SISTEMAS DISTRIBUÍDOS E TOLERÂNCIA A FALHAS

2011

José Castanheira m3852

Óscar Pinto m4360

Ajax - Asynchronous Javascript And XML

Esta apresentação tem por objectivo dar a conhecer de uma forma geral, o Ajax – Asynchronous Javascript And XML.

Deste modo serão mostradas as diferentes tecnologias que servem de suporte ao mesmo.

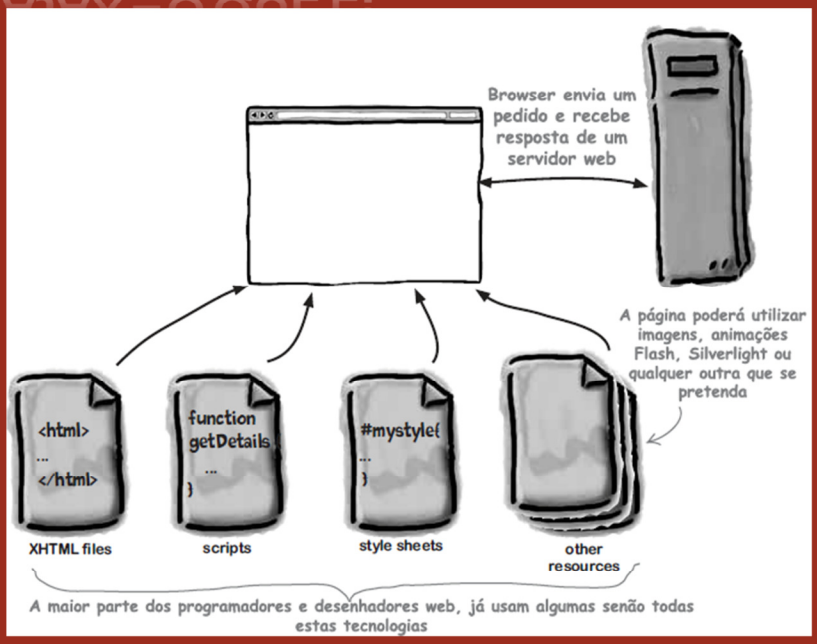
AJAX – O QUE É?

- não é uma nova linguagem de programação
- nem uma tecnologia em si
- também não é uma *framework* ou uma API
- digamos que é apenas uma nova maneira de pensar.

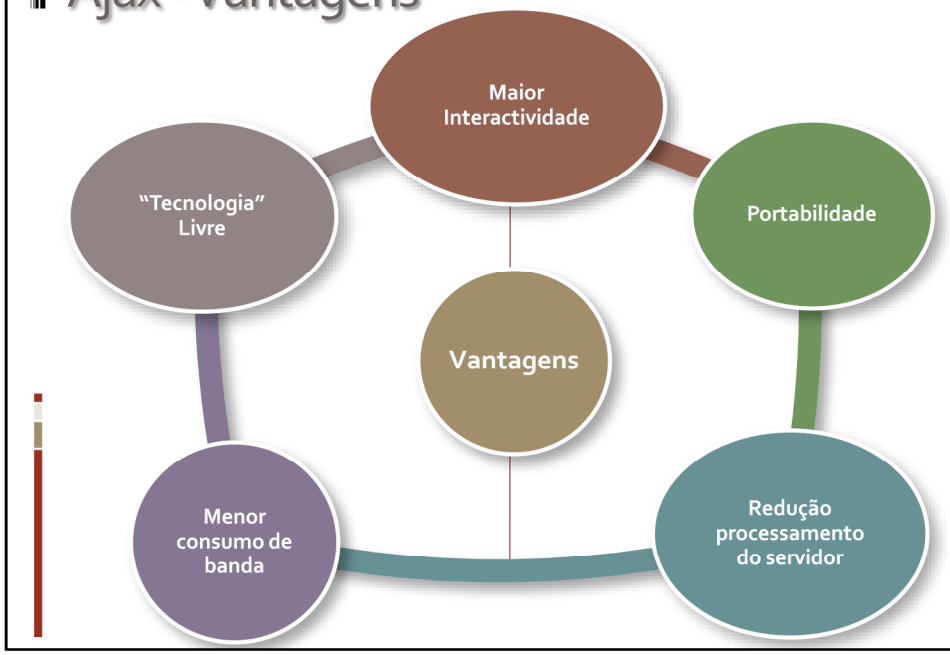
No fundo o Ajax é uma funcionalidade implementada por um conjunto de objectos JavaScript, sendo o mais importante o XMLHttpRequest.

