

TUTORIAIS JAVASCRIPT

O Objeto Date

Copyright 2013 – Todos os Direitos Reservados
Jorge Eider F. da Silva

Proibida a reprodução deste documento no todo ou em parte por quaisquer meios, seja digital, eletrônico ou impresso sem a expressa autorização do autor por escrito. Os infratores serão punidos de acordo com a Lei.

O Objeto Date

Introdução

O objeto **Date** é utilizado para trabalhar com datas e horas. O objeto **Date** é criado utilizando o construtor **Date()**. Veja a seguir como criar um objeto **Date**.

Há quatro maneiras de inicializar uma data:

```
new Date()           // Data e hora atual.  
new Date(milliseconds) // Millisegundos desde 01/01/1970.  
new Date(dateString)  
new Date(year, month, day, hours, minutes, seconds, milliseconds)
```

A maioria desses parâmetros são opcionais. Se algum não for especificado, será considerado como 0 (zero).

Uma vez um objeto **Date** esteja criado, você poderá utilizar os métodos disponíveis para ele da maneira que achar mais conveniente, tais como, **day**, **month**, **year**, **hour**, **minute**, etc. Posteriormente mostraremos alguns desses métodos em ação.

Todas as datas são calculadas a partir de 01/01/1970 Universal Time (UTC), com um dia contendo 86.400.000 milisegundos. Vejamos alguns exemplos de inicialização de datas:

```
var minhaData = new Date();  
var d1 = new Date("October 13, 1975 11:13:00");  
var d2 = new Date(79,5,24);  
var d3 = new Date(79,5,24,11,33,0);
```

Podemos facilmente manipular a data e a hora utilizando os métodos disponíveis para o objeto **Date**. Vejamos a seguir alguns métodos e como utilizá-los na prática:

O método getDate()

Esse método retorna um número representando o dia do mês, ou seja, de 1 a 31. Por exemplo:

```
var minhaData = new Date();  
var diaMes = minhaData.getDate();
```

O método getDay()

Esse método retorna um número representando o dia da semana, ou seja, de 0 (para Domingo) a 6 (para Sábado). Por exemplo:

```
var minhaData = new Date();  
var diaSemana = minhaData.getDay();
```

Para substituir os números retornados pelo nome do dia da semana você poderia criar uma função para isso. Veja no exemplo mais adiante como fazer essa função.

O método getFullYear()

Esse método retorna o ano atual representado por quatro dígitos (2013). Por exemplo:

```
var minhaData = new Date();  
var anoAtual = minhaData.getFullYear();
```

Vejam agora um exemplo prático utilizando esses métodos:

Exemplo 1

Nesse exemplo vamos utilizar os três métodos anteriores para criar a data atual do sistema, do tipo:

Natal (RN), 23 de Fevereiro de 2013 (Sábado).

Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js15-exemplo1.html**.

```
<!DOCTYPE html>  
<html>  
Tutorial JavaScript15 - Exemplo 1 - O Objeto Date  
<head>  
  <title>Tutorial JavaScript15 - Exemplo 1 - O Objeto Date</title><p>  
<script language="javascript">  
  var minhaData = new Date();  
  // Dia do mês: 1 a 31.  
var diaMes = minhaData.getDate();  
// Número representando o mês: 0 a 11.  
var mes = minhaData.getMonth();  
  // Número representando o dia da semana: 0 a 6.  
var diaSemana = minhaData.getDay();  
  // Ano com quatro dígitos.  
var anoAtual = minhaData.getFullYear();  
var dia;  
  // Função para os dias da semana.  
  function semana(){  
    if(diaSemana==0){  
      dia="Domingo";  
    }  
    else if(diaSemana==1){  
      dia="Segunda-feira";  
    }  
    else if(diaSemana==2){  
      dia="Terça-feira";  
    }  
    else if(diaSemana==3){  
      dia="Quarta-feira";  
    }  
    else if(diaSemana==4){  
      dia="Quinta-feira";  
    }  
    else if(diaSemana==5){  
      dia="Sexta-feira";  
    }  
    else if(diaSemana==6){  
      dia="Sábado";  
    }  
    return dia;  
  }  
</script>  
</p>  
</head>  
</html>
```

```

// Função para os nomes dos meses do ano.
function nomeMes(){
    var meses = ["Janei-
ro","Fevereiro","Março","Abril","Maio","Junho","Julho","Agosto","Setembro","Outubro
","Novembro","Dezembro"];
    var m = meses[mes];
    return m;
}
document.write("Natal (RN), " + diaMes + " de " + nomeMes() + " de " +
anoAtual + " (" + semana()+ ").");
</script>
</head>
<body>
</body>
</html>

