

Sistemas Multimédia

TPW-II

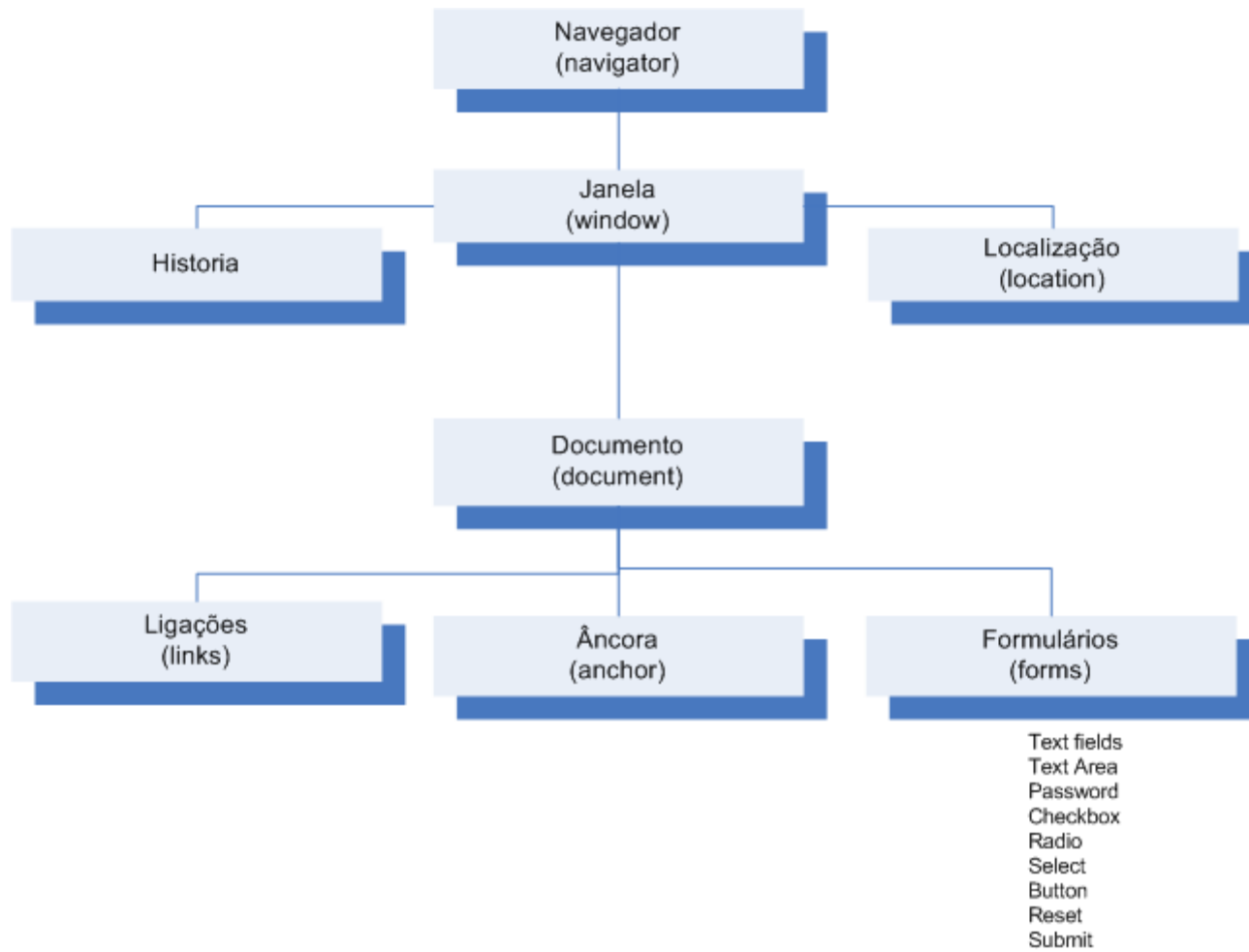
Tecnologia e Prática da Web II

07

JavaScript: outros objectos II

□ Javascript: outros objectos

- Image
- Navigator
- History



Image()

- Para criar um objecto deste tipo, é utilizada a abordagem tradicional dos construtores como, por exemplo:
 - *minhaImagem = new Image();*
- Um objecto do tipo Image() tem as seguintes propriedades:
 - **border** - a propriedade **border** é uma propriedade apenas de leitura e não permite a mudança do seu valor. Para uma imagem criada com o construtor, esta propriedade é sempre 0. Para fazer a leitura desta propriedade, use a sintaxe seguinte: *minhaImagem.border*;

