

Tecnologias Web

6ª AULA, 18 de março

WWW

Roberto Lam, Instituto Superior de Engenharia,
Universidade do Algarve

rlam@ualg.pt

2023/24



JavaScript



Clientes HTTP (browsers)

BOM – Browser Object Model

Comportamentos distintos nos clientes HTTP (API *browsers*)



Javascript, Client-side scripting?

Servidores específicos em HTTP (NODE.js, cloud computing)

Lenny Bourdette, The JavaScript Pocket Guide 2010, Peachpit Press

JavaScript



Window (atributos)

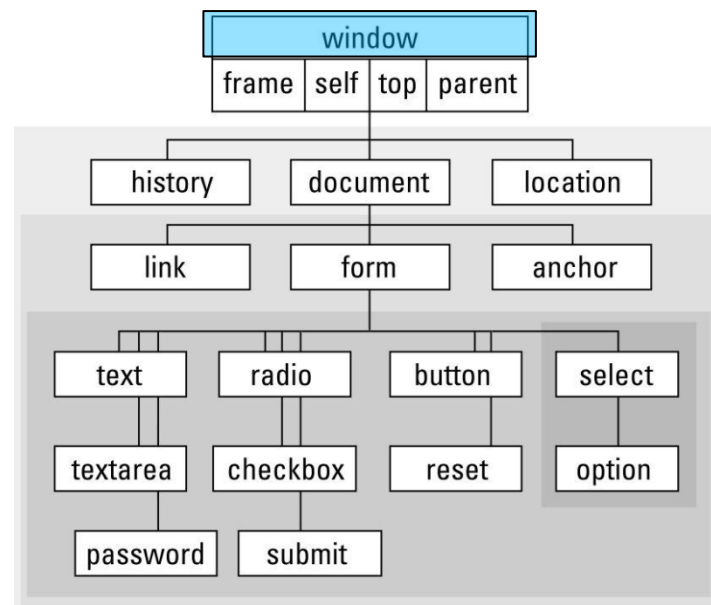
innerHeight, innerWidth

outerHeight, outerWidth

pageXOffset, pageYOffset

screenLeft, screenTop,
screenX, screenY

`window.innerHeight=150;`



consultar: http://www.w3schools.com/jsref/obj_window.asp

<http://vkanakaraj.wordpress.com/2009/12/18/javascript-vs-dom-vs-bom-relationship-explained/>

JavaScript



window (métodos)

open(), close(), resizeTo(), moveTo() ,scrollTo()

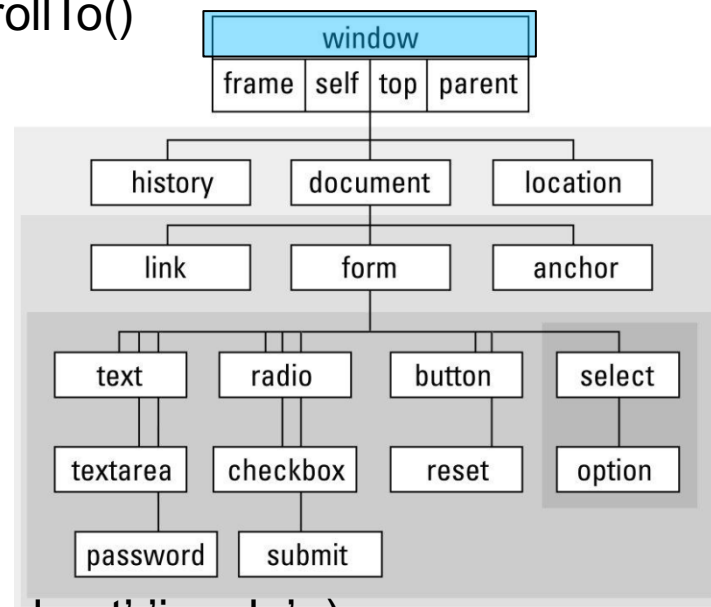
resizeBy(), moveBy(), scrollBy()

focus(), blur(), print()

alert()

confirm()

prompt()



```
var wind=window.open('http://www.ualg.pt','janela', );
```

JavaScript



history (atributo e métodos)

length

window.length=0;

go(), back(), forward()

window.history.go(-4);

