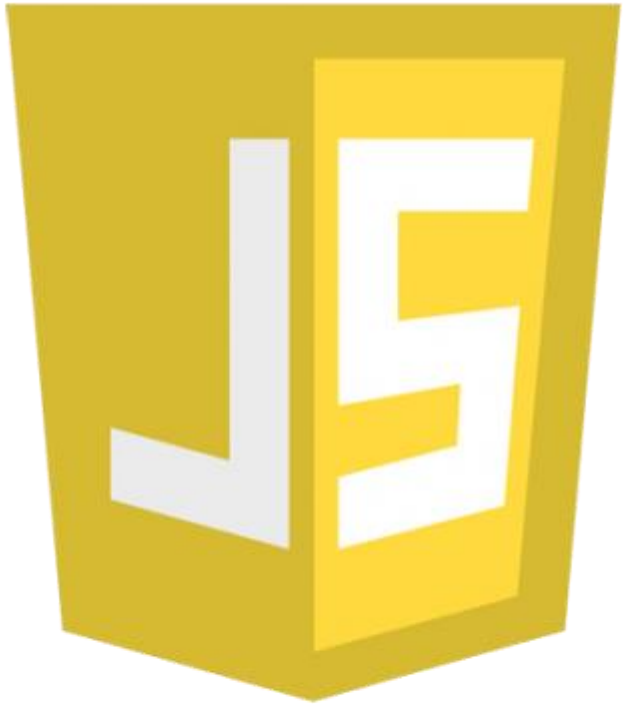
RECODE pro



JavaScript

Trabalhando com funções do Javascript (String, Math, Date)

Flávio Mota

RECODE pro

Métodos e propriedades de Strings

Existem várias funções disponíveis para manipular strings, conhecer estas funções é muito importante, saber manipular strings nos abre uma série de possibilidades.
Neste capítulo vamos aprender algumas funções para manipular strings, faça muito proveito e use sua criatividade para ver as possibilidades de cada método.

Propriedade Length: devolve o tamanho da string

```
var txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
var sln = txt.length;  // Saida = 26
```

O método indexOf () retorna o índice (da posição) da primeira ocorrência de um texto especificado em uma string

```
var str = "Please locate where 'locate' occurs!";
var pos = str.indexOf("locate"); // Saida = 7
```

Fonte: https://www.w3schools.com/js/js_string_methods.asp

Métodos e propriedades de Strings

Extraindo partes de Strings

Existem 3 métodos para extrair uma parte de uma sequência:

`Slice, substring, substr`

O Método Slice()

extrai uma parte de uma sequência e retorna a parte extraída em uma nova sequência. O método utiliza 2 parâmetros: a posição inicial e a posição final (final não incluído). Este exemplo corta uma parte de uma sequência da posição 7 para a posição 12 (13-1)

Example

```
var str = "Apple, Banana, Kiwi";
var res = str.slice(7, 13);
```

The result of res will be:

```
Banana
```

Fonte: https://www.w3schools.com/js/js_string_methods.asp

Métodos e propriedades de Strings

Extraindo partes de Strings

O Método substring()

A diferença é que substring () não pode aceitar índices negativos.

Example

```
var str = "Apple, Banana, Kiwi";
var res = str.substring(7, 13);
```

The result of *res* will be:

```
Banana
```

Fonte: https://www.w3schools.com/js/js_string_methods.asp

Métodos e propriedades de Strings

Extraindo partes de Strings

O Método substr()

substr () é semelhante a slice (). A diferença é que o segundo parâmetro especifica o tamanho da string extraída.

Example

```
var str = "Apple, Banana, Kiwi";
var res = str.substr(7, 6);
```

The result of res will be:

```
Banana
```

Fonte: https://www.w3schools.com/js/js_string_methods.asp

Métodos e propriedades de Strings

Substituindo o conteúdo da string

O Método replace()

Substituindo o conteúdo da string. O método replace () substitui um valor especificado por outro valor em uma sequência:

Example

```
str = "Please visit Microsoft!";
var n = str.replace("Microsoft", "W3Schools");
```

A string n vai receber uma nova string onde a palavra “Microsoft” será substuitda por “W3Schools”, O método replace () não altera a string em que é chamada. Retorna uma nova string.

