Para usar ECMAScript 5, o ficheiro de JavaScript deve começar com 'use strict'.

- Alterações:
  - Não existem variáveis globais não declaradas.
  - Não se declaram variáveis com var.
  - Alguns warnings passaram a erros.

Variáveis são declaradas com let. Os seus nomes podem ter apenas números, letras, $ e _ (e não podem começar com um número).

Constantes são declaradas com const.

Variáveis declaradas com var:

- Não têm block scope (só function scope).
- São processadas quando uma função começa.

Tipos de dados primitivos:

- Number (double).
- String (text).
- Boolean (true ou false).
- Null (apenas 1 valor: null).
- Undefined (ainda não foi atribuído um valor).

Strings podem ser definidas com plicas, aspas ou backticks (‘). Com a última, expressões dentro de ${...} são avaliadas e o resultado faz parte da string.

Operador + soma números ou concatena strings (se pelo menos um operando for uma string).

É possível usar String(), Boolean() e Number() para converter valores para os tipos pretendidos.

=== e !== comparam valores. == e != comparam tipos.

Funções são definidas com a keyword function. Tipos primitivos são passados por valor, mas tipos não primitivos são passados por referência. Funções com um return vazio ou sem return, retornam undefined.

Arrow functions são uma forma mais compacta de declarar funções.

Pode-se usar a keyword this dentro de um objeto para referir ao objeto atual.

Call e Apply são alternativas a chamadas a funções. Ambas recebem o contexto como primeiro argumento.

Objetos podem ser acedidos com [] como num array associativo.

for ... in permite executar código para cada propriedade de um objeto/array.

Cada função tem uma propriedade prototype interna que é inicializada como um objeto vazio. Quando o new é utilizado, é criado um novo objeto derivado do prototype. É possível alterar o prototype de uma função deretamente.

Quando um objeto é criado com new, uma propriedade __proto__ é inicializada com o prototype da função que o criou.

A keyword class é uma forma simplificada de criar classes protoype-based. Só podem ter métodos e getters/setters.

Arrays são objetos list-like. Só podem ser acedidos com []. Os seus índices começam em 0.

Podem ser lançadas exceções (ou qualquer outro objeto) com throw.

Exceções devem ser envolvidas em blocos try .. catch.

JavaScript pode ser incluído num ficheiro HTML com a tag script.

Scripts devem ter 1 dos seguintes métodos:

- async: é corrido assim que seja feito o seu download sem bloquear o browser.
- defer: é corrido apenas quando a página é carregada e por ordem.

document representa o documento atual. Contém métodos importantes para aceder a elementos. Um objeto Element representa um elemento HTML (id, innerHTML, outerHTML, style, getAttribute(), setAttribute(), remove()). A função createElement() de document cria um novo elemento, mas não é inserido no documento. O objeto Node representa um nó na árvore do documento (appendChild(), replaceChild(), removeChild(), insertBefore()).

XMLHttpRequest permite enviar pedidos HTTP facilmenteL

- open(method, url, async)
  - Method: get ou post.
  - URL: URL a procurar.
  - Async: se for falso, a execução para à espera de uma resposta.
- encodeForAjax(obj): cria um objeto a passar ao PHP.
- send(): envia os dados.
- JSON.parse(): permite receber a resposta do servidor.