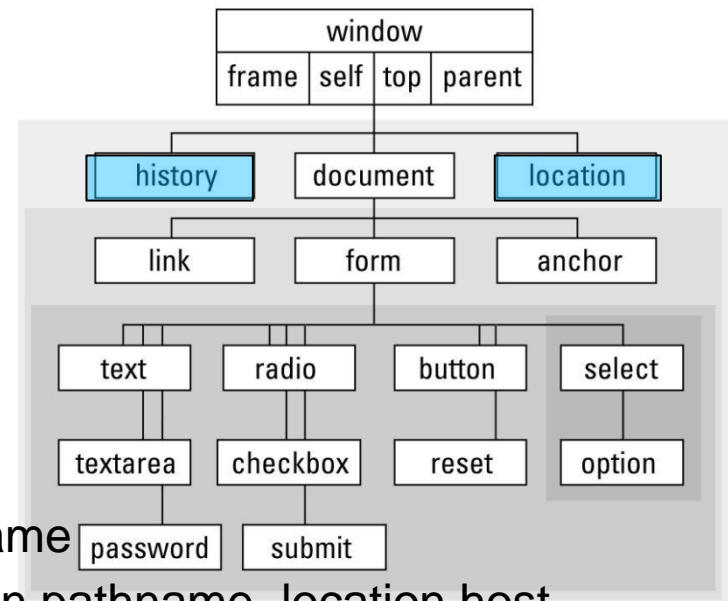
window.history.forward();

location (atributos, metodos)

location, location.href, location.pathname

location.port, location.protocol, location.pathname, location.host

location.assign(), location.replace(), location.reload()



<http://www.example.com:8080/dir/script.php?param=value¶m2=value2> ?????

JavaScript



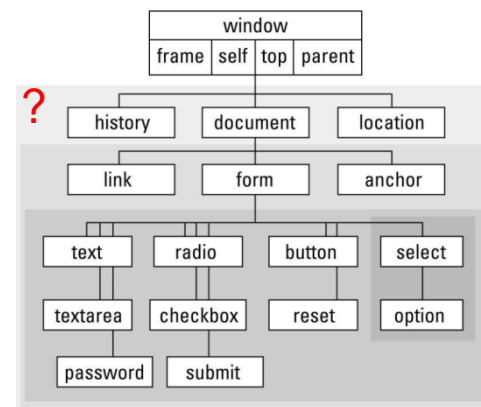
navigator (atributo e métodos)

Navigator Object Properties

| Property | Description |
|-------------------------------|--|
| appName | Returns the code name of the browser |
| appVersion | Returns the name of the browser |
| cookieEnabled | Returns the version information of the browser |
| onLine | Determines whether cookies are enabled in the browser |
| platform | Boolean, returns <i>true</i> if the browser is on line, otherwise <i>false</i> . |
| userAgent | Returns for which platform the browser is compiled |
| userAgent | Returns the user-agent header sent by the browser to the server |

Navigator Object Methods

| Method | Description |
|--------------------------------|--|
| javaEnabled() | Specifies whether or not the browser has Java enabled |
| taintEnabled() | Specifies whether or not the browser has data tainting enabled |