- **De uma maneira generalista, podemos dizer que a principal função do Ajax é a arte de trocar dados com um servidor e actualizar partes de uma página da web, sem ser necessário recarregar a página inteira.**

AJAX – O QUE É?


Browser envia um pedido e recebe resposta de um servidor web
A página poderá utilizar imagens, animações Flash, Silverlight ou qualquer outra que se pretenda
<html>
...
</html>
function getDetails
...
}
#mystyle{
...
}
XHTML files
scripts
style sheets
other resources
A maior parte dos programadores e desenhadores web, já usam algumas senão todas estas tecnologias

Ajax - Vantagens



AJAX – TECNOLOGIA

O Ajax é formado por um conjunto de várias tecnologias:

- Objecto capaz de fazer requisições assíncronas;
- HTML;
- JavaScript;
- XML eXtensible Markup Language;
- Json JavaScript Object Notation;
- DHTML;
- CSS.

AJAX – TECNOLOGIA

Pré-Requisitos para trabalhar com AJAX:

-HTML - *HyperText Markup Language*

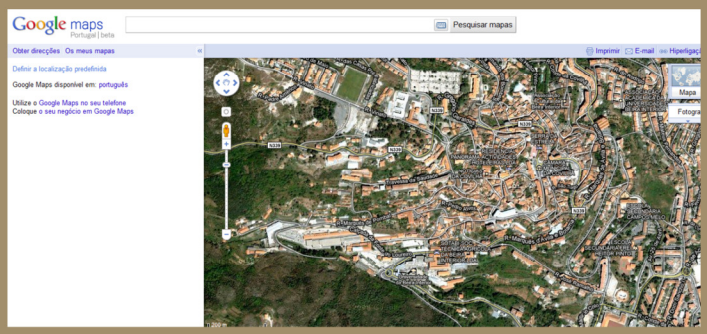
- CSS - *Cascading Style Sheets* é uma linguagem de estilo utilizada para definir a apresentação de documentos escritos numa linguagem de marcação, como o HTML ou XML. O seu principal benefício é podermos a separar o formato e o conteúdo de um documento.

-Javascript - é uma linguagem de programação baseada na linguagem de programação ECMAScript e é atualmente a principal linguagem para programação client-side em web browsers. Foi concebida para ser uma linguagem script orientada a objectos baseada em protótipos.

AJAX – ONDE ENCONTRAR…

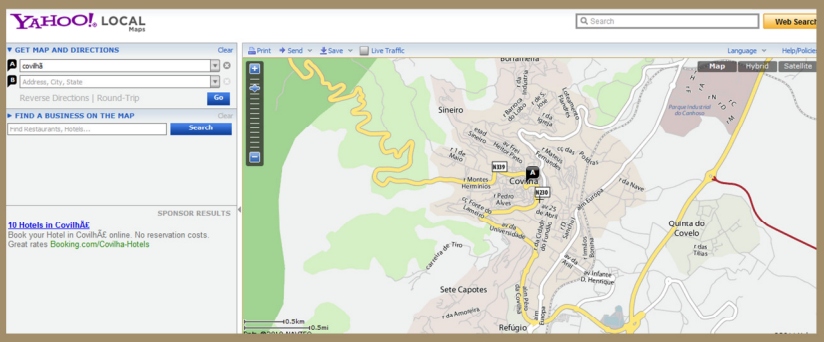
Um utilizador da internet muitas vezes é levado a crer que um certo site/aplicação Web funciona com Ajax, quando são outros processos que se passam por detrás.

- a Google utiliza o Ajax no seu serviço de email, o Gmail , e no de mapas, Google Maps.



AJAX – ONDE ENCONTRAR…

O sistema de mapas do Yahoo, funcionou durante bastante tempo através da utilização de Flash, ao invés do Ajax.
Apesar disso, o seu funcionamento e visualização era em tudo idêntico ao do Google Maps que sempre funcionou com Ajax.



