

Programa o Estruturada

Aula 08 - Operadores Javascript: Datas

Videoaula 02: Formatos de datas



Videoaula 02: Formatos de datas

Olá! Nesta videoaula, você conhecerá um pouco mais sobre os construtores de datas. De uma maneira geral, temos três formatos textuais de data que podem ser usados como entrada: Formato ISO, Data Curta e Data Longa.

O ISO 8601 é o padrão internacional para a representação de datas e tempos. A sintaxe ISO 8601 é o formato de datas favorito em JavaScript. Nele, temos quatro dígitos para o ano, seguido de um hífen, seguido de dois dígitos para o mês, seguido de outro hífen e, por fim, dois dígitos representando os dias. No slide, temos A, M e D, representando dígitos de ano, mês e dia, respectivamente. Por exemplo, 1969-07-20 representa 0 horas do dia 20 de julho de 1969. Novamente, a data calculada no seu computador dependerá do fuso horário de sua máquina. Por isso, o resultado para o nosso exemplo pode variar entre os dias 19 e 20 de julho. No horário padrão de Brasília, por ser três horas a menos que o GMT, teremos como resultado às 21 horas do dia 19 de julho de 1969.

Figura 1 - Formatos ISO

- **Formato ISO (Padrão Internacional):**

- **Formato AAAA-MM-DD:** `new Date("1969-07-20")`
`Sat Jul 19 1969 21:00:00 GMT-0300 (Horário Padrão de Brasília)`
- **Formato AAAA-MM:** `new Date("1969-07")`
`Mon Jun 30 1969 21:00:00 GMT-0300 (Horário Padrão de Brasília)`
- **Formato AAAA:** `new Date("1969")`
`Tue Dec 31 1968 21:00:00 GMT-0300 (Horário Padrão de Brasília)`

As datas ISO podem ser escritas sem o dia, ou, também, sem o dia e o mês. Quando não especificarmos o dia, será considerado o primeiro dia do mês. No horário padrão de Brasília, sempre teremos como resultado o último dia do mês anterior. No nosso exemplo, teremos o dia 30 de junho de 1969. Além disso, quando não usarmos o mês, será considerado o primeiro dia do primeiro mês do ano. No horário padrão de Brasília, sempre teremos como resultado o último dia do ano anterior. Nesse caso, teremos o dia 31 de dezembro de 1968.

A maneira mais completa de especificar datas ISO é adicionando horas, minutos, segundos e fuso horário. Para isso, você deverá usar um T maiúsculo entre o dia e a hora, onde teremos dois dígitos para as horas, seguido de dois pontos, seguidos de dois dígitos para os minutos, seguidos de outros dois pontos e seguidos de dois dígitos para os segundos. Finalmente, temos a indicação do fuso horário.

Pode-se usar um Z maiúsculo para indicar o fuso horário GMT, ou +HH:MM, ou -HH:MM para os outros fusos horários. No slide, temos H, M e S, representando dígitos de horas, minutos e segundos, respectivamente. Por exemplo, 1969-07-20T23:56:15Z, representa as 23 horas, 56 minutos e 15 segundos do dia 20 de julho de 1969 no fuso horário GMT, ou seja, no fuso horário padrão de Londres. Por isso, se tentarmos exibir essa data no horário padrão de Brasília, teremos a exibição de três horas antes disso, ou seja, 20 horas, 56 minutos e 15 segundos.

Figura 2 - Formatos ISO com Fuso Horário

- Formato ISO (Padrão Internacional):
 - Formato AAAA-MM-DDTHH:MM:SSZ: `new Date("1969-07-20T23:56:15Z")`
`Sun Jul 20 1969 20:56:15 GMT-0300 (Horário Padrão de Brasília)`
 - Formato AAAA-MM-DDTHH:MM:SSZ: `new Date("1969-07-20T23:56:15-03:00")`
`Sun Jul 20 1969 23:56:15 GMT-0300 (Horário Padrão de Brasília)`

Para representar esse horário no fuso horário padrão de Brasília, o qual é o horário correto em que Neil Armstrong pisou na lua, precisamos substituir o Z por -03:00 no final da *string*, como podemos ver no slide. A omissão do T ou do Z trará resultados diferentes em navegadores diferentes.