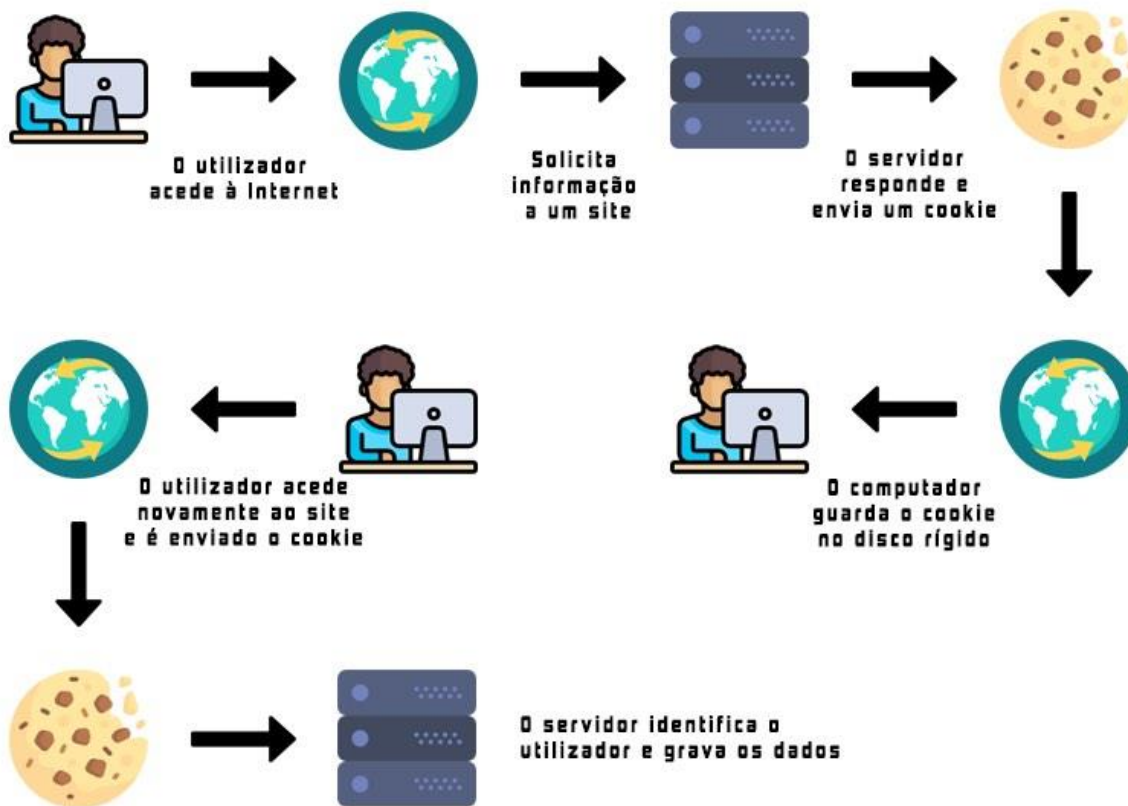


consultar: https://www.w3schools.com/jsref/obj_window.asp

JavaScript



Cookies



JavaScript



Cookies



JavaScript



Cookies (Domain e Path)

Domain, Path, Expires, Max-Age, Secure e HttpOnly

```
Set-Cookie: LSID=DQAAAK...Eaem_vYg; Domain=docs.foo.com; Path=/accounts; Expires=Wed, 13-Jan-2021 22:23:01 GMT; Secure; HttpOnly
Set-Cookie: HSID=AYQEVN....DKrdst; Domain=.foo.com; Path=/; Expires=Wed, 13-Jan-2021 22:23:01 GMT; HttpOnly
Set-Cookie: SSID=Ap4P....GTEq; Domain=.foo.com; Path=/; Expires=Wed, 13-Jan-2021 22:23:01 GMT; Secure; HttpOnly
.....
```

Cookies JavaScript

Definição

```
document.cookie = "Nome=Valor";
```

```
document.cookie = "outroNome=outroValor";
```

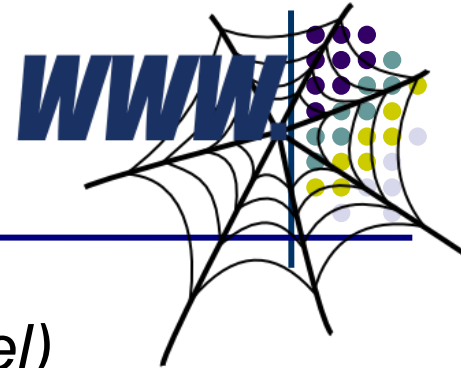
Acesso

```
var cookiesString=document.cookie;
```

cookiesString contém: "Nome=Valor; outroNome=outroValor "

<http://www.thesitewizard.com/javascripts/cookies.shtml>

<http://www.troyhunt.com/2013/03/c-is-for-cookie-h-is-for-hacker.html>



DOM – HTML, XML (Document Object Model)

XML (eXtensible Markup Language)

O Document Object Model (DOM) é um interface de programação (API) para documentos HTML e XML. Ele fornece uma representação estruturada do documento e define o modo como a estrutura pode ser acessada, a partir de programas, para que seja possível alterar a estrutura do documento, estilo e conteúdo.

O DOM fornece uma representação do documento como um conjunto estruturado de nós e objectos que têm propriedades e métodos.



DOM Level1, DOM Level2 e DOM Level3

document é o objecto (nó, elemento) raiz de um documento HTML ou XML (DOM)

Características básicas dos nós (nodes)

nodeName é o nome do nó, exemplos: #document, #text

nodeType é o numero que descreve o tipo de nó: 1 para tags *HTML*, 3 nós texto e 9 para o nó document

nodeValue é o texto contido num nó texto.

innerHTML é o *HTML* contido num nó.

id é o valor do atributo ID do nó.

classname é o valor do atributo classe do nó.

https://developer.mozilla.org/en-US/docs/DOM_Levels

<http://www.w3.org/TR/REC-DOM-Level-1/>

https://developer.mozilla.org/en-US/docs/Gecko_DOM_Reference/Examples



Relações entre nós (nodes)

firstChild é o primeiro nó filho do corrente nó.

lastChild é o último nó filho do corrente nó.

childNodes é um array com todos filhos do corrente nó.

previousSibling é o nó irmão anterior ao corrente nó.

nextSibling é o nó irmão seguinte ao corrente nó.

parentNode é o nó pai (objecto) do corrente nó.

