

Objeto Date

new Date()

Este método reenvia todas as informações "data e hora" do computador do usuário.

```
variavel=new Date();  
(as variáveis não levem acentos, por isso "variavel" e não "variável")
```

Estas informações são gravadas pelo Javascript sob o formato :

```
"Fri Feb 17 09:23:30 2001"
```

Atenção! a data e a hora no Javascript começa em 1º Janeiro 1970. Toda referência a uma data anterior irá dar um resultado aleatório.

O método **new date()** sem argumentos reenvia a data e a hora corrente.

Para introduzir uma data e uma hora determinada, será da seguinte maneira:

```
variavel=new date("Jan 1, 2000 00:00:00");
```

getFullYear()

```
variavel_data=new date();  
an=variavel_data.getFullYear();
```

Devolve os dois últimos números do ano na variavel_data. Ou seja aqui 01.

Como só se tem dois números, pode-se colocar 20 como prefixo, assim:

```
an="20"+variavel_data.getFullYear();
```

getMonth()

```
variavel_data=new date();  
mes=variavel_data.getMonth();
```

Devolve o mês na variável_date sob a forma de um inteiro compreendido entre 0 e 11 (0 para janeiro, 1 para fevereiro, 2 para março, etc.). Ou seja 11 (o mês menos 1). Por isso para obter o mês certo pode-se fazer como este exemplo seguinte:

```
<SCRIPT LANGUAGE="Javascript">  
<!--  
variavel_data=new date();  
mes=variavel_data.getMonth();  
document.write("Estamos no "+(mes+1)+"º mês do ano.");  
//-->  
</SCRIPT>
```

o resultado será:

Estamos no 2º mês do ano.

getDate()

```
variavel_data=new date();  
diam=variavel_data.getDate();
```

Devolva o dia do mês na variavel_data sob a forma de um inteiro compreendido entre 1 e 31. Pois é, aqui começa-se em 1 em vês de 0 (porquê???). Não confundir com getDay() que Devolva o dia da semana.

getDay()

```
variavel_data=new date();  
dia=variavel_data.getDay();
```

Devolva o dia da semana na variavel_data sob a forma de um inteiro compreendido entre 0 e 6 (0 para Domingo, 1 para Segunda-feira, etc.).

getHours()

```
variavel_data=new date();  
horas=variavel_data.getHours();
```

Devolva a hora na variavel_data sob a forma de um inteiro compreendido entre 0 e 23.

getMinutes()

```
variavel_data=new date();  
minutos=variavel_data.getMinutes();
```

Devolva os minutos na variavel_data sob a forma de um inteiro compreendido entre 0 e 59.

getSeconds()

```
variavel_data=new date();  
segundos=variavel_data.getSeconds();
```

Devolva os segundos na variavel_data sob a forma de um inteiro compreendido entre 0 e 59.

Exemplo: Um script que dá a hora em que entrou na página.

4/11/192003 10:43:06

```
<HTML>  
<HEAD>  
<SCRIPT LANGUAGE="Javascript">  
<!--  
function getDt(){  
dt=new date();  
cal=""+ dt.getDate()+"/"+(dt.getMonth()+1)  
+dt1.getYear();  
hrs=dt.getHours();  
min=dt.getMinutes();  
sec=dt.getSeconds();  
tm=" "+((hrs<10)?"0":"")+hrs+":";  
tm+=((min<10)?"0":"")+min+":";  
tm+=((sec<10)?"0":"")+sec+" ";  
document.write(cal+tm);  
}  
// -->  
</SCRIPT>  
</HEAD>  
<BODY >  
<SCRIPT LANGUAGE="Javascript">  
<!--
```

```
getDt();
// -->
</SCRIPT>
</BODY>
</HTML>
```

Exemplo: Um script com horas dinâmicas

Para quem quer que a hora altera-se todos os segundos. Recorda-se do temporizador seTimeout [Um pouco de tudo...]. Basta acrescentar ao script um seTimeout que insere a hora todos os segundos. A função que afixa a hora é getDt(), a instrução à acrescentar é **seTimeout("getDt()",1000)**; E o resultado é:

10:43:11

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="Javascript">
<!--
function getDt(){
dt=new date();
hrs=dt.getHours();
min=dt.getMinutes();
sec=dt.getSeconds();
Tm=" "+((hrs<10)?"0":"")+hrs+":";
Tm+=((min<10)?"0":"")+min+":";
Tm+=((sec<10)?"0":"")+sec+" ";
document.relogio.display.value=Tm;
setTimeout("getDt()",1000);
}
// -->
</SCRIPT>
</HEAD>
<BODY onLoad="getDt()">
<FORM name="relogio">
<INPUT TYPE="text" NAME="display" SIZE=15 VALUE ="">
</FORM>
</BODY>
</HTML>
```

Outras propriedades (menos frequentes talvez)

getTime()

Devolva a hora corrente na variavel_data sob a forma de um inteiro representando o número de milisegundos que passaram desde 1 Janeiro 1970 00:00:00.

getTimezoneOffse()

Devolva a diferença entre a hora local e a hora GMT (Greenwich, UK Mean Time) sob a forma de um inteiro representando o número de minutos (e não em horas).

seMonth(x)

Devolva um mês ao atual valor da variavel_data sob a forma de um inteiro compreendido entre 0 e 11.

Exemplo : variavel_data.seMonth(1);

seDate(x)

Devolva um dia do mês ao atual valor da variavel_data sob a forma de um inteiro compreendido entre 1 e 31.

Exemplo : `variavel_data.seDate(1);`

seHours(x)

Devolva uma hora ao atual valor da variavel_data sob a forma de um inteiro compreendido entre 1 e 23.

Exemplo : `variavel_data.seHours(1);`

seMinutes(x)

Devolva os minutos ao atual valor de variavel_data sob a forma de um inteiro compreendido entre 1 e 59.

Exemplo : `variavel_data.seMinutes(1);`

seSeconds(x)

Devolva os segundos ao atual valor de variavel_data sob a forma de um inteiro compreendido entre 1 e 59.

Exemplo : `variavel_data.seSeconds(0);`

seTime(x)

Devolva a data desejada na variavel_data sob a forma de um inteiro representando o número de milisegundos que passaram desde 1 Janeiro 1970 00:00:00.

toGMTString()

Devolva o valor do atual valor da variavel_data em hora GMT (Greenwich Mean Time)