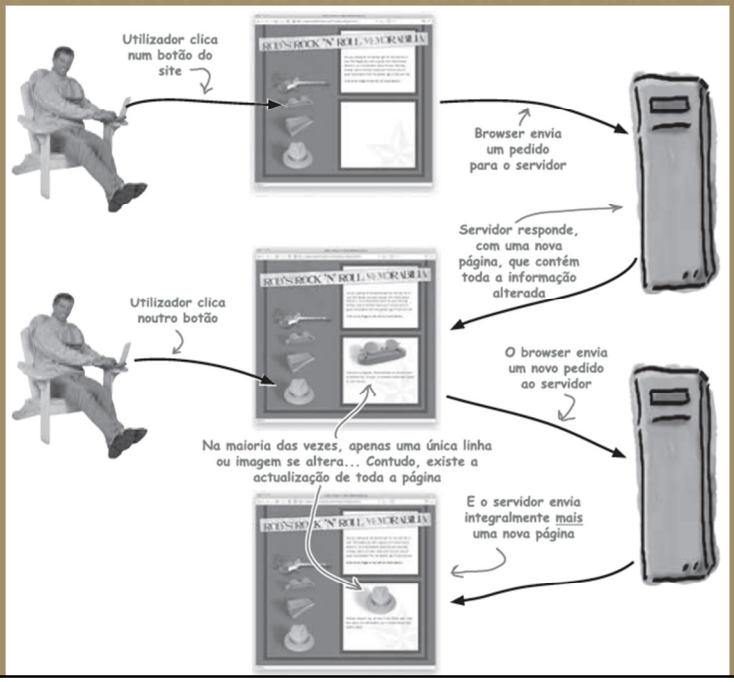
AJAX – COMUNICAÇÃO ASSÍNCRONA

- Enquanto a maior parte das aplicações clássicas seguem uma sequência de troca de informações totalmente síncrona, **o AJAX por defeito tem uma sequência assíncrona**, porém esta pode ser configurada para ser síncrona.

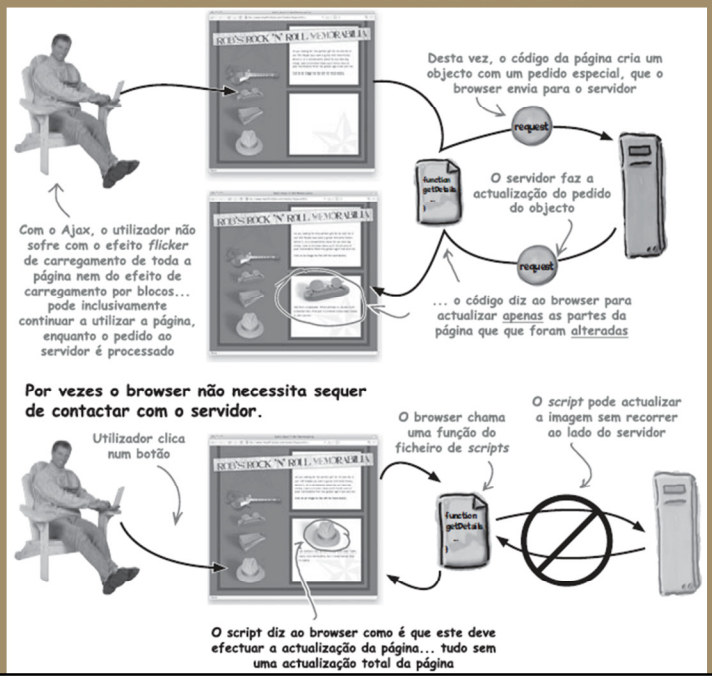
- **Comunicação síncrona**: ao existir uma pedido ao servidor, o utilizador fica à espera de resposta do mesmo. Quando chega a resposta , toda a página é carregada/recarregada e o conteúdo é exibido. Este método pode ser útil nas situações em que um utilizador tem mesmo de esperar por uma validação de alguma operação.