Característica Style dos nós (nodes)

O nó **style** sob cada objecto DOM contém os atributos definidos pelos style sheet properties. Definidos pela marca <style>.,



Métodos dos nós (nodes)

appendChild(node) adiciona um nó filho depois de todos filhos existentes.

insertBefore(node,oldnode) insere um nó antes do nó especificado.

replaceChild(node,oldnode) substitui o nó especificado.

removeChild(node) remove o nó filho especificado.

hasChildNodes() devolve true se tem um ou mais filhos.

cloneNode() devolve uma cópia do corrente nó.

getAttribute(attribute_name) devolve o valor do atributo.

setAttribute(attribute _ name , value) coloca o valor ao atributo.

removeAttribute(attribute_name) , hasAttributes()



Propriedades e métodos do **document**

document.getElementById(*ID*) devolve o nó identificado com ID.

document.getElementsByTagName(*tag*) devolve um array com os nós com marca= tag.

document.createElement(*tag*) cria um novo elemento com marca tag.

document.createTextNode(*text*) cria um o nó texto.

document.documentElement objecto que auto representa-se.

document.body? só em DOM dos documentos em HTML.

JavaScript



Percorrer a estrutura DOM

```
<!DOCTYPE html>
<html>
  <head>
    <title>Walking the DOM</title>
  </head>
  <body>
    <h1>Heading</h1>
    <p>Paragraph with <a href="#">Link</a>
  </body>
</html>
```



BREAK TO
SEE THINGS
HAPPENING!!

JavaScript



Exemplos para práticas

```
document.documentElement;
```

```
<html>
```

```
document.body;
```

```
<body>
```

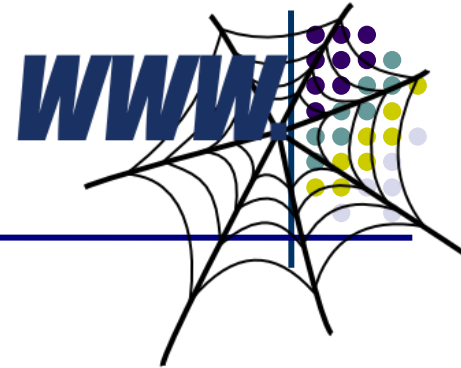
```
document.documentElement.firstChild;
```

```
<head>
```

```
document.documentElement.lastChild;
```

```
<head>
```


JavaScript



...práticas...

Que tipo de
variável é?

```
var bodyChildNodes = document.body.childNodes;  
for (var i = 0, l = bodyChildNodes.length; i < l; i++) {  
    console.log(bodyChildNodes[i]);  
}
```

Array? Então
tem o atributo
length

```
document.body.firstChild.nextSibling;
```

```
<h1>
```

```
document.body.childNodes[1];
```

```
<h1>
```

JavaScript



Localização de nós

`document.getElementById(domId)`

```
<div id="content">
  <div id="sidebar"></div>
</div>
```

`document.getElementById("content");`

```
<div id="content">
```

`document.getElementsByTagName(name)`

JavaScript



```
var content = document.getElementById("content");  
content.getElementsByTagName("div");
```

```
[ <div class="sidebar"> ]
```

document.getElementsByClassName(name)

```
<p>Normal paragraph</p>  
<p class="special">Special paragraph</p>  
<p class="extra special">Special paragraph</p>
```

```
document.getElementsByClassName("special");
```

```
[ p.special, p.extra ]
```

```
document.getElementsByClassName("extra special");
```

```
[ p.extra ]
```

Não está disponível I. Explorer 6 or 7

JavaScript



Criação de nós

`document.createElement (TagName)`

```
document.createElement("div");  
<div>  
document.createElement("a");  
<a>  
document.createElement("body");  
<body>
```

