivesse na própria página.

A nomenclatura para fazer uma chamada para um arquivo JavaScript externo é a seguinte:

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

Este método de criar os scripts é um pouco mais lento, pois tem de se criar no ficheiro .js todas as funções de que se irá precisar para poderem ser chamadas a partir do HTML. Mas também tem a vantagem de deixar o código HTML muito mais limpo e, além disso, é altamente recomendado para técnicas de posicionamento na web.

## Exemplo 01

---

Este primeiro exemplo prático irá mostrar como carregar um ficheiro .js no código HTML.

Comece por abrir o bloco de notas (ou editor que desejar) e escreva o seguinte comando:

```
alert("Sou o primeiro Script JS.")
```

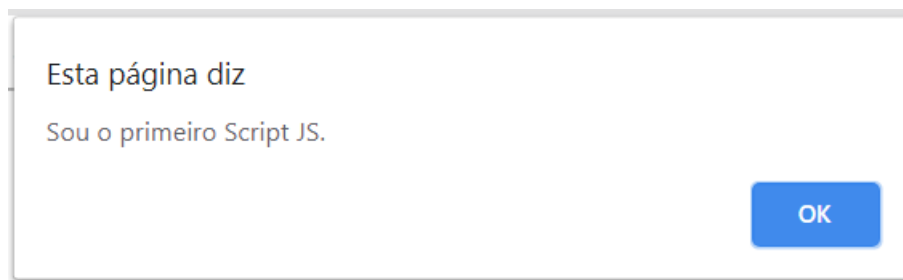
E guarde o ficheiro como "codigo1.js".

Num segundo ficheiro, que será o código HTML, escreva o seguinte código:

```
<html>
<head>
  <script type="text/javascript" src="codigo1.js"></script>
</head>
<body>
</body>
</html>
```

Este ficheiro foi guardado como "pagina1.html".

Ao abrir a "pagina1.html" com o navegador, irá abrir-se uma caixa de alerta com o texto "Sou o primeiro Script JS.", que foi colocado no alert (uma função predefinida pelo JavaScript para exibir mensagens), na primeira parte do exemplo.



Resultado do exemplo 01.

## 2.2. COMENTÁRIOS

Os comentários são pedaços de texto que o navegador não interpreta como código, e são muito úteis para os programadores manterem o código organizado e também esclarecerem algumas funções no código.

Para criar comentários, basta escrever o texto entre barras e asteriscos, como no exemplo abaixo:

```
/* sou um comentário */
```

Também é possível criar comentários colocando uma barra dupla em frente do comentário, mas isso só comentará essa frase ou linha:

```
// sou um comentário
```

Os comentários podem aparecer a qualquer altura no código JavaScript, desde que sempre dentro das tags `<script>` do código HTML; no caso de um ficheiro `.js` externo, como não é necessário escrever as tags `<script>`, é possível comentar livremente ao longo do código.

Em algumas páginas web, podem observar-se códigos JavaScript definidos entre as seguintes tags:

```
<!-- ... código ... -->
```

Este código é definido assim para evitar que alguns navegadores mais antigos que não interpretam JavaScript (apesar de obsoleto, ainda está presente em alguns sites) exibam o código JavaScript no HTML. O que este código faz é colocar todo o script dentro de um comentário HTML, deste modo, se o navegador não for capaz de interpretar o script, não mostrará o "alerta" escrito na página, pois está dentro de um comentário.

## 3. CONCEITOS BÁSICOS

---

Neste ponto, abordaremos algumas regras especiais da linguagem JavaScript que afetam a forma como o código é programado.

### 3.1. ESPAÇOS EM BRANCO

A linguagem JavaScript é interpretada de forma linear, ou seja, é lida consecutivamente linha por linha; se encontrar uma linha em branco, o JavaScript ignora-a e passa para a próxima, o que permite usar linhas e espaços em branco de modo a deixar o código mais organizado e limpo.