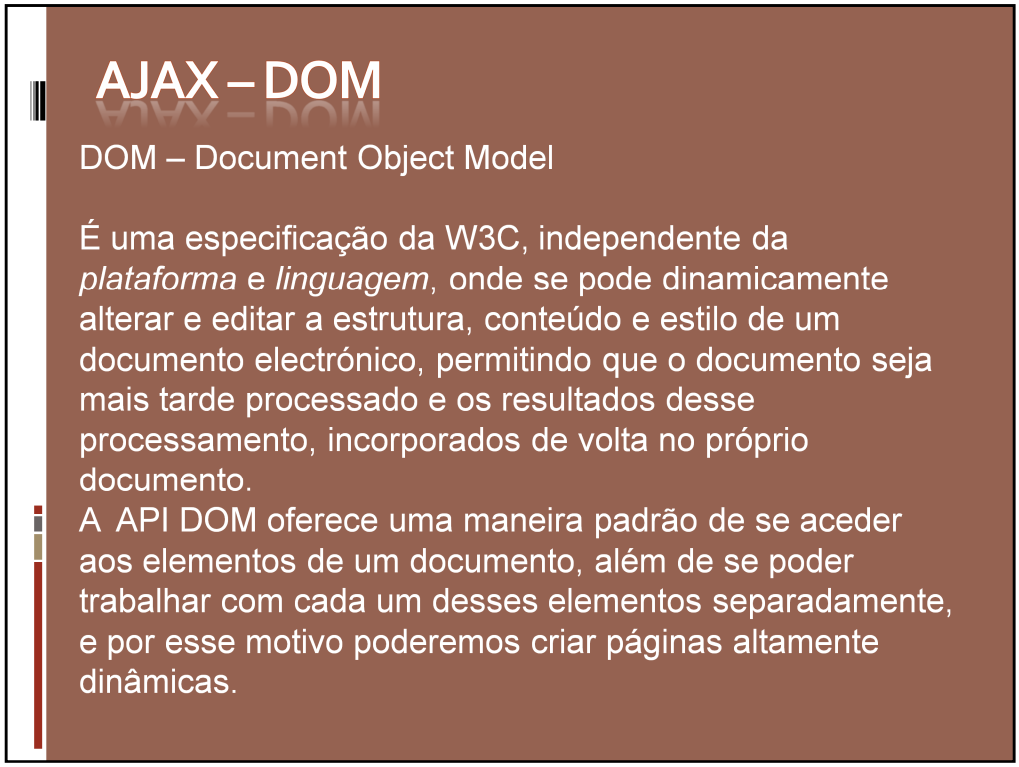
- **Comunicação assíncrona**: ao existir um pedido ao servidor, tudo é processado para que o utilizador do sistema não tome consciência disso. Assim, o utilizador continua a trabalhar no sistema e após receber o conteúdo utiliza-se Javascript para processar e exibir o resultado no ecrã.

UTILIZAÇÃO TRADICIONAL SEM AJAX



AJAX – UTILIZAÇÃO COM AJAX





O **DOM** fornece a representação estrutural de documentos HTML e XML, definindo a forma como a estrutura pode ser acedida por programas e scripts, possibilitando a sua modificação do estilo e do conteúdo do documento.

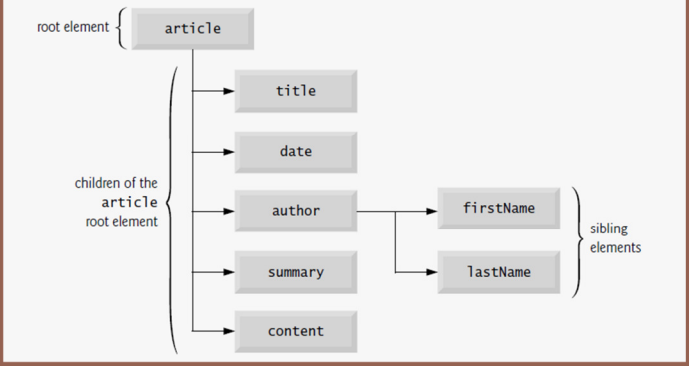
O DOM não é uma linguagem. Apenas concede a estrutura de um documento e seus elementos.

Então para trabalhar, pode-se utilizar a conjugação do DOM com o JavaScript. Desta forma, teremos acesso à estrutura, estilo e conteúdo de um documento através do DOM e com o JavaScript poderemos manipulá-los.

O DOM foi desenvolvido para ser independente de qualquer linguagem de programação, o que é importante porque diversas linguagens utilizam-no para ter acesso a determinado documento.

AJAX – DOM

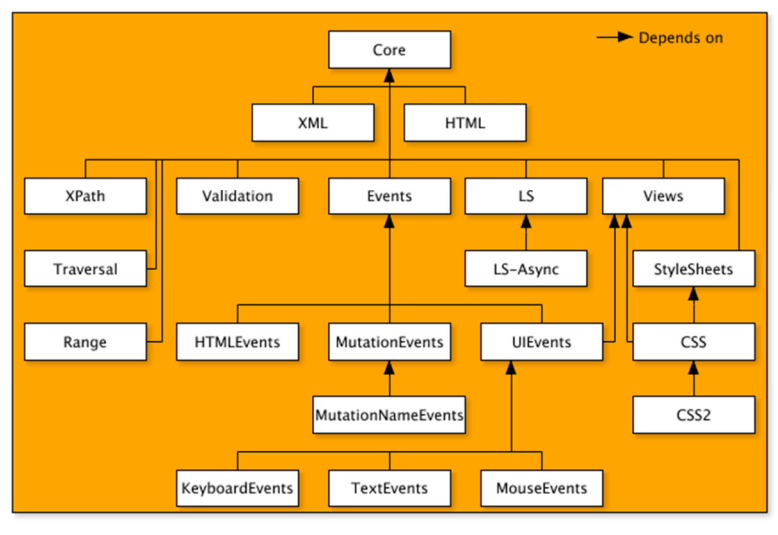
A representação do documento ocorre através de uma árvore de nós (tree of node) em que cada objecto possui propriedade e método.



