

Sistemas Multimédia

TPW-II

Tecnologia e Prática da Web II

02

JavaScript: objects

□ Javascript

- Classes e Objectos
- Objectos predefinidos
- Exemplos com objectos

Javascript

objectos

- Um objecto é uma colecção de propriedades. Estas propriedades podem ser variáveis, outros objectos ou funções (que no caso dos objectos são chamadas de métodos).
 - Classe de objectos
 - Principais elementos dos objectos:
 - Propriedades
 - Métodos

- Declarações para manipular objectos
 - new
 - this
 - for..in
 - with

- Características das classes
 - Construtor
 - Herança
 - Polimorfismo
 - Encapsulamento

□ new

- Para criar uma instancia de objecto é preciso usar uma função especial chamada de função de construção (constructor function) ou simplesmente construtor (constructor).

- O construtor corresponde ao comando **new**

- Sintaxe:

```
novaVar = new tipoObjecto(parametrosOpcionais);
```

- Exemplo:

```
DataHora = new Date();
```

```
Wed Mar 19 13:55:50 UTC 2008
```

□ **this**

- Os objectos são constituídos por *propriedades e métodos*
- Muitos dos objectos embutidos (como *Document* e *Window*) contêm outros objectos, os quais podem conter outros objectos
- Às vezes é difícil saber exactamente onde estamos na árvore de objectos, ou simplesmente para simplificar (não ter que escrever o caminho completo), podemos usar a declaração **this**.
- Exemplo:
 - documento.cookie
 - Ou
 - this.cookie
 - ``
aqui **this** representa a âncora, e propriedade *href* é `http://www.islagaia.pt`

□ for .. in

- Uma declaração *for..in* repete uma variável sobre todas as propriedades de um objecto.
- É um tipo especial de declaração de ciclo que (em vez de usar um contador ou incremento) varre todas as propriedades de um objecto.
- Sintaxe:

```
for (nomePropriedade in objecto) {  
    comandos;  
}
```

Exemplo:

```
for (propertyName in document){  
    document.write(propertyName + "<BR>");  
}
```

Lista todas as propriedades do objecto document

□ with

- Se quisermos manipular várias propriedades de um objecto, podemos utilizar a declaração *with* para evitar escrever sempre o nome do objecto.

- Sintaxe:

```
with (document) {  
}
```

exemplo:

```
with (document){  
    fgColor = "#CC0000";  
    bgColor = "#FFFFFF";  
}
```

```
document.fgColor = "#CC0000";  
document.bgColor = "#FFFFFF";
```


Javascript

objetos predefinidos

Javascript: objetos predefinidos

❑ Objetos predefinidos

Objeto	Propriedades	Métodos	Event Handlers
Window	defaultStatus frames opener parent scroll self status top window	alert blur close confirm focus open prompt clearTimeout setTimeout	onLoad onUnload onBlur onFocus
Frame	defaultStatus frames opener parent scroll self status top window	alert blur close confirm focus open prompt clearTimeout setTimeout	nenhum
Location	hash host hostname href pathname por protocol search	reload replace	nenhum

❑ Objetos predefinidos (cont)

Objeto	Propriedades	Métodos	Event Handlers
History	length forward go	back	nenhum
Navigator	appName appVersion mimeTypes plugins userAgent top window	javaEnabled clearTimeout setTimeout	nenhum
document	alinkColor anchors applets area bgColor cookie fgColor forms images lastModified linkColor links location referrer title vlinkColor	clear close open write writeln	nenhum

□ Objectos predefinidos (cont)

Objecto	Propriedades	Métodos	Event Handlers
image	border complete height hspace lowsrc name src vspace width	nenhum	nenhum
form	action elements encoding FileUpload method name target	submit reset	onSubmit onReset
text	defaultValue name type value	focus blur select	onBlur onCharge onFocus onSelect

- **Window** – representa a janela do browser;
- **Frame** – representa o frame do browser;
- **Location** – representa a URL da página que está a ser visitada;
- **History** – representa o histórico das páginas visitadas pelo utilizador;
- **Navigator** – representa o browser;
- **document** – representa a página que está a ser vista;
- **image** – representa uma imagem da página que está a ser vista;
- **form** – representa um formulário que exista na página;
- **text** – representa um texto na página.