`document.createTextNode (TagName)`

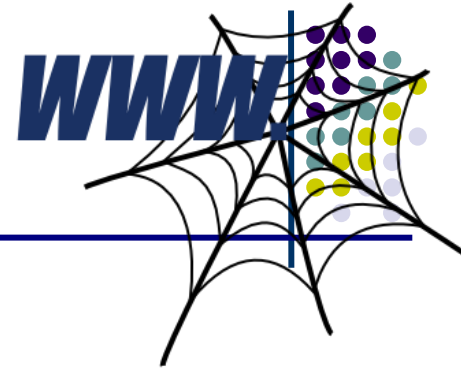
```
<p id="myPara"></p>
```

```
var textNode = document.createTextNode("Inserted Text");  
document.getElementById("myPara").appendChild(textNode);
```

```
<TextNode textContent="Inserted Text">
```

```
<p id="myPara">Inserted Text</p>
```

JavaScript



cloneDeep(booleanValue)

```
<p id="myPara">Click <a href="#">here</a>.</p>
```

```
var myPara = document.getElementById("myPara");  
var deepClone = myPara.cloneNode(true);  
var shallowClone = myPara.cloneNode(false);
```

```
deepClone.childNodes.length;  
3  
shallowClone.childNodes.length;  
0
```

innerHTML(booleanValue)

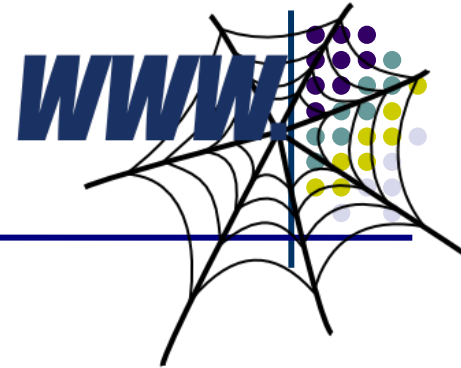
Novo
conteúdo

```
var myPara = document.getElementById("myPara");  
myPara.innerHTML = 'Click <a href="#">here</a>.';
```

```
<p id="myPara">Click <a href="#">here</a>.</p>
```

```
var newLink = myPara.getElementsByTagName("a")[0];
```

acesso



Adição, ordenação e remoção de nós

`appendChild(Node)`

`insertNode(Node, reference)`

`removeChildNode(ChildNode)`

Alteração e inspecção de elementos HTML

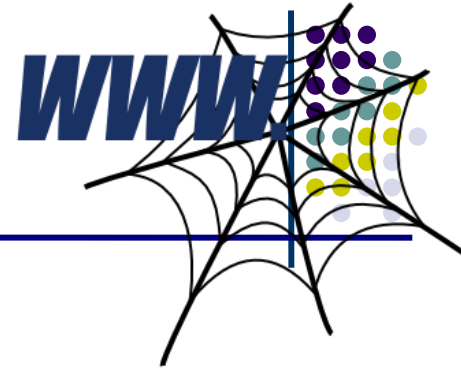
Atributos dos elementos através operadores: ponto (.) ou []

```
<a href="http://www.google.com">Search Engine</a>  
<input type="text" name="name" id="name" value="" />
```

```
var link = document.getElementsByTagName("a")[0];  
link.href;
```

```
"http://www.google.com"
```

JavaScript



```
link.href = "http://www.bing.com";
```

```
"http://www.bing.com"
```

```
var nameInput = document.getElementById("name");  
nameInput.type;
```

```
"text"
```

```
<input type="radio" id="radio1" checked="checked" />  
<input type="radio" id="radio2" />
```

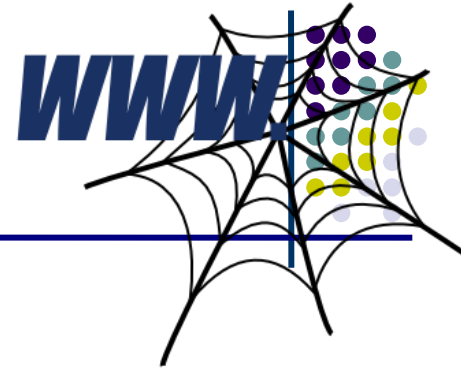
```
var radio1 = document.getElementById("radio1");  
var radio2 = document.getElementById("radio2");  
radio1.checked;
```

```
true
```

```
radio2.checked;
```

```
false
```

```
radio2.checked = true; // radio2 tem valor true (checked)
```



Estilos através de CSS

```
/* In your CSS code */  
.hide {  
  display: none  
}  
  
// In JavaScript  
element.className = "hide";
```

A large, red, multi-pointed starburst or explosion shape with a black outline, positioned in the lower right area of the slide.

**ATENÇÃO!
REVER CSS E
BOX MODEL!!**



Perguntas?

Quem viu JavaScript AJAX?