```

2. O resultado desse código após executado dependerá da data atual do sistema, mas será do tipo:

Tutorial JavaScript15 - Exemplo 1 - O Objeto Date

Natal (RN), 1 de Maio de 2013 (Quarta-feira).

O método getHours()

Esse método retorna um número representando as horas, entre 0 e 23. Por exemplo:

```

var minhaData = new Date();
var horas = minhaData.getHours();

```

O método getMinutes()

Esse método retorna um número representando os minutos, entre 0 e 59. Por exemplo:

```

var minhaData = new Date();
var minutos = minhaData.getMinutes();

```

O método getSeconds()

Esse método retorna um número representando os segundos, entre 0 e 59. Por exemplo:

```

var minhaData = new Date();
var segundos = minhaData.getSeconds();

```

Vejamos agora um exemplo prático utilizando esses métodos:

Exemplo 2

Nesse exemplo vamos utilizar os três métodos anteriores para mostrar a hora atual do sistema, do tipo:

Agora são exatamente: 00:00:00.

Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js15-exemplo2.html**.

```

<!DOCTYPE html>
<html>
Tutorial JavaScript15 - Exemplo 2 - O Objeto Date
<head>
  <title>Tutorial JavaScript15 - Exemplo 2 - O Objeto Date</title><p>
<script language="javascript">
  var minhaData = new Date();
  var hora = minhaData.getHours(); // Captura a hora atual: 0 a 23.
  var minutos = minhaData.getMinutes(); // Captura os minutos: 0 a 59.
  var segundos = minhaData.getSeconds(); // Captura os segundos: 0 a 59.
  document.write("Agora são exatamente: " + hora + ":" + minutos + ":" +
segundos + ".");
</script>
</head>
<body>
</body>
</html>

```

2. O resultado desse código após executado dependerá da hora atual do sistema, mas será do tipo:

Tutorial JavaScript15 - Exemplo 2 - O Objeto Date

Agora são exatamente: 9:39:13.

Existem mais métodos para o objeto **Date**, mas esses são os mais utilizados.

Comparando duas datas

O objeto **Date** também pode ser usado para comparar duas datas. Vejamos um exemplo prático:

Exemplo 3

Nesse exemplo vamos mostrar através de uma janela de diálogo a comparação entre a data atual do sistema com a data: 01 de julho de 2014. Será mostrada uma mensagem informando se a data atual é anterior ou posterior à data dada.

Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js15-exemplo3.html**.

```

<!DOCTYPE html>
<html>
Tutorial JavaScript15 - Exemplo 3 - O Objeto Date
<head>
  <title>Tutorial JavaScript15 - Exemplo 3 - O Objeto Date</title><p>
<script language="javascript">
  var novaData = new Date();
  novaData.setFullYear(2014,6,1);
  var hoje = new Date();
  if (novaData>hoje)
  {
    alert("A data de hoje é anterior a 01 de Julho de 2014.");
  }
  else

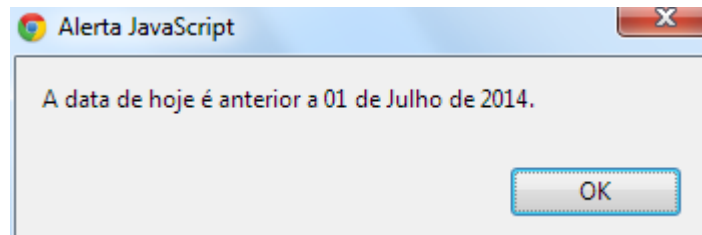
```

```

    {
      alert("A data de hoje é posterior a 01 de Julho de 2014.");
    }
  </script>
</head>
<body>
</body>
</html>