Conforme falei há pouco, o segundo formato textual de data que pode ser usado como entrada é Data Curta, que tem o formato MM/DD/AAAA. Por exemplo, como podemos ver no slide, ao criarmos uma data passando 07/20/1969, temos como resultado exatamente esse dia. É muito importante seguir estritamente esse padrão. Caso você esqueça de colocar um zero antes de numerais menores, ou caso coloque o ano antes do mês, ou use hífen ao invés da barra, o comportamento dependerá do navegador que está sendo usado. Alguns navegadores produzirão o resultado esperado, mas outros produzirão erros ou o valor NaN.

Figura 3 - Datas Curtas

- Formato MM/DD/AAAA: `new Date("07/20/1969")`

`Sun Jul 20 1969 20:56:15 GMT-0300 (Horário Padrão de Brasília)`

- Comportamento Inesperado
 - `new Date("7/20/1969")`
 - `new Date("1969/07/20")`
 - `new Date("07-20-1969")`

Agora, vou falar sobre o terceiro e último formato textual de data que pode ser usado como entrada, a data longa. Geralmente, usa-se o formato MMM DD AA. Por exemplo, como podemos ver no slide, podemos passar a *string* "Jul 20 1969" para criarmos a data com o dia da chegada do homem à lua. Note que a hora, nesse caso, fica como 0 hora. No entanto, esse formato possui diversas opções: meses e dias podem ser colocados em qualquer ordem; os nomes dos meses podem ser escritos por completo, as letras podem ser maiúsculas ou minúsculas e eventuais vírgulas podem ser inseridas, mas serão ignoradas.

Figura 4 - Datas Longas

- Formato MMM DD AAAA
 - `new Date("Jul 20 1969")`
- Opções
 - `new Date("20 Jul 1969")`
 - `new Date("20 July 1969")`
 - `new Date("20 JULY 1969")`
 - `new Date("20 JULY, 1969")`

`Sun Jul 20 1969 00:00:00 GMT-0300 (Horário Padrão de Brasília)`

Caso você tenha uma *string* com um formato textual de data válido, também poderá obter diretamente os milissegundos que representam essa data usando o método `Date.parse(s)`. Como você viu anteriormente, esse valor é igual à quantidade de milissegundos que se passaram, até a data que está sendo representada, a partir da zero-hora GMT do dia 1º de janeiro de 1970. Usando o construtor `Date()`, com esse valor como parâmetro, como já vimos nesta aula, podemos criar o objeto data correspondente. Além disso, como veremos na próxima videoaula, podemos chamar vários métodos nos objetos do tipo `Date`. Dentre eles, o método `getTime`, que retorna exatamente essa quantidade de milissegundos.

Você pode utilizar o arquivo [08_4 Formatos de Data.html](#) para verificar o comportamento do construtor de datas com os vários formatos textuais de data que mencionei nesta videoaula.

Código 1 - 08_4 Formatos de Data.html

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

```
9 <h1>Formatos de Data</h1>
10
11 <p id="saida"></p>
12
13 <script>
14
15 // Formatos ISO
16 var data = new Date("1969-07-20");
17 var data = new Date("1969-07");
18 var data = new Date("1969");
19 var data = new Date("1969-07-20T23:56:15Z");
20 var data = new Date("1969-07-20T23:56:15-03:00");
21
22 // Data Curta
23 var data = new Date("07/20/1969");
24
25 // Data Longa
26 var data = new Date("Jul 20 1969");
27 var data = new Date("20 Jul 1969");
28 var data = new Date("20 July 1969");
29 var data = new Date("20 JULY 1969");
30 var data = new Date("20 JULY, 1969");
31
32 // Date.parse()
33 var ms = Date.parse("Jul 20 1969");
34 var data = new Date(ms);
35
36 document.getElementById("saida").innerHTML = data;
37
38 </script>
39 </body>
40 </html>
41
```

Na próxima videoaula, você conhecerá os vários métodos que podem ser chamados a partir dos objetos do tipo `Date()`.