Programação Estruturada

Aula 08 - Operadores Javascript: Datas

Videoaula 03: Métodos de datas

Videoaula 03: Métodos de datas

Assim como para *strings*, JavaScript oferece diversos métodos para objetos de data. Alguns deles permitem obter informações e outros permitem alterar informações do objeto. Nesta videoaula, você conhecerá alguns desses métodos. Caso você deseje conhecer todos os métodos disponíveis, acesse a especificação de JavaScript através do QR Code que apresento agora no slide.

Saiba Mais

Acesse o link abaixo ou o QR Code para conhecer todos os metodos disponíveis:

https://www.ecma-international.org/ecma-262/10.0/index.html#sec-date-objects

Vamos então começar pelos métodos que nos permitem obter informações dos objetos de data. A maioria começa com o prefixo *get*, que em português significa obter. Os métodos `getFullYear()`, `getMonth()`, `getDate()`, retornam, respectivamente, o ano com quatro dígitos, o mês e o dia da data. É importante lembrar que o mês é armazenado com um número de 0 a 11, e não como de 1 a 12, como estamos acostumados, ou seja, temos o mês de janeiro representado como 0, o mês de fevereiro representado como 1, e assim por diante, até o mês de dezembro, que é representado como 11.

Além desses métodos, temos os métodos `getHours()`, `getMinutes()`, `getSeconds()` e `getMilliseconds()`, que retornam, respectivamente, as horas, os minutos, os segundos e os milissegundos do objeto. Outro método bastante útil é o método `getDay()`, que retorna um número entre 0 e 6 e representa o dia da semana da data. Em JavaScript, o 0 corresponderá ao domingo, o 1 corresponderá à segunda-feira, e assim por diante, até o 6, que corresponderá ao sábado. Por fim, temos o método `getTime()`, que retorna o número de milissegundos desde a época Unix, ou seja, desde a hora zero, GMT, de 01 de janeiro de 1970.

No slide, vemos o resultado que obtemos ao chamarmos esses sete métodos no objeto que corresponde à data de 20 de julho de 1969, às 23 horas, 56 minutos, 15 segundos e 960 milissegundos. Você saberia explicar por que o último resultado é um número negativo? Procure descobrir o porquê. Caso você não consiga, consulte os seus colegas ou o seu professor.

Figura 1 - Métodos de Data (get...)

```
var data = new Date(1969, 06, 20, 23, 56, 15, 960);
```

Método	Resultado	Observação
`data.getFullYear()`	1969	4 dígitos
`data.getMonth()`	6	0 (Janeiro) a 11 (Dezembro)
`data.getDate()`	20	1 a 31
`data.getHours()`	23	0 a 23
`data.getMinutes()`	56	0 a 59
`data.getSeconds()`	15	0 a 59
`data.getMilliseconds()`	960	0 a 999
`data.getDay()`	0	0 (domingo) a 6 (sábado)
`data.getTime()`	-14159024040	

JavaScript também oferece métodos que podem ser usados para trabalhar com datas UTC. Eles começam com o prefixo *getUTC*. Esses métodos retornam a mesma informação dos métodos do slide anterior, porém, no fuso horário UTC, ou, como sabemos, também chamado de GMT. Usando a mesma data do slide anterior como exemplo, e considerando que estamos no horário padrão de Brasília, ou seja, GMT-3, o que corresponde a três horas a menos que o horário padrão de Londres, o método `getUTCYear()` e `getUTCMonth()` também retorna 1969 e 6, porém, o método `getUTCDate()` retorna 21, e não 20, como usado na criação do objeto.

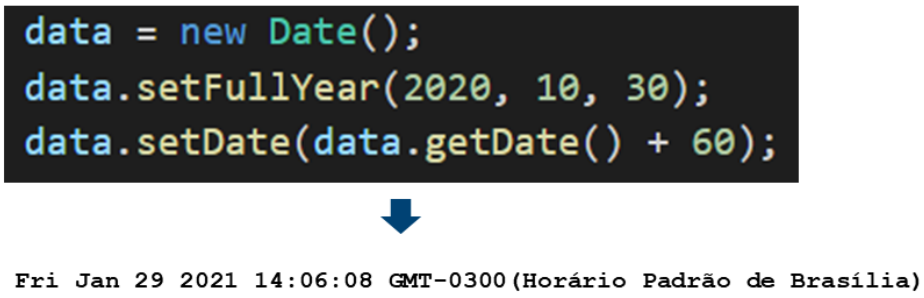
Isso porque esses métodos retornam as informações referentes ao objeto ao qual eles são aplicados, mas no fuso horário GMT. No exemplo, ao adicionarmos três horas à data 20 de julho de 1969, às 23 horas, 56 minutos, 15 segundos e 960 milissegundos, teremos 21 de julho de 1969, às 2 horas, 56 minutos, 15 segundos e 960 milissegundos. Pelo mesmo motivo, `getUTCHours()` retorna 2 e `getUTCDay()` retorna 1, ou seja, segunda-feira.