No exemplo acima temos a estrutura de um documento XML que foi representado pelo DOM. Podemos encará-lo como uma árvore genealógica.

AJAX – DOM

A seguinte representação contém todos os módulos de DOM:



AJAX – JAVASCRIPT E DOM

Conjugando Javascript com DOM, podemos então manipular todos os objectos da nossa página Web.

The most common DOM methods at a glance

Reaching Elements in a Document

`document.getElementById('id')`: Retrieves the element with the given `id` as an object

`document.getElementsByTagName('tagname')`: Retrieves all elements with the tag name `tagname` and stores them in an array-like list

Reading Element Attributes, Node Values and Other Data

`node.getAttribute('attribute')`: Retrieves the value of the attribute with the name `attribute`

`node.setAttribute('attribute', 'value')`: Sets the value of the attribute with the name `attribute` to `value`

`node.nodeType`: Reads the type of the `node` (1 = element, 3 = text node)

`node.nodeName`: Reads the name of the `node` (either element name or #textNode)

`node.nodeValue`: Reads or sets the value of the `node` (the text content in the case of text nodes)

Navigating Between Nodes

`node.previousSibling`: Retrieves the previous sibling node and stores it as an object.

`node.nextSibling`: Retrieves the next sibling node and stores it as an object.

`node.childNodes`: Retrieves all child nodes of the object and stores them in an list. here are shortcuts for the first and last child node, named `node.firstChild` and `node.lastChild`.

`node.parentNode`: Retrieves the node containing `node`.

Creating New Nodes

`document.createElement(element)`: Creates a new element node with the name `element`. You provide the name as a string.

`document.createTextNode(string)`: Creates a new text node with the node value of `string`.

`newNode = node.cloneNode(bool)`: Creates `newNode` as a copy (clone) of `node`. If `bool` is `true`, the clone includes clones of all the child nodes of the original.

`node.appendChild(newNode)`: Adds `newNode` as a new (last) child node to node.

`node.insertBefore(newNode,oldNode)`: Inserts `newNode` as a new child node of node before `oldNode`.

`node.removeChild(oldNode)`: Removes the child `oldNode` from node.

`node.replaceChild(newNode, oldNode)`: Replaces the child node `oldNode` of node with `newNode`.

`element.innerHTML`: Reads or writes the HTML content of the given element as a string—including all child nodes with their attributes and text content.

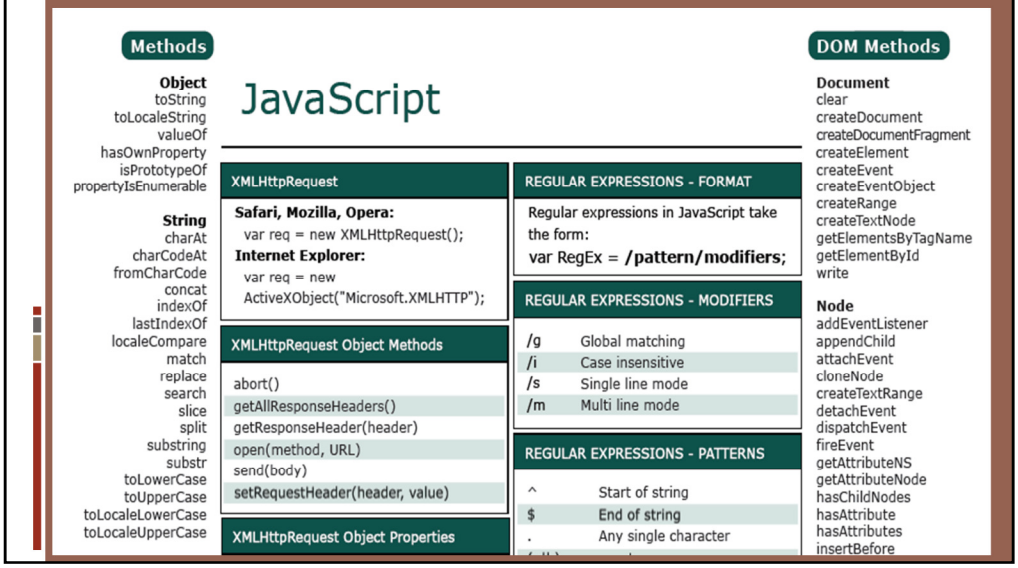
Known browser quirks:

`getAttribute` and `setAttribute` are not reliable. Instead, assign the property of the element object directly: `obj.property = value`. Furthermore, some attributes are actually reserved words, so instead of `class` use `className` and instead of `for` use `HTMLfor`.

