

Administração Central
Cetec Capacitações

JS

Aula 12 – Objetos e Métodos

1. Objeto DATE

O objeto DATE permite que o usuário possa trabalhar com datas e horas.

```
<html>
<head>
<title> Aula de JavaScript </title>
</head>
<body>
<div id="data"><script>document.write(new Date());</script></div>
</body>
</html>
```

Objeto01.html

Vamos modificar este exemplo, usando uma função para mostrar a data. Desta vez vamos criar uma função automática, ou seja, será executada sem a ação do usuário.

```
<html>
<head>
<title> Aula de JavaScript </title>
<script type="text/javascript">
    funcaoTeste = function()
    {
        novaData = new Date();
        var dataAtual = document.getElementById("data");
        dataAtual.innerHTML = novaData;
    };
</script>
</head>
<body>
<div id="data">
<script>funcaoTeste()</script>
</div>
</body>
```

Administração Central
Cetec Capacitações

</html>

Objeto02.html

O mesmo exemplo, mas agora usando o evento **onload** para ativar a função.

```
<html>
<head>
<title> Aula de JavaScript </title>
</head>
<script type="text/javascript">
function DataHora() {
    var novadata = new Date();
    document.getElementById("datahora").innerHTML = novadata;
}
</script>
<body onload="DataHora()">
    <div id="datahora"></div>
</body>
</html>
```

Objeto03.html

O objeto DATE não possui propriedades, por outro lado, possui muitos métodos:

- **getDate()**: devolve o dia do mês, um inteiro entre 1 e 31. Não confundir com **getDay** que retorna o dia da semana.
- **getDay()**: devolve o dia da semana, inteiro entre 0 e 6 (0 para Domingo).
- **getHours()**: retorna a hora, inteiro entre 0 e 23.
- **getMinutes()**: devolve os minutos, inteiro entre 0 e 59.
- **getSeconds()**: devolve os segundos, inteiro entre 0 e 59.
- **getMonth()**: devolve o mês, um inteiro entre 0 e 11 (0 para Janeiro).
- **getTime()**: devolve os segundos transcorridos entre o dia 1 de janeiro de 1970 e a data correspondente ao objeto ao que se passa a mensagem.
- **getFullYear()**: retorna o ano, os últimos dois números do ano. Por exemplo, para o 2006 retorna 06. Este método está obsoleto em Netscape a partir da versão 1.3 de Javascript e agora se utiliza **getFullYear()**.

Administração Central
Cetec Capacitações

- **getFullYear():** retorna o ano com todos os dígitos. Funciona com datas posteriores ao ano 2000.
- **setDate(d):** atualiza o dia do mês.
- **setHours(h):** atualiza a hora.
- **setMinutes(m):** muda os minutos.
- **setMonth(m):** muda o mês (atenção ao mês que começa por 0).
- **setSeconds(s):** muda os segundos.
- **setTime(t):** atualiza a data completa. Recebe um número de segundos desde 1 de janeiro de 1970.
- **setYear(y):** Muda o ano, recebe um número, ao que lhe soma 1900 antes de colocá-lo como ano data. Por exemplo, se receber 95 colocará o ano 1995. Este método está obsoleto a partir de Javascript 1.3 em Netscape. Agora se utiliza **setFullYear()**, indicando o ano com todos os dígitos.
- **setFullYear():** muda o ano da data ao número que recebe por parâmetro. O número se indica completo ex: 2005 ou 1995. Utilizar este método para estar certo de que tudo funciona para datas posteriores a 2000.
- **getTimezoneOffset():** 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).
- **toGMTString():** devolva o valor do atual em hora GMT (Greenwich Mean Time)

Fonte: <http://www.linhadecodigo.com.br/artigo/1003/entendendo-o-objeto-date-do-javascript.aspx#ixzz4l2YwvhXk>

Esses métodos devem ser utilizados com o objeto DATE
O exemplo abaixo utiliza o método **getDate()**.

```
<html>
<head>
<title> Aula de JavaScript </title>
</head>
<script type="text/javascript">
function DataAtual() {
    var dataAtual = new Date();
    var dia = dataAtual.getDate();
    document.getElementById("datahora").innerHTML = dia;
}
</script>
```

Administração Central
Cetec Capacitações

```
<body onload="DataAtual()">
  <div id="datahora"></div>
</body>
</html>
```

Objeto04.html

Veja um exemplo mais elaborado, execute o código **RelógioJavaScript.html**.

