mdn web docs _

This page was translated from English by the community. Learn more and join the MDN Web Docs community.

Date

Resumo

Cria uma instância JavaScript de `Date` que representa um único momento no tempo. Objetos Date são baseados no valor de tempo que é o número de milisegundos desde 1º de Janeiro de 1970 (UTC).

Construtor

```
new Date();
new Date(valor);
new Date(dataString);
new Date(ano, mês, dia, hora, minuto, segundo, milissegundo);
```

`Date` `Date` new `Date` `Date`

null

Parâmetros para o constructor Date

Nota: Quando Date for chamado como um construtor com mais de um argumento, se os valores forem maiores do que seu limite lógico (e.g. se 13 for fornecido como um valor para mês ou 70 for o valor para minuto), o valor adjacente será ajustado. E.g. new Date(2013, 13, 1) é equivalente a new Date(2014, 1, 1), ambos criam uma data para 2014-02-01 (note que o mês começa em 0). Similarmente para outros valores: new Date(2013, 2, 1, 0, 70) é equivalente a new Date(2013, 2, 1, 1, 10), pois ambos criam uma data para 2013-03-01T01:10:00.

`value`

Um valor inteiro representando o número de milisegundos desde 1 de Janeiro de 1970 00:00:00 UTC (Era Unix ou Marco Zero).

`dataString`

Um valor do tipo String que representa uma data. A string deverá estar uma formato reconhecido pelo método `Date.parse()` (IETF-compliant RFC 2822 timestamps e também uma versão da ISO8601).

`year`

Um valor inteiro que representa o ano. Valores de 0 a 99 correspondem aos anos de 1900 a 1999. Veja o exemplo abaixo.

`month`

Um valor inteiro que representa o mês, começando com 0 para Janeiro até 11 para Dezembro.

`day`

Um valor inteiro que representa o dia do mês.

`hour`

Um valor inteiro que representa a hora do dia.

`minute`

Um valor inteiro que representa o segmento de um minuto de tempo.

`second`

Um valor inteiro que representa o segmento de segundo do tempo.

`millisecond`

Um valor inteiro que representa o segmento de milisegundo do tempo.

Descrição

- Se nenhum argumento for fornecido, o construtor criará um objeto JavaScript Date com a data e hora corrente de acordo com as configurações do sistema.
- Se ao menos 2 argumentos forem fornecidos, os argumentos ausentes serão configurados como 1 (se o dia estiver ausente) ou 0 para todos os outros.
- A data do JavaScript é baseada no valor de tempo em milisegundos desde a meia noite de 01 de Janeiro de 1970, UTC. Um dia corresponde a 86.400,000 milisegundos. O intervalo do objeto Date no JavaScript é de -100.000,000 dias a 100.000,000 dias relativo a 01 de Janeiro de 1970, UTC.

- O objeto Date no JavaScript tem um comportamento uniforme nas plataformas. O valor do tempo pode ser transmitido entre sistemas para representar o mesmo instante no tempo e se for usado para criar um objeto de data local, ele refletirá o tempo local equivalente.
- O objeto Date JavaScript suporta vários métodos UTC (universal), assim como métodos de tempo locais. UTC, também conhecido como Tempo Médio de Greenwich (Greenwich Mean Time, GMT), refere-se ao tempo como definido pelo Padrão de Tempo Mundial (World Time Standard). O tempo local é o tempo conhecido pelo computador onde o JavaScript é executado.
- Invocar o objeto Date no JavaScript como uma função (i.e., sem o operador new) retornatá uma string representando a data e hora corrente.

Propriedades

`Date.prototype` **(en-US)**

Permite adicionar propriedades a um objeto javaScript Date.

Date.length

O valor de `Date.length` é 7. Esse é o número de argumentos manipulados pelo construtor.

> Properties inherited from `Function`:
> `arity`, `caller`, constructor (en-US), `length`, `name`

Métodos

`Date.now()`

Retorna o valor numérico correspondente ao tempo corrente - o número de milisegundos passados desde 1 de Janeiro de 1970 00:00:00 UTC.

`Date.parse()`

Analisa uma string que representa uma data e retorna o número de milisegundos desde 1 de Janeiro, 1970, 00:00:00, hora local.

`Date.UTC()`

Aceita os mesmos parâmetros como a forma mais longa do construtor (i.e. 2 até 7) e retorna o número de milisegundos desde 1 de Janeiro, 1970, 00:00:00 UTC.

> Methods inherited from `Function`:
> `apply`, `call`, `toSource`, `toString`

Instâncias JavaScript de `Date`

Todas as instâncias `Date` são herdadas de `Date.prototype` (en-US). O objeto protótipo do construtor `Date` pode ser modificado para afetar todas as instâncias de `Date`.