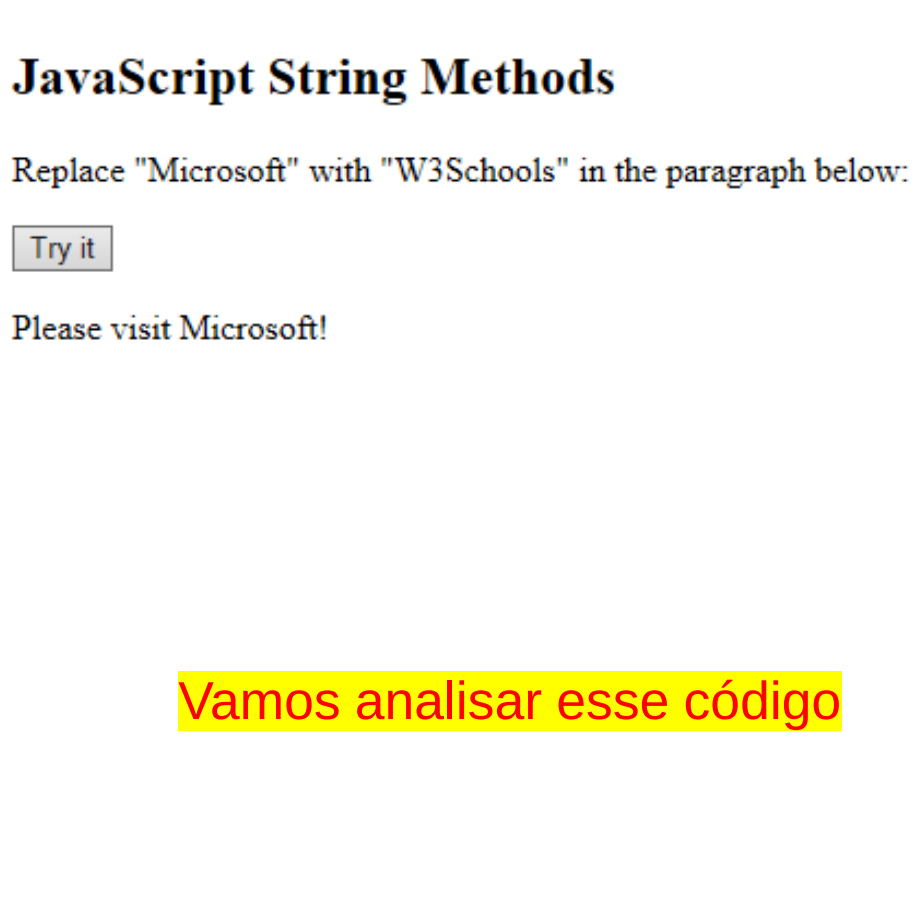
Fonte: https://www.w3schools.com/js/js_string_methods.asp

Métodos e propriedades de Strings

Substituindo o conteúdo da string

Exemplo Método replace()

```
<!DOCTYPE html>
<html>

<body>

<h2>JavaScript String Methods</h2>

<p>Replace "Microsoft" with "W3Schools" in the paragraph below:</p>

<button onclick="myFunction()">Try it</button>

<p id="demo">Please visit Microsoft!</p>

<script>
function myFunction() {
  var str = document.getElementById("demo").innerHTML;
  var txt = str.replace("Microsoft","W3Schools");
  document.getElementById("demo").innerHTML = txt;
}
</script>

</body>
</html>
```



Vamos analisar esse código

Fonte: https://www.w3schools.com/js/js_string_methods.asp

Métodos e propriedades de Strings

Convertendo para Maiúsculas e Minúsculas

Método toUpperCase()

Uma string é convertida para maiúscula com o método toUpperCase()

```
var text1 = "Hello World!";       // String
var text2 = text1.toUpperCase();  // text2 é o text1 convertido para maisculo
// Saida “HELLO WORLD!”
```

Fonte: https://www.w3schools.com/js/js_string_methods.asp

Métodos e propriedades de Strings

Convertendo para Maiúsculas e Minúsculas

Método toLowerCase()

Uma string é convertida para minúscula com o método toLowerCase()

```
var text1 = "Hello World!";       // String
var text2 = text1.toLowerCase();  // text2 é o text1 convertido para minúsculo
// Saida “hello world!”
```

Fonte: https://www.w3schools.com/js/js_string_methods.asp

Métodos e propriedades de Strings

Método split()

Dividir uma string em uma matriz de substrings

Example

Split a string into an array of substrings:

```
var str = "How are you doing today?";
var res = str.split(" ");
```