Em JavaScript, também podemos definir os valores de uma data usando métodos do objeto de data. Eles começam com o prefixo *set*. São eles: `setFullYear()`, `setMonth()`, `setDate()`, `setHours()`, `setMinutes()`, `setSeconds()`, `setMilliseconds()` e `setTime()`, os quais, usando os valores recebidos, alteram,

respectivamente, o ano, o mês, o dia do mês, as horas, os minutos, os segundos, os milissegundos, e o número de milissegundos, desde a época Unix. Opcionalmente, o método pode receber também o mês e o dia do mês.

Note que você pode usar esse método para adicionar ou remover quaisquer unidades de tempo do objeto data a uma data existente. Eventuais mudanças nas outras unidades de tempo que essa alteração possa causar serão tratadas automaticamente pelo JavaScript. Por exemplo, no slide temos um exemplo no qual criamos uma data com o momento atual, depois alteramos o ano, o mês e o dia para 2020, 10, ou seja, novembro, e 30. Após isso, adicionamos 60 dias ao dia dessa data.

Figura 2 - Métodos de Data (set...)

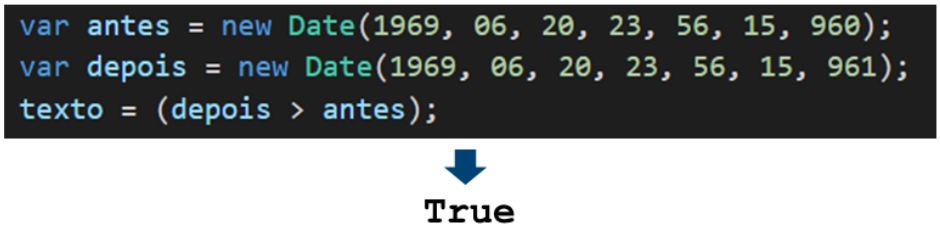


Note que a data final tem o horário em que o objeto foi criado, ou seja, 14 horas, 06 minutos e 08 segundos, o horário em que eu carreguei essa página. Porém, foram adicionados 60 dias a 30 de novembro de 2020, resultando no dia 29 de janeiro de 2021. Note que, ao adicionarmos 60 dias, JavaScript automaticamente tratou as mudanças não só de mês, mas também de ano. Você deve lembrar que já vimos esse comportamento quando passamos valores fora do intervalo no construtor de datas.

Como vimos anteriormente, em JavaScript, datas são armazenadas como números. Isso possibilita que comparações entre datas sejam feitas de maneira muito simples, usando os comparadores aritméticos. No exemplo que vemos no slide, temos duas datas que se diferenciam apenas nos milissegundos: a data `antes` tem 1 milissegundo a menos que a data `depois`. Nesse caso, podemos compará-las de maneira bem

simples, usando comparadores aritméticos como, por exemplo, o >. No slide, vemos que a expressão `depois > antes` resulta no valor booleano `true`, conforme esperamos que aconteça.

Figura 3 - Comparando Datas



Tudo bem até aqui? Todos os métodos que você conheceu nesta videoaula estão exemplificados no arquivo **08_5 Métodos de Data.html**. Fique à vontade para editar esse arquivo a fim de fazer vários testes.

Código 1 - 08_5 Métodos de Data.html

```
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Programação Estruturada - Aula 08</title>
  </head>
  <body>
    <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>

    <h1>Métodos de Data</h1>

    <p id="saida"></p>

    <script>

    var data = new Date(1969, 06, 20, 23, 56, 15, 960);

    // Métodos get
    var texto = data.getFullYear();
    //var texto = data.getMonth();
    //var texto = data.getDate();
    //var texto = data.getHours();
    //var texto = data.getMinutes();
    //var texto = data.getSeconds();
    //var texto = data.getMilliseconds();
```

```
25      //var texto = data.getDay();
26      //var texto = data.getTime();
27
28      // Métodos getUTC
29      //var texto = data.getUTCFullYear();
30      //var texto = data.getUTCMonth();
31      //var texto = data.getUTCDate();
32      //var texto = data.getUTCHours();
33      //var texto = data.getUTCMinutes();
34      //var texto = data.getUTCSeconds();
35      //var texto = data.getUTCMilliseconds();
36      //var texto = data.getUTCDay();
37
38      // Métodos set
39      //var data = new Date();
40      //data.setFullYear(2020, 10, 30);
41      //data.setDate(data.getDate() + 60);
42      //var texto = data;
43
44      // Comparando Datas
45      //var antes = new Date(1969, 06, 20, 23, 56, 15, 960);
46      //var depois = new Date(1969, 06, 20, 23, 56, 15, 961);
47      //texto = (depois > antes);
48
49      document.getElementById("saida").innerHTML = texto;
50
51      </script>
52    </body>
53  </html>
54
```

Para finalizar, reforço que você não esqueça também que a habilidade de programar só se adquire com muita prática. Por isso, não deixe de fazer os exercícios propostos. Até a próxima aula!