Real DOM compliant browsers will return linebreaks as text nodes in the `childNodes` collection, make sure to either remove them or test for the `nodeType`.

AJAX – JAVASCRIPT E DOM

Exemplo de Métodos de Javascript:

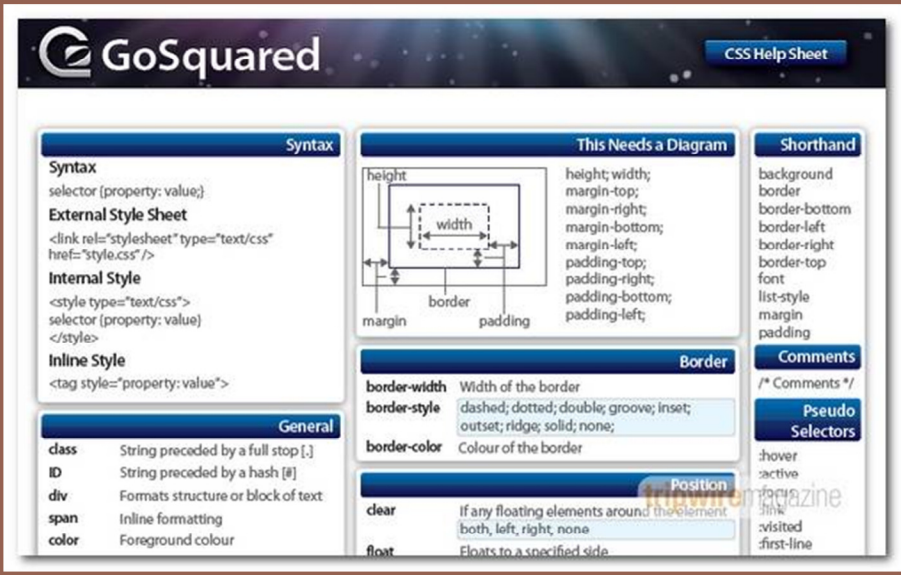


AJAX – CSS

CSS é outra parte importante para o conjunto de tecnologia que envolve o AJAX e tem uma sintaxe simples e utiliza:

- ✓ Uma série de palavras em inglês para especificar os nomes de diferentes estilos de propriedade de uma página. Uma folha de estilo consiste em uma lista de regras.
- ✓ Cada regra ou conjunto de regras consiste de um ou mais selector e um bloco de declaração. Um declaração de bloco é composta por uma lista de declarações entre chaves. Cada declaração em si é uma *propriedade*, dois pontos (:), um *valor*, então um ponto e vírgula (;).
- ✓ Os selectores podem ser aplicados a todos os elementos de um tipo específico, ou apenas aqueles elementos que correspondam a um determinado atributo; os elementos podem ser combinados, dependendo de como eles são colocados em relação uns aos outros no código de marcação, ou como eles estão aninhados dentro do objecto de documento modelo.

AJAX – CSS


GoSquared
CSS Help Sheet
Syntax
Syntax
selector {property: value;}
External Style Sheet
<link rel="stylesheet" type="text/css" href="style.css" />
Internal Style
<style type="text/css">
selector {property: value}
</style>
Inline Style
<tag style="property: value">
General
class String preceded by a full stop [.]
ID String preceded by a hash [#]
div Formats structure or block of text
span Inline formatting
color Foreground colour
This Needs a Diagram
height
width
border
margin
padding
height; width;
margin-top;
margin-right;
margin-bottom;
margin-left;
padding-top;
padding-right;
padding-bottom;
padding-left;
Border
border-width Width of the border
border-style dashed; dotted; double; groove; inset; outset; ridge; solid; none;
border-color Colour of the border
Position
clear If any floating elements around the element both, left, right, none
float Floats to a specified side
Shorthand
background
border
border-bottom
border-left
border-right
border-top
font
list-style
margin
padding
Comments
/* Comments */
Pseudo Selectors
:hover
:active
:focus
:link
:visited
:first-line

AJAX – XML

XML (**eXtensible Markup Language**) é :

- ✓ uma recomendação da W3C para gerar linguagens de marcação para necessidades especiais. Capaz de descrever diversos tipos de dados.
- ✓ A sua principal característica é criar uma infra-estrutura única para diversas linguagens.
- ✓ é um formato para a criação de documentos com dados organizados de forma hierárquica, como se vê, frequentemente, em documentos de texto formatados.
- ✓ torna-se numa forma de mostrar dados em formato de texto, que pode ser interpretado por linguagens de scripting , de forma a que os utilizadores também possam olhar para os dados de uma forma fácil.