Métodos

{{ page("/en-US/docs/JavaScript/Reference/Global_Objects/Date/prototype", "Methods") }}

Exemplos

Várias formas de se criar um objeto Date

Os seguintes exemplos mostram várias formas de se criar datas em JavaScript:

> **Nota:** a conversão de *strings* com o construtor de `Date` (`Date.parse` é equivalente ao contrutor) é fortemente desencorajada devido às inconsistências e diferenças dos navegadores.

```
var today = new Date();
var birthday = new Date("December 17, 1995 03:24:00");
var birthday = new Date("1995-12-17T03:24:00");
var birthday = new Date(1995,11,17);
var birthday = new Date(1995,11,17,3,24,0);
```

Anos com dois dígitos mapeados para 1900 - 1999

Para criar e obter datas entre os anos 0 e 99 os métodos `Date.prototype.setFullYear()` e `Date.prototype.getFullYear()` devem ser usados.

```
var data = new Date(98, 1); // Dom Fev 01 1998 00:00:00 GMT+0000 (GMT)

// Métodos em desuso, 98 mapeia para 1998 aqui também
data.setYear(98);           // Dom Fev 01 1998 00:00:00 GMT+0000 (GMT)

data.setFullYear(98);       // Sab Fev 01 0098 00:00:00 GMT+0000 (BST)
```

Calculando o tempo decorrido

Os seguintes exemplos mostram como determinar o tempo decorrido entre duas datas no JavaScript em milissegundos.

Devido aos tamanhos diferentes dos dias (devido à mudança do horário de verão), meses e dias, expressar o tempo decorrido em unidades maiores que horas, minutos e segundos requer analisar os problemas e deve ser cuidadosamente investigado antes de se tentar utilizar.

```
// usando objetos Date
var inicio = Date.now();

// o evento para o tempo vai aqui:
facaAlgoPorUmLongoTempo();
var fim = Date.now();
var decorrido = fim - inicio; // tempo decorrido em milisegundos
```

```
// utilizando métodos embutidos
var inicio = new Date();

// o evento para o tempo vai aqui:
facaAlgoPorUmLongoTempo();
var fim = new Date();
var decorrido = fim.getTime() - inicio.getTime(); // tempo decorrido em milisegundos
```

```
// para testar uma função e obter o seu retorno
function imprimirTempoDecorrido(fTeste) {
        var nHoraInicial = Date.now(),
        vRetorno = fTeste(),
        nHoraFinal = Date.now();
```

```
        alert("Tempo decorrido: " + String(nHoraFinal - nHoraInicial) + " milisegundos");
        return vRetorno;
}

retornoDaSuaFuncao = imprimirTempoDecorrido(suaFuncao);
```

> Nota: Em navegadores que suportam a API de Desempenho Web (Web Performance API) com o recurso de tempo de alta resolução, `Performance.now()` pode fornecer medidas de tempo decorrido mais confiáveis e precisas do que `Date.now()`.

Especificações

Especificação	Estado	Comentário
ECMAScript (ECMA-262) The definition of 'Date' in that specification.	Padrão em tempo real	
ECMAScript 2015 (6th Edition, ECMA-262) The definition of 'Date' in that specification.	Padrão	
ECMAScript 5.1 (ECMA-262) The definition of 'Date' in that specification.	Padrão	
ECMAScript 1st Edition (ECMA-262)	Padrão	Definição inicial. Implementado no JavaScript 1.1.

Compatibilidade com navegadores

> **Estamos convertendo nossos dados de compatibilidade para o formato JSON**. Esta tabela de compatibilidade ainda usa o formato antigo, pois ainda não convertemos os dados que ela contém. **Descubra como você pode ajudar! (en-US)**

- Desktop
- Dispositivo móvel

Recurso	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari
Suporte Básico	(Yes) [1]	(Yes) [1]	(Yes) [2]	(Yes) [1]	(Yes) [1]

Recurso	Android	Chrome for Android	Firefox Mobile (Gecko)	IE Mobile	Opera Mobile	Safari Mobile
Suporte Básico	(Yes)	(Yes)	(Yes)	(Yes)	(Yes)	(Yes)

[1] Alguns navegadores pode ter problemas quando verificam datas: 3/14/2012 blog from danvk Comparing FF/IE/Chrome on Parsing Date Strings

[2] ISO8601 Date Format não é suportado o Internet Explorer 8 e outras versões podem ter problemas quando convertem datas

Last modified: 2 de ago. de 2021, by MDN contributors