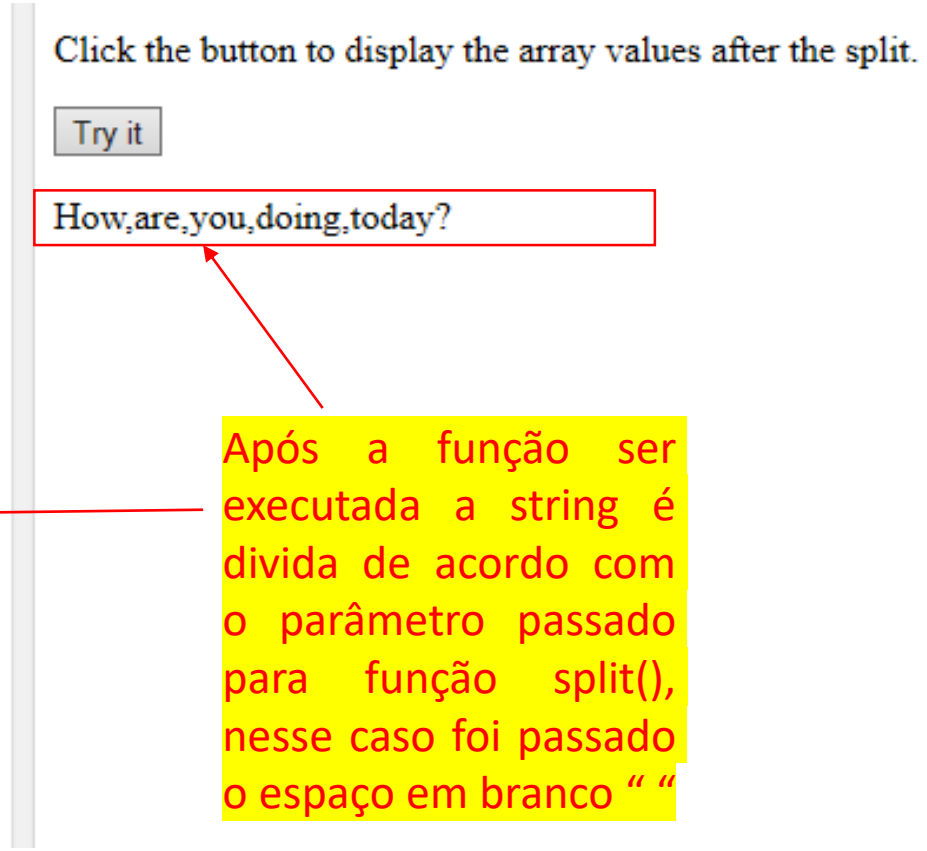
Definição e Uso O método split () é usado para dividir uma string em uma matriz de substrings e retorna a nova matriz. Dica: Se uma sequência vazia ("") for usada como separador, a sequência será dividida entre cada caractere.

Fonte: https://www.w3schools.com/js/js_string_methods.asp

Métodos e propriedades de Strings

Exemplo Método split()

```
<!DOCTYPE html>
<html>
<body>

<p>Click the button to display the array values after the split.</p>

<button onclick="myFunction()">Try it</button>

<p id="demo"></p>

<script>
function myFunction() {
  var str = "How are you doing today?";
  var res = str.split(" ");
  document.getElementById("demo").innerHTML = res;
}
</script>

</body>
</html>
```



Após a função ser executada a string é divida de acordo com o parâmetro passado para função split(), nesse caso foi passado o espaço em branco “ “

Fonte: https://www.w3schools.com/js/js_string_methods.asp

RECODE pro

Desafio 01

Criar uma interface de cadastro de livros e uma saída de referência bibliográfica no padrão da ABNT.

O formulário deve conter os campos:

- nome do autor: **`Eça de Queiróz`**
- título: **`O primo Basílio`**
- número da edição: **`25`**
- local de publicação: **`Rio de Janeiro`**
- nome da editora: **`Ediouro`**
- ano de publicação: **`1878`**

Por fim, o programa deve apresentar a seguinte saída:

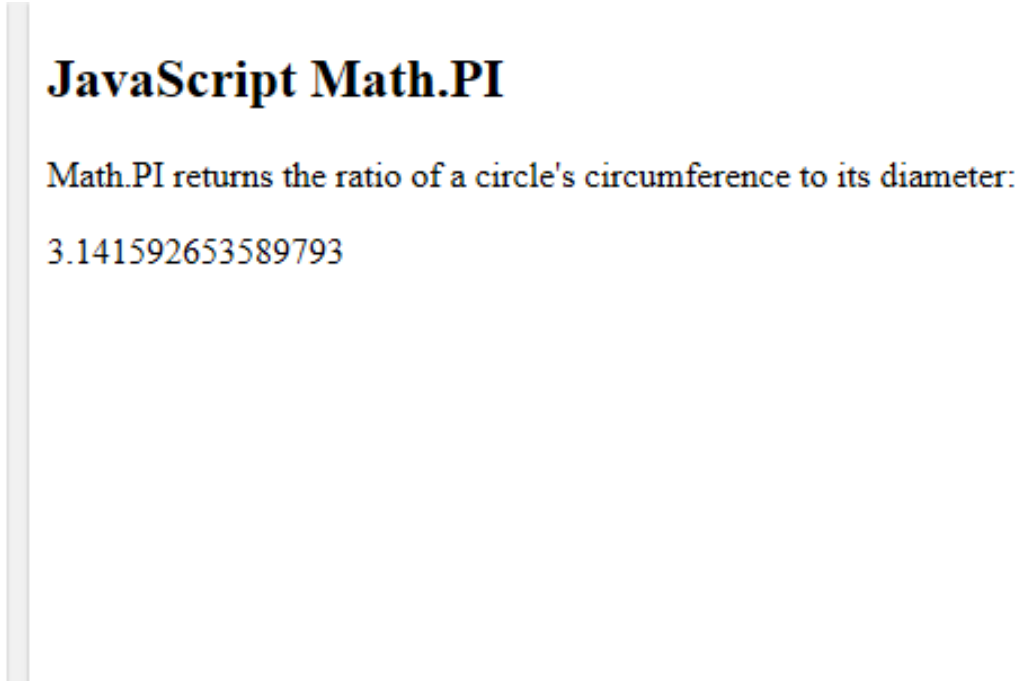
```
QUEIRÓZ, E. O Primo Basílio: 25. ed. Rio de Janeiro: Ediouro, 1878
```