A utilização do XMLHttpRequest (XHR) é relativamente simples.

Basta construir um objecto desse tipo, abrir um URL (ou um arquivo local no servidor) e enviar um pedido (como GET ou POST), com ou sem parâmetros.

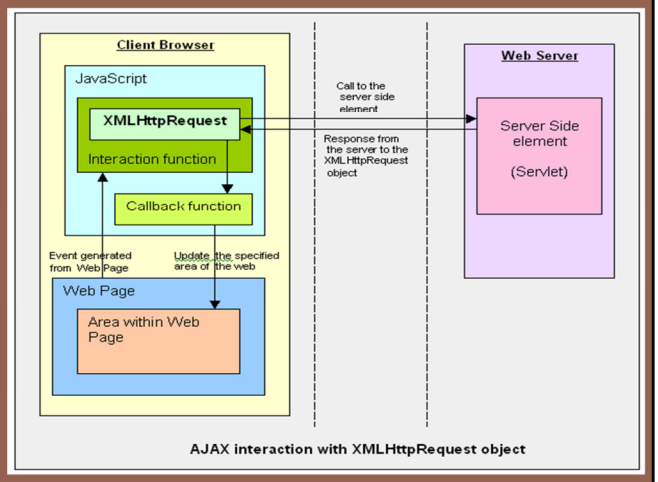
O código de status HTTP do pedido (resposta) e os dados (documento) ligados a ele estarão disponíveis através desta instância do objecto.

O objecto XHR dispõe de um método de chamada e de retorno, que permite que o navegador continue a funcionar normalmente até que o pedido enviado seja realizado e tratado.

AJAX – XMLHTTPREQUEST

XMLHttpRequest pode ser considerado um objecto Javascript que torna possível a comunicação assíncrona com o servidor, sem a necessidade de recarregar a página por completo.

O objecto XMLHttpRequest é hoje parte da especificação do DOM, nível 3.



A utilização do XMLHttpRequest (XHR) é relativamente simples.

Basta construir um objecto desse tipo, abrir um URL (ou um arquivo local no servidor) e enviar um pedido (como GET ou POST), com ou sem parâmetros.

O código de status HTTP do pedido (resposta) e os dados (documento) ligados a ele estarão disponíveis através desta instância do objecto.

O objecto XHR dispõe de um método de chamada e de retorno, que permite que o navegador continue a funcionar normalmente até que o pedido enviado seja realizado e tratado.

AJAX – XML HTTP REQUEST

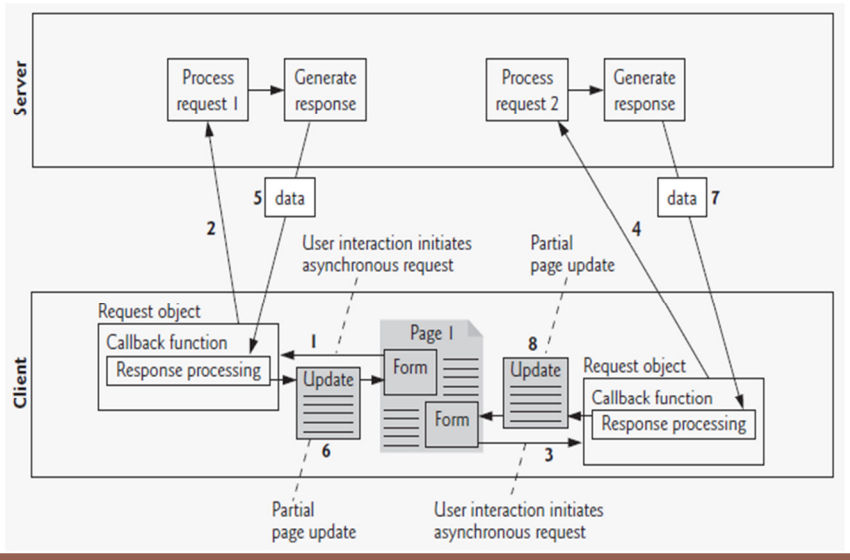
Propriedades do objeto XMLHttpRequest	
readyState	A requisição se apresenta em 4 (quatro) estágios; ambos representando por um número. • 0 - não inicializado; • 1 - carregamento; • 2 - carregado; • 3 - interativo; • 4 - completo.
status	Código numérico do status HTTP retornado pelo servidor web.
statusText	Texto associado ao código numérico do status HTTP. Por exemplo: 200 significa "OK" e 404 significa "Página não encontrada".
responseText	String que contém os dados retornados pelo servidor web.
responseXML	Se o servidor web retornar um documento XML, lhe permitindo acessá-lo através de funções JavaScript utilizando o DOM.

