**Para saber mais: Objeto Date**

Nesta aula, aprendemos como utilizar o objeto [Date](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Date) para formatar minutos e segundos. Entretanto, é possível utilizar suas propriedades e métodos para mais funcionalidades, como manipular e exibir datas e horas no seu código.

Vamos conferir mais formas de seu uso?

Para criar uma instância do objeto ‘Date’, você pode usar uma das várias formas de construção:

* Construtor sem argumentos:

**const** dataAtual = **new** **Date**();

* Construtor com argumentos (ano, mês, dia, hora, minuto, segundo, milissegundo):

**const** dataEspecifica = **new** **Date**(2023, 7, 3, 12, 30, 0, 0);

* Construtor com uma string que representa a data (formato padrão é "yyyy-mm-dd"):

**const** dateString = "2023-08-03";

**const** formatoDeData = **new** **Date**(dateString);

Uma vez criado um objeto **Date**, você pode acessar informações específicas da data e hora, como o ano, mês, dia, hora, minuto e segundo usando os métodos de acesso do objeto:

**const** currentDate = **new** **Date**();

**const** ano = currentDate.getFullYear(); // Acessa o ano

**const** mês = currentDate.getMonth(); // Acessa o mês - Janeiro é 0, Fevereiro é 1, ..., Dezembro é 11

**const** dia = currentDate.getDate(); // Acessa o dia

**const** horas = currentDate.getHours(); // Acessa as horas

**const** minutos = currentDate.getMinutes(); // Acessa os minutos

**const** segundos = currentDate.getSeconds(); // Acessa os segundos

**const** milissegundos = currentDate.getMilliseconds(); // Acessa os milissegundos

Você também pode modificar a data e hora usando os métodos de definição:

**const** **data** = new Date();

**data**.setFullYear(2024); // Define o ano

**data**.setMonth(10); // Define o mês

**data**.setDate(25); // Define o dia

**data**.setHours(10); // Define as horas

**data**.setMinutes(30); // Define os minutos

**data**.setSeconds(0); // Define os segundos

Além disso, o objeto ‘Date’ também fornece vários métodos para trabalhar com datas, como comparar datas, adicionar ou subtrair períodos de tempo, obter o dia da semana, entre outros.

Lembre-se de que, dependendo do fuso horário e configurações regionais do sistema em que o código está sendo executado, os resultados podem variar. Se você precisar de mais funcionalidades e suportes a diferentes formatos de zonas de tempo, uma boa alternativa é o uso de bibliotecas de manipulação de datas, como o Moment.js ou o date-fns.