Objeto Math

O objeto JavaScript Math permite executar tarefas matemáticas em números.

```
Math.PI;            // returns 3.141592653589793
```

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Math.PI</h2>

<p>Math.PI returns the ratio of a circle's circumference to its diameter:</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = Math.PI;
</script>

</body>
</html>
```



Fonte: https://www.w3schools.com/js/js_math.asp

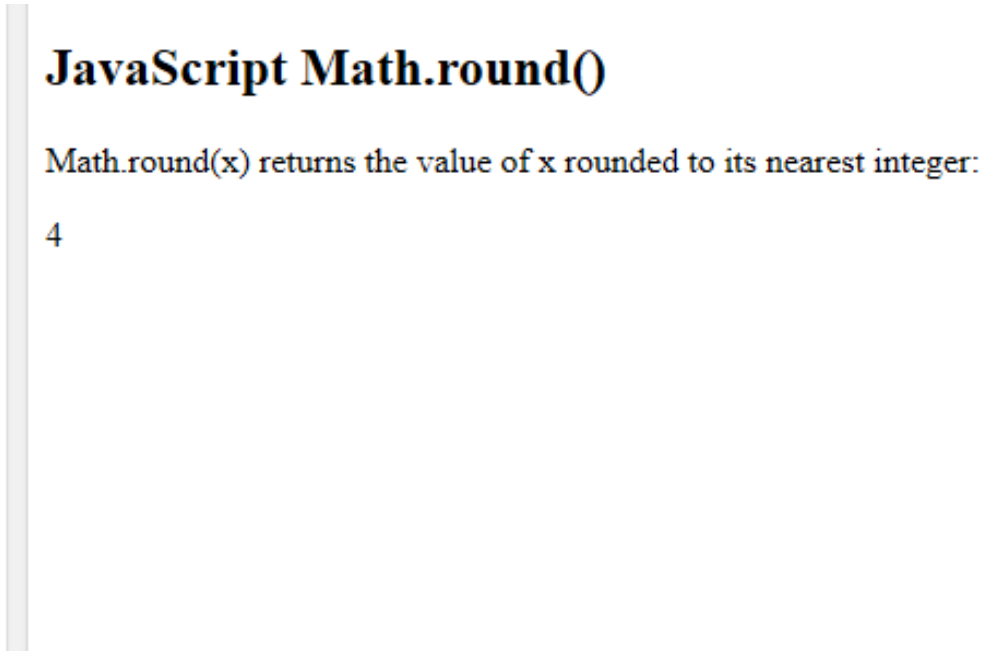
Objeto Math

O objeto JavaScript Math permite executar tarefas matemáticas em números.

Math.round (x) retorna o valor de x arredondado para o número inteiro mais próximo:

```
Math.round(4.7);     // returns 5
Math.round(4.4);     // returns 4
```

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Math.round()</h2>

<p>Math.round(x) returns the value of x rounded to its nearest integer:</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = Math.round(4.4);
</script>

</body>
</html>
```