❑ Objetos predefinidos (cont)

Objecto	Propriedades	Métodos	Event Handlers
Array	length	join reverse sort xx	nenhum
Date	nenhum	getDate getDay getHours getMinutes getMonth getSeconds getTime getTimeZoneoffset getYear parse prototype setDate setHours setMinutes setMonth setSeconds setTime setYear toGMTString toLocaleString UTC	nenhum

❑ Objectos predefinidos (cont)

Objecto	Propriedades	Métodos	Event Handlers
String	length prototype	anchor big blink bold charAt fixed fontColor fontSize indexOf italics lastIndexOf link small split strike sub substring sup toLowerCase toUpperCase	Window
<ul style="list-style-type: none">▪ Array - permite a criação de matrizes;▪ Date - fornece a data e hora do sistema e permite fazer operações com estes;▪ String - permite lidar com textos associados aos objectos.			

Javascript

Exemplos com objectos

□ Date()

```
var datahora = new Date();  
var hora = datahora .getHours();
```

```
= var hora = new Date().getHours();
```

```
if (hora < 12)  
    alert("Bom Dia");  
else if (hora >= 12 && hora < 18)  
    alert("Boa Tarde");  
else  
    alert("Boa Noite");
```

```
var dataHora = new Date();  
//dataHora = Thu Mar 13 2014 21:59:37 GMT+0000 (Hora padrão de GMT)
```


Javascript: exemplos com objectos

□ Date()

■ Data corrente

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

<head>
  <title>Data Corrente</title>

  <script language=JavaScript>
    <!-- ocultar script para browsers mais antigos --

    function getDataCorrente(){

      var DataHora = new Date();
      var DataCorrente;

      var Mes = DataHora.getMonth();

      if (Mes == 0) MesExtenso = 'Janeiro';
      if (Mes == 1) MesExtenso = 'Fevereiro';
      if (Mes == 2) MesExtenso = 'Março';
      if (Mes == 3) MesExtenso = 'Abril';
      if (Mes == 4) MesExtenso = 'Maio';
      if (Mes == 5) MesExtenso = 'Junho';
      if (Mes == 6) MesExtenso = 'Julho';
      if (Mes == 7) MesExtenso = 'Agosto';
      if (Mes == 8) MesExtenso = 'Setembro';
      if (Mes == 9) MesExtenso = 'Outubro';
      if (Mes == 10) MesExtenso = 'Novembro';
      if (Mes == 11) MesExtenso = 'Dezembro';

      var Ano = DataHora.getFullYear();
      if (eval(Ano) < 2000)
        Ano = "19" + sYear;
      DataCorrente = DataHora.getDate() + " de " + MesExtenso + " de " + Ano

      return(DataCorrente)
    }
    -->
  </script>
</head>

<body>

  <script language=JavaScript>
    document.write("Data: " + getDataCorrente() + '<br>');
  </script>

</body>

</html>
```

□ Outra forma

```
function DataCorrenteSimplificada()
{
    var arrayDiasSemana = new Array("domingo", "segunda-feira", "terca-feira",
    "quarta-feira", "quinta-feira", "sexta-feira", "sabado")

    var arrayMeses = new Array("janeiro", "fevereiro", "marco",
    "abril", "maio", "junho", "julho", "agosto", "setembro",
    "outubro", "novembro", "dezembro")

    var dataHora = new Date();
    //dataHora = Thu Mar 13 2014 21:59:37 GMT+0000 (Hora padrão de GMT)

    var dataFormatada;
    //dataFormatada = quinta-feira, 13 de março de 2014

    var diaSemana = dataHora.getDay(); //0..dom, 1-seg...
    var diaSemanaExtenso;
    var diaMes = dataHora.getDate();

    var mesNumerico = dataHora.getMonth(); //0..jan, 1-fev...
    var mesExtenso;

    diaSemanaExtenso = arrayDiasSemana[diaSemana];
    mesExtenso = arrayMeses[mesNumerico];

    dataFormatada = diaSemanaExtenso + ", " + diaMes + " de " + mesExtenso + " de " + dataHora.getFullYear();

    return dataFormatada;
}
```