### 3.2. USO DE PLICAS

Todos os textos usados no código JavaScript, tanto para fornecer valores quanto para se referir a elementos, são colocados entre plicas. Em JavaScript pode-se usar aspas ou plicas, mas deverá ter sempre em mente que elas servem apenas como caracteres de abertura ou fecho de texto. Portanto, se quiser mostrar uma mensagem que contenha aspas ou plicas dentro dela, deverá usar o '\ ' antes das plicas para que funcionem. Por exemplo:

```
alert ('Avisa o primo 'Mario' para esperar')
```

Este código dará um erro; o código correto seria:

```
alert ('Avisa o primo \'mario\' para esperar')
```

### 3.3. MODOS DE EXECUTAR JAVASCRIPT

Existem quatro formas de executar código JavaScript numa página web em HTML:

1. Nos scripts, dentro dessas tags HTML, podemos colocar o código JavaScript para ser executado quando o navegador carregar essas linhas.

```
<script language="javascript">alert('Olá');</script>
```

2. Carregando em ficheiros externos que serão chamados a partir do código HTML. Estes ficheiros serão executados quando o navegador passar pela linha onde foi inserido o ficheiro externo.

```
<script type="text/javascript" src="codigo1.js"></script>
```

3. Após a ativação de algum evento HTML, como, por exemplo, mouseover, onload, etc.

```
<body onload="alert('Olá')"></body>
```

4. Substituindo um link por um comando de JavaScript.

```
<a href="Javascript:alert('Olá')">
```

São utilizados mais frequentemente os eventos, pois permitem interagir com o utilizador no nosso website. Segue-se uma lista dos **eventos** mais usados entre HTML e JavaScript:

■ Eventos com o rato:

- onMouseUP: inicia quando o botão esquerdo do rato é solto.
- onMouseDown: iniciado quando o botão esquerdo do rato é pressionado.
- onClick: é iniciado ao clicar com o rato num elemento do HTML.
- onDbClick: é iniciado quando um elemento HTML é clicado duas vezes.
- onMouseOver: é iniciado ao mover o cursor sobre o elemento HTML.
- onMouseOut: é iniciado ao deixar de passar o cursor sobre o elemento HTML.
- onDragDrop: é lançado quando o elemento HTML é arrastado pela página web.
- onDragStart: é lançado quando o elemento HTML começa a ser arrastado.
- onDrag: é iniciado quando um item é arrastado pela página web ou por outra página web.
- onDragEnter: é lançado quando um elemento HTML é arrastado para um destino válido do mesmo navegador ou não.
- onDragOver: tal como o anterior, mas inicia posteriormente.
- onDragLeave: igual ao anterior, mas sem ser eliminado pelo elemento HTML.
- onDragEnd: é iniciado quando o elemento HTML é solto após arrastar.
- onDrop: é iniciado quando o botão do rato é solto após uma operação de arrastar um elemento.



■ Eventos de teclado:

- `onKeyPress`: é iniciado quando o utilizador pressiona e para de pressionar uma tecla ou quando a mantém pressionada.
- `onKeyDown`: é lançado quando uma tecla é pressionada.
- `onKeyUp`: é lançado quando paramos de pressionar uma tecla.
- `onHelp`: é iniciado ao pressionar a tecla F1.

■ Eventos de formulário e modificações:

- `onChange`: é lançado quando o conteúdo do campo do formulário é alterado.
- `onPropertyChange`: é iniciado quando a propriedade de um elemento HTML muda.
- `onReadyStateChange`: é lançado quando o estado de um elemento muda (muito utilizado em AJAX, como poderá verificar mais adiante).
- `onSubmit`: é lançado quando pressionamos um botão de envio do formulário.
- `onReset`: é lançado quando pressionamos o botão vazio do formulário.

■ Outros eventos:

- `onFocus`: é lançado quando o cursor entra no elemento HTML (muito utilizado em campos de formulários).
- `onBlur`: é lançado quando o cursor (via teclado ou rato) sai de um elemento HTML.
- `onSelect`: é lançado quando o texto é selecionado dentro de uma caixa de texto, tanto com o rato quanto com o teclado.
- `onLoad`: é iniciado quando a janela ou uma imagem termina de carregar.
- `onUnload`: o oposto do anterior; é lançado quando uma janela é deixada para carregar outra ou também para recarregar a mesma.
- `onResize`: é lançado quando o utilizador altera o tamanho da janela, tornando-a maior ou menor.
- `onScroll`: é lançado ao deslizar com o rato ou teclado pela janela ou qualquer elemento que contenha scroll.