Fonte: https://www.w3schools.com/js/js_math.asp

RECODE pro

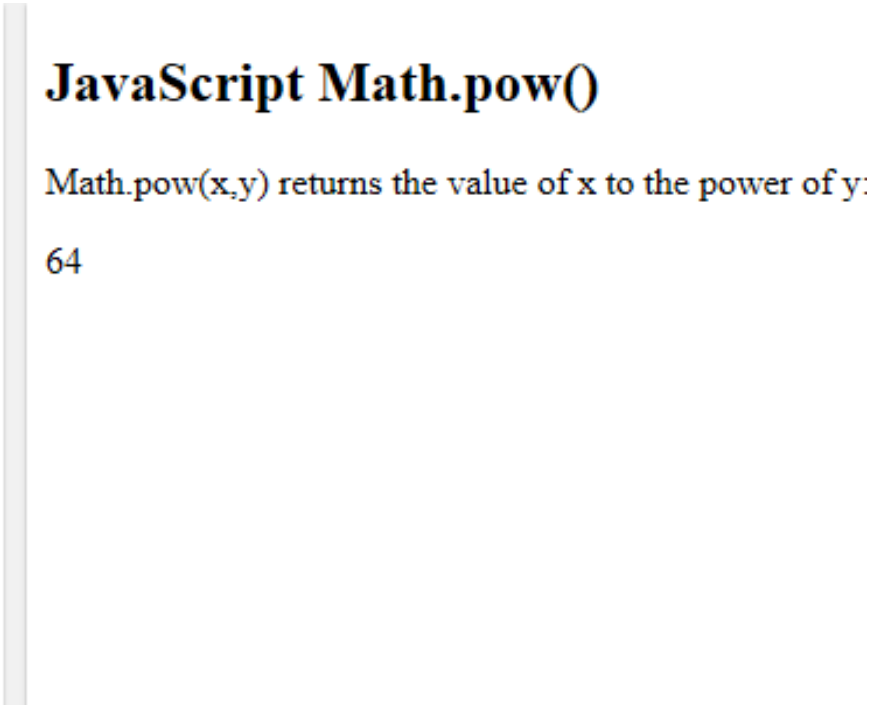
Objeto Math

O objeto JavaScript Math permite executar tarefas matemáticas em números.

Math.pow (x, y) retorna o valor de x elevado a y:

```
Math.pow(8, 2);       // returns 64
```

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Math.pow()</h2>

<p>Math.pow(x,y) returns the value of x to the power of y:</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = Math.pow(8,2);
</script>

</body>
</html>
```



Fonte: https://www.w3schools.com/js/js_math.asp

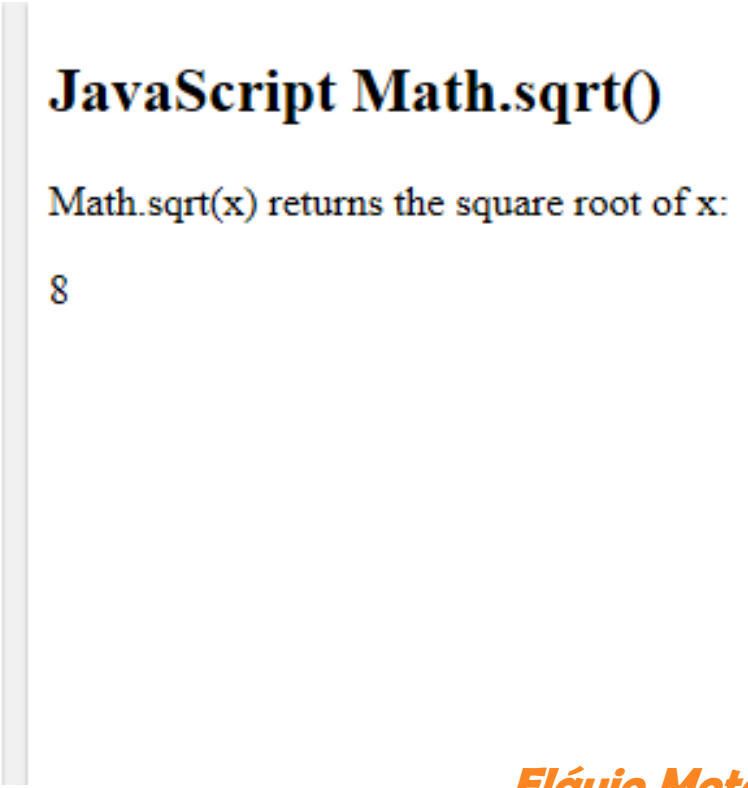
Objeto Math

O objeto JavaScript Math permite executar tarefas matemáticas em números.

Math.sqrt (x) retorna a raiz quadrada de x:

```
Math.sqrt(64);      // returns 8
```

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Math.sqrt()</h2>

<p>Math.sqrt(x) returns the square root of x:</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = Math.sqrt(64);
</script>

</body>
</html>
```