```

2. O resultado desse código após executado dependerá da data que você utilizar em relação à data do sistema. Se ela for anterior, a caixa de diálogo a seguir será mostrada:



Conheça outros métodos do objeto **Date**:

Método	Descrição
<u>getMilliseconds()</u>	Retorna os milissegundos da hora (de 0-999).
<u>getTime()</u>	Retorna o número de milissegundos desde a meia-noite de 1 de Janeiro de 1970.
<u>setDate()</u>	Define o dia do mês de um objeto Date.
<u>setFullYear()</u>	Define o ano (quarto dígitos) de um objeto Date.
<u>setHours()</u>	Define a hora de um objeto Date.
<u>setMilliseconds()</u>	Define os milissegundos de um objeto Date.
<u>setMinutes()</u>	Define os minutos de um objeto Date.
<u>setMonth()</u>	Define o mês de um objeto Date.
<u>setSeconds()</u>	Define os segundos de um objeto Date.
<u>toDateString()</u>	Converte a porção da data de um objeto Date em uma string que possa ser lida.
<u>toString()</u>	Converte um objeto Date em uma string.

Exercícios de fixação

- 1) Que método retorna um número representando o dia do mês, ou seja, de 1 a 31?
 - a) `getMonth()`
 - b) `getDate()`
 - c) `getWeek()`
 - d) `getDay()`

- 2) Qual das alternativas a seguir é a forma correta para se criar uma instância de um objeto **Date**?
 - a) `var minhaData = new Date{};`
 - b) `var minhaData = new Date[];`
 - c) `var minhaData = new Date;`
 - d) `var minhaData = new Date();`

- 3) O método _____ retorna um número representando o dia da semana, ou seja, de 0 (para Domingo) a 6 (para Sábado).
 - a) `getMonth()`
 - b) `getDate()`
 - c) `getWeek()`
 - d) `getDay()`

- 4) O método **getMinutes()** retorna um número representando os minutos, entre 0 e 59. Essa declaração é:
 - a) Verdadeira
 - b) Falsa

- 5) O método **getSeconds()** retorna um número representando os segundos, entre 1 e 60. Essa declaração é:
 - a) Verdadeira
 - b) Falsa

- 6) O método _____ retorna um número representando a hora do dia, ou seja, de 0 a 23.
 - a) `getHour()`
 - b) `getDate()`
 - c) `getWeek()`
 - d) `getDay()`

- 7) O método _____ retorna o ano atual com quatro dígitos.
 - a) `getFullDate()`
 - b) `getFullYear()`
 - c) `getYear()`
 - d) `getDate()`

- 8) O método _____ define os segundos de um objeto **Date**.
 - a) `setFullDate()`
 - b) `getSeconds()`
 - c) `setSeconds()`
 - d) `setDate()`

9) O método _____ converte um objeto **Date** em uma string.

- a) setDateToString()
- b) toDateString()
- c) toDate()
- d) toString()

10) Qual o valor da variável **anoAtual** após o código ser executado?

```
var minhaData = new Date();  
var anoAtual = minhaData.getFullYear() - 53;
```

- a) 1953
- b) 1960
- c) 2053
- d) 2023

JORGE EIDER

Exercícios propostos

- 1) Escreva uma rotina utilizando os dias da semana, e para cada dia mostre uma mensagem diferente em uma caixa de diálogo.
- 2) Escreva uma rotina que mostre em uma caixa de diálogo somente os meses com 31 dias.
- 3) Crie uma rotina que mostre três horas diferentes do dia, sendo uma para cada parte do dia, por exemplo, manhã, tarde e noite, e mostre uma mensagem para cada hora em uma caixa de diálogo.
- 4) Crie uma rotina para calcular o número de dias que faltam para a Copa do Mundo 2014.

JORGE EIDER