## 3.4. CARACTERES ESPECIAIS

Ao usar textos em JavaScript, há vários caracteres que não são escritos como faríamos num texto normal e que são úteis para formatar o texto. Em seguida, veremos estes caracteres especiais:

CARÁCTER	NOMENCLATURA
Quebra de linha	\r
Tabulador	\t
Nova linha	\n
Barra	\\
Plicas	\'
Aspas	\"

## 3.5. LINHAS DE CÓDIGO

O último elemento a considerar em JavaScript é como criar as linhas de código, embora não seja necessário, já que, na maioria dos navegadores, funciona simplesmente escrevendo a frase. Deverá terminar cada linha de função com ';', ou seja, cada linha que indica um evento de programação deve terminar com ';', enquanto as instruções de controlo (if, else, for ...) não carregam este ';'.

## CONCLUSÃO

---

O JavaScript é uma linguagem que nasceu da necessidade de uma maior interação entre a página web e o utilizador.

É uma ferramenta muito poderosa e versátil, pois permite-nos controlar eventos HTML e gerar scripts que são executados com estes mesmos eventos.

Nesta unidade didática, extremamente importante para a sua aprendizagem, vimos como, onde e por que razão podemos executar o código JavaScript da forma mais eficiente possível.



## AUTOAVALIAÇÃO

---

- 1. Em que se baseia a linguagem JavaScript?**
  - a) Tags.
  - b) Scripts de execução no servidor.
  - c) Scripts de execução no cliente.
  - d) Funções.
  
- 2. Originalmente, a linguagem JavaScript foi desenvolvida:**
  - a) Como linguagem de programação web para interagir com o utilizador.
  - b) Como um módulo de implementação Java na web.
  - c) Como uma linguagem independente para criar sites somente com ela.
  - d) Como uma linguagem concorrente ao Java.
  
- 3. Em que ano o JavaScript foi criado pela Netsape?**
  - a) 1970.
  - b) 1980.
  - c) 1990.
  - d) 1995.

**4. Quais são as tags HTML para inserir código JavaScript?**

- a) `<javascript> </javascript>`.
- b) `<script> </script>`.
- c) `<jscript> </jscript>`.
- d) `<scriptJS> </scriptJS>`.

**5. Como se pode comentar várias linhas em JavaScript?**

- a) `\\ ....`
- b) `\* ... *\.`
- c) `<!-- ... -->`.
- d) Não se pode comentar código.

**6. Em qual dos métodos não é possível inserir scripts?**

- a) Em links.
- b) Em eventos HTML.
- c) Dentro das tags `<javascript></javascript>`.
- d) Em ficheiros externos .js.

**7. Quando é que o evento onload é iniciado?**

- a) Quando a web começa a carregar.
- b) Ao pressionar o botão "Enviar" de um formulário.
- c) Ao carregar um ficheiro .js externo.
- d) Quando a página web termina de carregar.

8. Se quiser iniciar um script quando uma tecla for pressionada e solta, qual é o evento correto?
- a) `onKeyUp`.
  - b) `onKeyPress`.
  - c) `onKeyDown`.
  - d) `onClick`.
9. Quando é iniciado o evento `onMouseOver`?
- a) Ao passar com o rato sobre um elemento HTML.
  - b) Ao clicar com o rato num elemento HTML.
  - c) Ao sair de um elemento HTML com o rato.
  - d) Ao arrastar um elemento HTML.
10. Se quiser exibir uma mensagem com duas linhas de texto, que carácter especial de quebra de linha é utilizado?
- a) `\br`.
  - b) `\sl`.
  - c) `\r`.
  - d) `\t`.





## SOLUÇÕES

---

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

## PROPOSTAS DE DESENVOLVIMENTO DO ESTUDO

---

Para aprofundar a história do JavaScript:

- <http://pt.wikipedia.org/wiki/JavaScript>

Para ver todos os eventos possíveis que podem ser gerados a partir de HTML:

- [https://www.w3schools.com/js/js\\_events.asp](https://www.w3schools.com/js/js_events.asp)

## BIBLIOGRAFIA

---

- VV. AA. (2010). *JavaScript*. Madrid: Anaya Multimedia.