Fonte: https://www.w3schools.com/js/js_math.asp

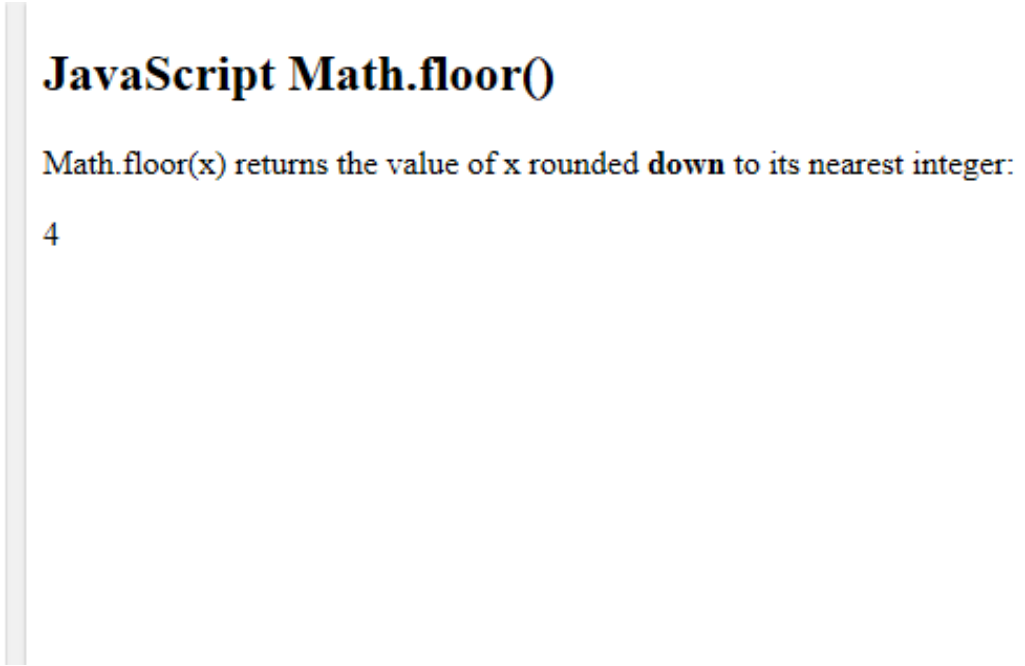
Objeto Math

O objeto JavaScript Math permite executar tarefas matemáticas em números.

Math.floor (x) retorna o valor de x arredondado para baixo o número inteiro mais próximo:

```
Math.floor(4.7);     // returns 4
```

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Math.floor()</h2>

<p>Math.floor(x) returns the value of x rounded <strong>down</strong> to its
nearest integer:</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = Math.floor(4.7);
</script>

</body>
</html>
```



Fonte: https://www.w3schools.com/js/js_math.asp

RECODE pro

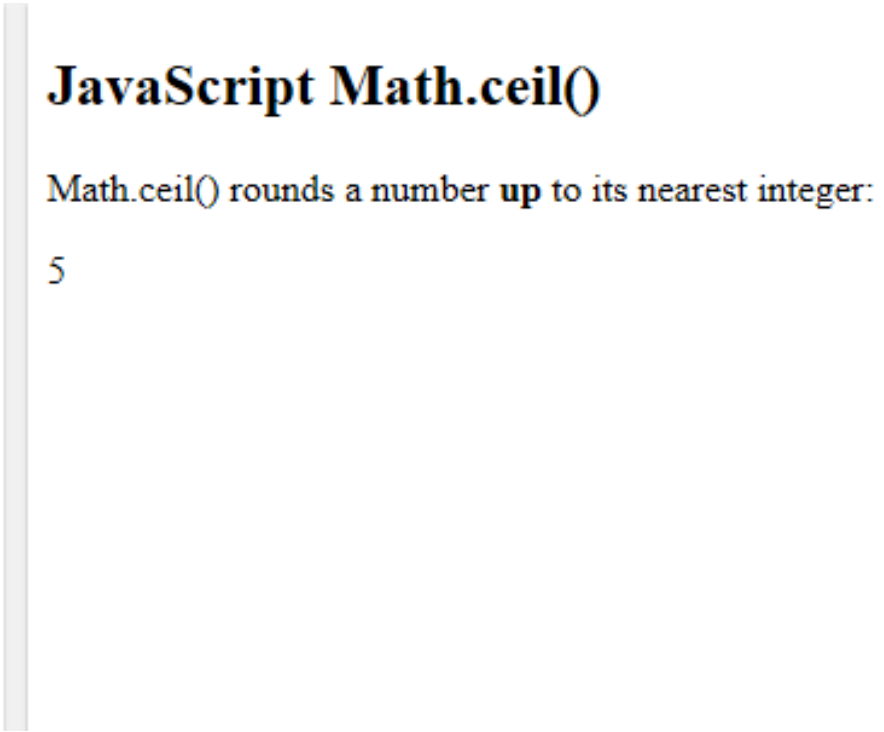
Objeto Math

O objeto JavaScript Math permite executar tarefas matemáticas em números.

Math.ceil (x) retorna o valor de x arredondado para cima o número inteiro mais próximo:

```
Math.ceil(4.4);     // returns 5
```

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Math.ceil()</h2>

<p>Math.ceil() rounds a number <strong>up</strong> to its nearest integer:</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = Math.ceil(4.4);
</script>

</body>
</html>
```



Fonte: https://www.w3schools.com/js/js_math.asp

RECODE pro

Desafio 02

Crie um formulário com botões e inputs para realizar no mínimo 5 operações usando funções do próprio Javascript, os resultados devem ser apresentados em containers do tipo paragrafo.