- ❑ **complete** - a propriedade **complete** é uma propriedade apenas de leitura e não permite a mudança do seu valor. Esta propriedade booleana, indica se o browser acabou (true) ou não (false) de carregar a imagem em questão. Para fazer a leitura desta propriedade, use a sintaxe seguinte: *minhaImagem.complete;*
- ❑ **height** - a propriedade **height** é uma propriedade apenas de leitura e não permite a mudança do seu valor. Esta propriedade indica a altura real da imagem em pixels. Para fazer a leitura desta propriedade, use a sintaxe seguinte: *altura = minhaImagem.height;*
- ❑ **hspace** - a propriedade **hspace** é uma propriedade apenas de leitura e não permite a mudança do seu valor. Esta propriedade indica a quantidade de pixels em branco existentes do lado esquerdo e direito da imagem. Para fazer a leitura desta propriedade, use a sintaxe seguinte: *margemHorizontal = minhaImagem.hspace;*

- ❑ **lowsrc** - a propriedade **lowsrc** indica a URL de uma versão de baixa resolução da imagem. Esta versão da imagem é carregada antes da imagem em alta-resolução, de forma a acelerar o processo de carregamento da página. Para utilizar esta propriedade, usar a sintaxe seguinte: *minhaImagem.lowsrc* = "url da imagem";
- ❑ **name** - a propriedade **name** é uma propriedade apenas de leitura e não permite a mudança do seu valor. Esta propriedade indica o nome da imagem. Para fazer a leitura desta propriedade usar a sintaxe seguinte: *minhaImagem.name* = "novo nome";
- ❑ **src** - a propriedade **src** é uma *string* que representa a URL da imagem. Esta propriedade pode ser mudada a qualquer instante. Entretanto, quando o fizermos, uma nova imagem é carregada no lugar da anterior, mas assume a altura e a largura da sua antecessora. Para utilizar esta propriedade, usar a sintaxe seguinte: *minhaImagem.src* = "nova url";

- ❑ **vspace** - a propriedade **vspace** é uma propriedade apenas de leitura e não permite a mudança do seu valor. Esta propriedade indica a quantidade de pixels em branco existentes em cima e em baixo da imagem. Para fazer a leitura desta propriedade, use a sintaxe seguinte: *margemVertical = minhaImagem.vspace;*
- ❑ **width** - a propriedade **width** é uma propriedade apenas de leitura e não permite a mudança do seu valor. Esta propriedade indica a largura real da imagem. Para fazer a leitura desta propriedade, use a sintaxe seguinte: *Largura = minhaImagem.width*

□ Exemplo:



```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title></title>

  <script language="JavaScript" type="text/javascript">

    imgon = new Image();
    imgon.src = "BotaoNormal.gif";
    imgoff = new Image();
    imgoff.src = "BotaoHover.gif";

    function mouseOver()
    {
      document.botaol.src = imgoff.src
    }
    function mouseOut()
    {
      document.botaol.src = imgon.src
    }
  </script>

</script>

</head>

<body>

  <a href="http://www.islagaia.pt"> 
  </a>

</body>

</html>
```

□ Exemplo:



```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title></title>

  <script type="text/javascript">

    function mouseOver()
    {
      document.botao1.src = "BotaoHover.gif";
    }
    function mouseOut()
    {
      document.botao1.src = "BotaoNormal.gif";
    }
  </script>

</head>

<body>
  <a href="http://www.islagaia.pt"> 
  </a>
</body>

</html>
```

Navigator

- ❑ Permite obter informações acerca da aplicação Navigator (Browser)
- ❑ Possui 4 propriedades:
 - `appName`
 - ❑ Retorna o Nome do Browser
 - `appVersion`
 - ❑ Retorna a versão do browser e a versão do sistema operativo
 - `appCodeName`
 - ❑ Retorna o nome do código de desenvolvimento interno
 - `userAgent`
 - ❑ É a combinação das propriedades `appCodeName` e `appVersion`
 - ❑ Os servidores HTTP usam esta informação para identificar os recursos de que o browser dispõe

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title></title>
  <script language="JavaScript" type="text/javascript">