AJAX – XML HTTP REQUEST

Métodos do objeto XMLHttpRequest	
open(método, url, síncrono, usuário, senha)	Inicia uma nova requisição, onde: • método - Requisição HTTP, na maiorida das vezes "GET", ou "POST"; • url - endereço da URL que será requisitada no servidor web; • síncrono - se o método trabalhará de forma síncrona ou assíncrona; o valor padrão é true - assíncrona; • usuário e senha - se o servidor web necessitar de uma autenticação.
setRequestHeader(nome, valor)	Informa um cabeçalho (header) para a requisição.
send(dados)	Envia a requisição. Enviando opcionalmente os dados.
abort()	Aborta uma requisição em atividade.
getResponseHeader(nome)	Retorna o valor do cabeçalho (header) informado.
getAllResponseHeaders()	Retorna uma string contendo todos os cabeçalhos (header).

Eventos do objeto XMLHttpRequest	
onreadystatechange	Elevando a cada mudança da propriedade readyState.

AJAX – XML HTTP REQUEST



Quando o utilizador interage com a página, o cliente cria um XMLHttpRequest para gerir um pedido (Passo1)

O objecto XMLHttpRequest object envia o pedido ao servidor e espera a resposta (Passo 2)

O pedido é assincrono, como tal o utilizador pode continuar a enteragir com a aplicação no lado do cliente (client-side) enquanto o servidor processa o pedido de forma concorrencial.

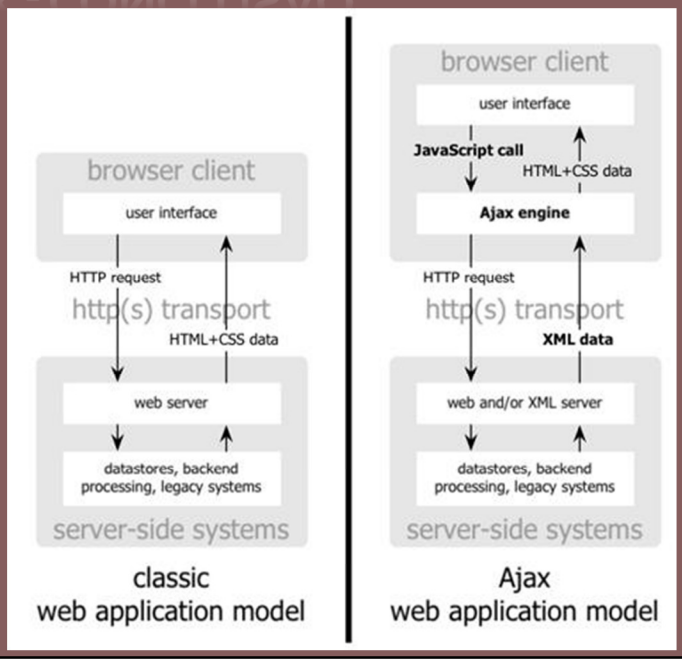
A interacção do utilizador pode resultar em pedidos adicionais ao servidor (Passo 3 e 4).

Uma vez processada a resposta do servidor relativa ao pedido original (Passo 5), o objecto XMLHttpRequest que fez o pedido chama a função do lado do cliente (client-side) para processar os dados devolvidos pelo servidor.

Esta função —conhecida como callback function— actualiza certas partes da página (Passo 6) para mostrar os dados na página actual, sem haver necessidade de actualizar a página na totalidade.

Ao mesmo tempo o servidor pode ir respondendo aos outros pedidos (Passo 7) e assim sempre com este ciclo (Passo 8) em que a callback functio apenas actualiza a respectiva parte da página. Este tipo de procedimento torna as aplicações mais próximas de aplicações de desktop.

AJAX - CONCLUSÃO



Questões?

Obrigado pela atenção

FIM