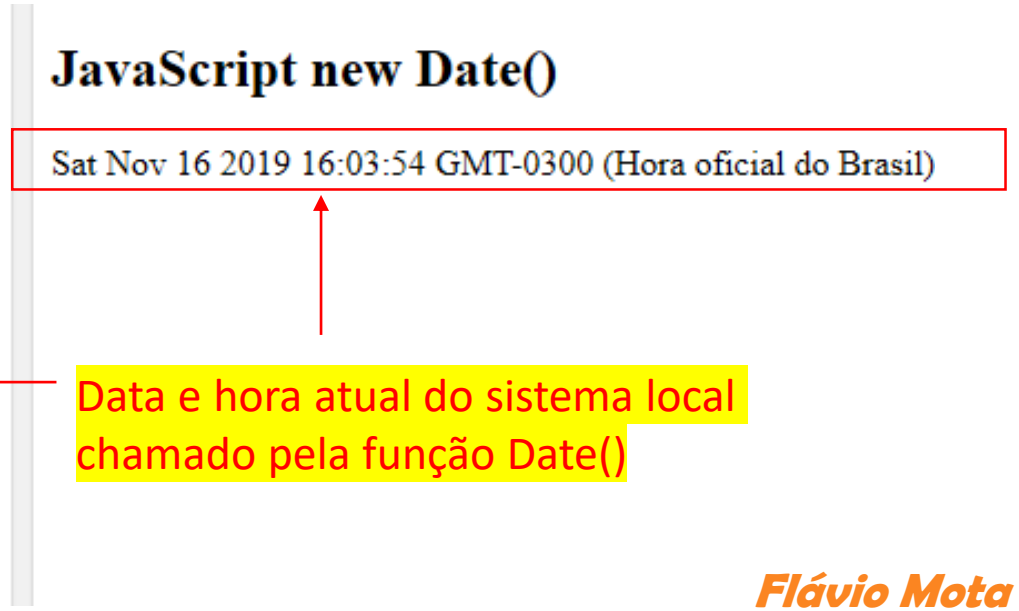
Objeto Date

Em Javascript temos a nossa disposição uma classe com todos os métodos necessários para trabalhar com data e hora, é a classe “Date”, já está tudo pronto, basta chamar os métodos desejado.

Uma observação importante é que a data e a hora retornadas pela classe Date, são a data e hora do computador do cliente e não do servidor, ou seja, o código busca a data e hora configurados em sua máquina

```
var d = new Date();
```

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript new Date()</h2>

<p id="demo"></p>

<script>
var d = new Date();
document.getElementById("demo").innerHTML = d;
</script>

</body>
</html>
```



Fonte: https://www.w3schools.com/js/js_date_methods.asp

Objeto Date

Criando objetos de data. Os objetos de data são criados com o novo construtor Date (). Existem 4 maneiras de criar um novo objeto de data.

```
new Date()
new Date(year, month, day, hours, minutes, seconds, milliseconds)
new Date(milliseconds)
new Date(date string)
```

```
var d = new Date(2018, 11, 24, 10, 33, 30, 0);
```

Nota: JavaScript conta meses de 0 a 11. Janeiro é 0. Dezembro é 11

Fonte: https://www.w3schools.com/js/js_date_methods.asp

Objeto Date

Formatos de data JavaScript

Geralmente, existem três tipos de formatos de entrada de data JavaScript:

Type	Example
ISO Date	"2015-03-25" (The International Standard)
Short Date	"03/25/2015"
Long Date	"Mar 25 2015" or "25 Mar 2015"

```
var d = new Date("2015-03-25");
var d = new Date("2015-03");
var d = new Date("2015");
```

Fonte: https://www.w3schools.com/js/js_date_methods.asp

Objeto Date

Formatos de data JavaScript

Datas curtas do JavaScript. Datas curtas são escritas com uma sintaxe "MM / DD / AAAA", como esta:

```
var d = new Date("03/25/2015");
```

Em alguns navegadores, meses ou dias sem zeros à esquerda podem gerar um erro:

```
var d = new Date("2015-3-25");
```

O comportamento de "AAAA / MM / DD" é indefinido. Alguns navegadores tentarão adivinhar o formato. Alguns retornarão NaN.