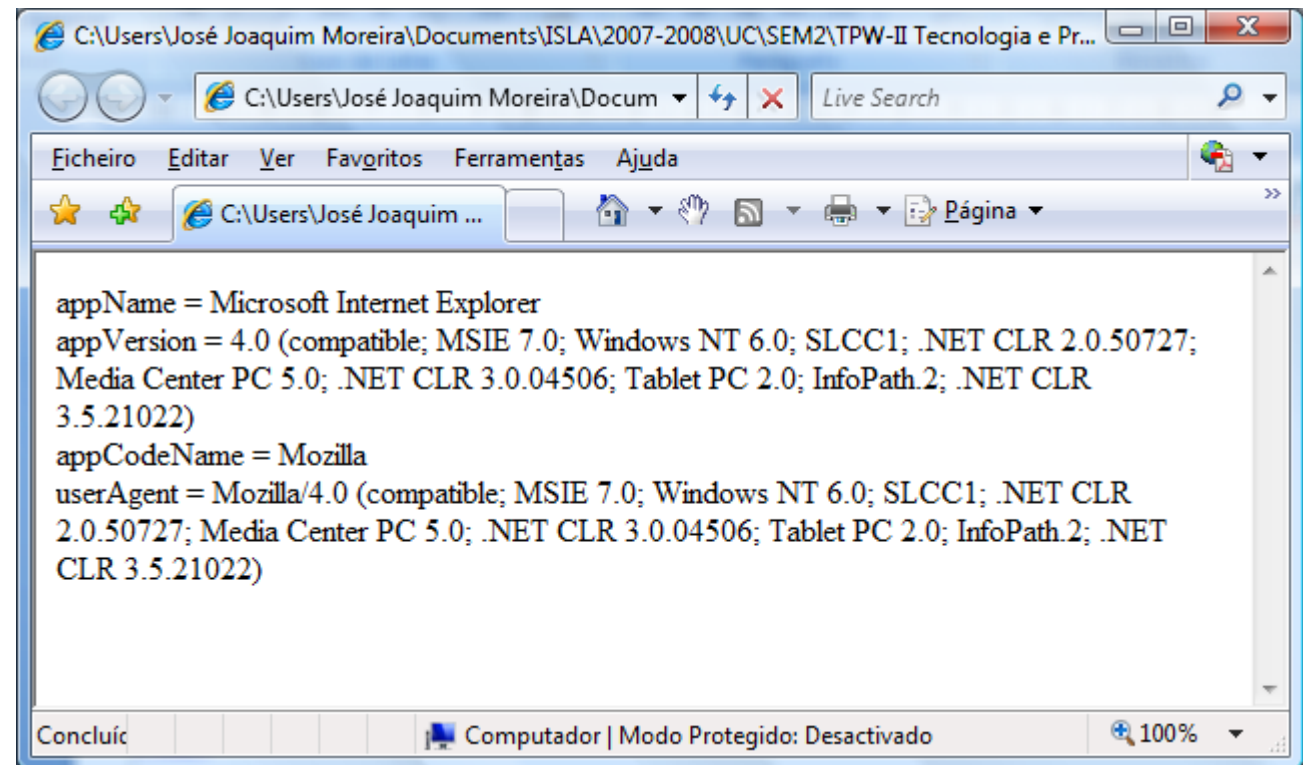
    document.write("appName = " + navigator.appName + " <br /> ");
    document.write("appVersion = " + navigator.appVersion + " <br /> ");
    document.write("appCodeName = " + navigator.appCodeName + " <br /> ");
    document.write("userAgent = " + navigator.userAgent + " <br /> ");

  </script>
</head>

<body>

</body>

</html>
```



History

- O objecto *history* é a interface do javascript com a lista de histórico.
- Podemos programar as acções que habitualmente estão acessíveis através dos botões do browser (avanzar, retroceder, escolher um URL do histórico) para que ocorram automaticamente, ou por exemplo através de um link ou botão na própria página.
- Uma única propriedade:
 - Length
 - Retorna o número de entradas (URLs que foram visitados) da janela activa. Cada janela tem o seu próprio histórico.
- Metodos
 - go()
 - back()
 - forward()

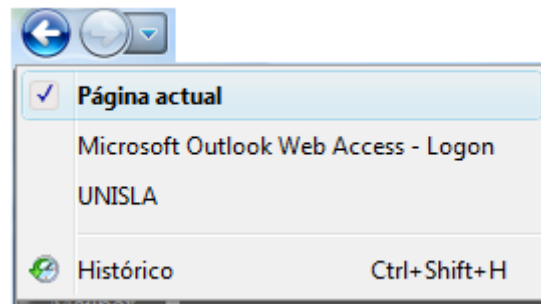
□ go()

- Permite invocar um URL específico da lista de histórico.

- Sintaxe:

- `History.go(alvo)`

- *Alvo* – identifica o URL da lista de histórico para onde pretendemos redireccionar o browser
 - Um valor inteiro corresponde á posição do URL na lista -
 - Um valor de 0 (zero) faz com a página corrente seja recarregada
 - Um valor positivo ou negativo carrega um página que está a tantas entradas antes ou depois na lista de histórico
 - Um valor de string corresponde a parte ou totalidade do URL



□ back()

- Permite retornar à página da qual saímos

- `history.back()`

equivalente a:

- `history.go(-1)`

- Podemos usar:

- `history.back()`

- `history.back()`

- Usar antes: `history.go(-2)`

- forward()

- Permite retornar à próxima página da lista de histórico

- history.forward()

equivalente a:

- history.go(1)

□ Exemplo

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title></title>
</head>

<body>

  <h1>Página principal</h1>
  <a href="07-novapagina.html">link para outra página</a>

</body>

</html>
```

Página principal

[link para outra página](#)

□ Exemplo

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title></title>
</head>

<body>
  <h1>Nova página</h1>

  <input type="button" onclick="history.back()" value="Voltar" />
</body>

</html>
```

Nova página

Voltar