2. Objeto STRING

O objeto String possui a propriedade **length()**, que informa o comprimento da string.

```
<script type="text/javascript">
  var frase = "Aula de JavaScript";
  document.write(frase.length);
</script>
```

Objeto05.html

Outro exemplo para utilização do **length()**.

```
<script type="text/javascript">
  document.write("Aula de JavaScript - Exemplo 02".length);
</script>
```

Objeto06.html

Por outro ponto de vista.

```
<html>
<head>
<title> Aula de JavaScript </title>
</head>
<body>
<div><script>document.write("Aula de JavaScript - Exemplo 03 - Novo".length);</script></div>
</body>
</html>
```

Objeto07.html

2.1 Método BIG()

Tem a função de aumentar a fonte e atribuir o estilo de negrito.

Administração Central
Cetec Capacitações

```
<script type="text/javascript">
    texto="Aula de JavaScript";
    document.write(texto);
    document.write("<br>");
    document.write(texto.big());
</script>
```

Objeto08.html

2.2 Método SMALL()

Tem a função de reduzir o tamanho da fonte.

```
<script type="text/javascript">
    texto="Aula de JavaScript";
    document.write(texto);
    document.write("<br>");
    document.write(texto.small());
</script>
```

Objeto09.html

2.3 Método BOLD()

Referente ao tag HTML que tem a função de atribuir o estilo de negrito sobre o texto.

```
<script type="text/javascript">
    texto="Aula de JavaScript";
    document.write(texto);
    document.write("<br>");
    document.write(texto.bold());
</script>
```

Objeto10.html

2.4 Método BOLD()

Este método é referente ao tag HTML <I> que atribui o estilo de itálico em um texto.

Administração Central
Cetec Capacitações

```
<script type="text/javascript">
    texto="Aula de JavaScript";
    document.write(texto);
    document.write("<br>");
    document.write(texto.italics());
</script>
```

Objeto11.html

Como utilizar mais de um método? Veja o exemplo.

```
<script type="text/javascript">
    texto="Aula de JavaScript";
    document.write(texto);
    document.write("<br>");
    document.write(texto.italics().bold());
</script>
```

Objeto12.html

2.5 Método FIXED()

Com o método fixed, é possível formatar o qualquer texto em fonte fixa, ou como conhecido em HTML, em fonte **monoespaço** definido pelo tag <TT>.

É importante ressaltar que dentro do contexto do HTML 4.01 o marcador <TT></TT> está em desuso e será obsoleto no HTML 5. Preferencialmente deve-se usar a tag <CODE></CODE>.

```
<script type="text/javascript">
    texto="Aula de JavaScript";
    document.write(texto.fixed());
</script>
```

Objeto13.html

2.6 Método STRIKE()

Este método tem a função de criar um texto tachado que exibe uma linha no meio do texto exibido.

Administração Central
Cetec Capacitações

```
<script type="text/javascript">
    texto="Aula de JavaScript";
    document.write(texto.strike());
</script>
```

Objeto14.html

2.7 Método FONTCOLOR()

Determina a cor da fonte em um texto de acordo com o tag HTML .

```
<script type="text/javascript">
    texto="Aula de JavaScript";
    document.write(texto.fontcolor("blue"));
</script>
```

Objeto15.html

Outro exemplo:

```
<script type="text/javascript">
    texto="Aula de JavaScript";
    document.write(texto.fontcolor("#ff0000"));
</script>
```

Objeto16.html

2.8 Método FONTSIZE()

Este método, determina o tamanho da fonte seguindo os padrões do tag HTML que possui tamanhos **que vão de 1 a 7**, assim como a possibilidade de valores relativos através dos sinais de + e -.

```
<script type="text/javascript">
    texto="Aula de JavaScript";
    document.write(texto.fontSize(01));
    document.write("<br>");
    document.write(texto.fontSize(07));
</script>
```

Objeto17.html

Outro exemplo:

Administração Central
Cetec Capacitações

```
<script type="text/javascript">
    texto="Aula de JavaScript";
    document.write(texto.fontSize(07).fontcolor("red"));
</script>
```

Objeto18.html

2.9 Método SUB()

Este método cria um texto subscrito tendo o mesmo efeito do tag HTML <SUB>.

```
<script type="text/javascript">
    texto="Aula de JavaScript ";
    document.write(texto);
    document.write(texto.sub());
</script>
```

Objeto19.html

2.10 Método SUP()

Este método cria um texto sobrescrito tendo o mesmo efeito do tag HTML <SUP>.

```
<script type="text/javascript">
    texto="Aula de JavaScript ";
    document.write(texto);
    document.write(texto.sup());
</script>
```

Objeto20.html

2.11 Método charAT()

Com este método o usuário poderá retornar o caractere em uma determinada posição em uma string.

```
<script type="text/javascript">
    texto="Aula de JavaScript ";
    document.write(texto.charAt(8)+texto.charAt(12));
</script>
```

Objeto21.html

Administração Central
Cetec Capacitações

2.12 Método indexOf()

Com o método indexOf o usuário pode retornar a posição de um caractere dentro de uma string. Um exemplo claro do método indexOf, é a maneira de saber se determinada string possui algum caractere específico. Caso a string não contiver o caractere específico, o método irá retornar o valor -1, caso haja a ocorrência do caractere procurado, será retornado o valor 0 ou superior, sendo que 0 é a posição do primeiro caractere da string, 1 a posição do segundo caractere e assim por diante.

```
<script type="text/javascript">
    texto="Aula de JavaScript @ 2017";
    document.write(texto.indexOf("@"));
</script>
```

Objeto22.html

2.13 Método indexOf()

Este método é similar ao tag HTML <A HREF> que tem a função de criar hiperlinks em uma página.

```
<script type="text/javascript">
    texto="Aula de JavaScript";
    document.write(texto.link("https://developer.mozilla.org/pt-BR/docs/Web/JavaScript"));
</script>
```

Objeto23.html

2.14 Método REPLACE()

Este método tem a função de trocar valores dentro de uma string.

```
<script type="text/javascript">
    texto="Aula de JavaScript 2016";
    texto = texto.replace("6","7");
    document.write(texto);
</script>
```

Objeto24.html

2.15 Método SUBSTRING()

Este método retorna uma parte de uma string.

Administração Central
Cetec Capacitações

```
<script type="text/javascript">
    texto="Aula de JavaScript 2017";
    document.write(texto.substring(8,18));
</script>
```

Objeto25.html

2.16 Método toLowerCase()

O método toLowerCase converte uma string em letras minúsculas.

```
<script type="text/javascript">
    texto="Aula de JavaScript 2017";
    document.write(texto.toLowerCase());
</script>
```

Objeto26.html

2.17 Método toUpperCase()

O método toUpperCase converte uma string em letras maiúsculas.

```
<script type="text/javascript">
    texto="Aula de JavaScript 2017";
    document.write(texto.toUpperCase());
</script>
```

Objeto27.html

3. Objeto IMAGE

Na linguagem JavaScript as imagens que são inseridas através da linguagem HTML são consideradas cada uma um objeto do tipo IMAGE. Com isto, podemos concluir que as imagens possuem propriedades e métodos assim como os outros objetos já existentes.

```
<html>
<head>
<title> Aula de JavaScript </title>
</head>
<body>

<br>
<input type="button" value="Muda Figura" onClick="document.imagem.src='smiley02.png'">
<br>
```

Administração Central
Cetec Capacitações

```
<input type="button" value="Volta Figura" onClick="document.imagem.src='smiley01.png'">
</body>
</html>
```

Objeto28.html

Este exemplo é diferente dos anteriores onde trocamos a imagem com o efeito do mouse, agora vamos trocar propriedades.

```
<html>
<head>
<title> Aula de JavaScript </title>
<script type="text/javascript">
    function figura(valor){
        document.imagem.width=valor;
    }
</script>
</head>
<body>

</body>
</html>
```

Objeto29.html

Entre diferentes soluções, esta seria uma delas para resolver o exercício 03 da semana 02!!!!