```
var d = new Date("2015/03/25");
```

O comportamento de "DD-MM-AAAA" também é indefinido. Alguns navegadores tentarão adivinhar o formato. Alguns retornarão NaN

```
var d = new Date("25-03-2015");
```

Fonte: https://www.w3schools.com/js/js_date_methods.asp

Objeto Date

Formatos de data JavaScript

Datas longas do JavaScript. As datas longas costumam ser escritas com uma sintaxe "MMM DD AAAA", como esta:

```
var d = new Date("Mar 25 2015");

var d = new Date("25 Mar 2015");

var d = new Date("January 25 2015");

var d = new Date("Jan 25 2015");

var d = new Date("JANUARY, 25, 2015");
```

Fonte: https://www.w3schools.com/js/js_date_methods.asp

Objeto Date

Métodos de obtenção de data do JavaScript

Method	Description
getFullYear()	Get the **year** as a four digit number (yyyy)
getMonth()	Get the **month** as a number (0-11)
getDate()	Get the **day** as a number (1-31)
getHours()	Get the **hour** (0-23)
getMinutes()	Get the **minute** (0-59)
getSeconds()	Get the **second** (0-59)
getMilliseconds()	Get the **millisecond** (0-999)
getTime()	Get the time (milliseconds since January 1, 1970)
getDay()	Get the weekday as a number (0-6)
Date.now()	Get the time. ECMAScript 5.

Fonte: https://www.w3schools.com/js/js_date_methods.asp

Objeto Date

O método getTime ()

O método getTime () retorna o número de milissegundos desde 1 de janeiro de 1970:

```
var d = new Date();
document.getElementById("demo").innerHTML = d.getTime(); //Saída 1573944048389
```

O método getFullYear ()

O método getFullYear () retorna o ano de uma data como um número de quatro dígitos:

```
var d = new Date();
document.getElementById("demo").innerHTML = d.getFullYear(); //Saída 2019
```

O método getMonth ()

O método getMonth () retorna o mês de uma data como um número (0-11):

```
var d = new Date();
document.getElementById("demo").innerHTML = d.getMonth(); //Saída 11
```

Fonte: https://www.w3schools.com/js/js_date_methods.asp

Objeto Date

O método getDate ()

O método getDate () retorna o dia de uma data como um número (1-31):

```
var d = new Date();
document.getElementById("demo").innerHTML = d.getDate(); //Saída 16
```

O método getHours ()

O método getHours () retorna as horas de uma data como um número (0-23):

```
var d = new Date();
document.getElementById("demo").innerHTML = d.getHours(); //Saída 19
```

O método getMinutes ()

O método getMinutes () retorna os minutos de uma data como um número (0-59):

```
var d = new Date();
document.getElementById("demo").innerHTML = d.getMinutes(); //Saída 7
```

Fonte: https://www.w3schools.com/js/js_date_methods.asp

RECODE pro

Objeto Date

Os métodos Definir Data permitem definir valores de data (anos, meses, dias, horas, minutos, segundos, milissegundos) para um Objeto de Data.

Method	Description
setDate()	Set the day as a number (1-31)
setFullYear()	Set the year (optionally month and day)
setHours()	Set the hour (0-23)
setMilliseconds()	Set the milliseconds (0-999)
setMinutes()	Set the minutes (0-59)
setMonth()	Set the month (0-11)
setSeconds()	Set the seconds (0-59)
setTime()	Set the time (milliseconds since January 1, 1970)

Fonte: https://www.w3schools.com/js/js_date_methods_set.asp

Objeto Date

O método setFullYear ()

O método setFullYear () define o ano de um objeto de data. Neste exemplo para 2020:

```
var d = new Date();
d.setFullYear(2020);                                    //Saída
document.getElementById("demo").innerHTML = d;  //Mon Nov 16 2020 20:16:44 GMT-0300 (Hora oficial do Brasil)
```

```
var d = new Date();
d.setFullYear(2020, 11, 3);                             //Saída
document.getElementById("demo").innerHTML = d;  // Thu Dec 03 2020 20:23:08 GMT-0300 (Hora oficial do Brasil)
```

Fonte: https://www.w3schools.com/js/js_date_methods_set.asp

Objeto Date

O método setMonth ()

O método setMonth () define o mês de um objeto de data (0-11):

```
var d = new Date();
d.setMonth(11);                                    //Saída
document.getElementById("demo").innerHTML = d;//Mon Dec 16 2019 20:27:38 GMT-0300 (Hora oficial do Brasil)
```

O método setDate ()

O método setDate () define o dia de um objeto de data (1-31):

```
var d = new Date();
d.setDate(15);                                     //Saida
document.getElementById("demo").innerHTML = d; //Fri Nov 15 2019 20:31:59 GMT-0300 (Hora oficial do Brasil)
```

Fonte: https://www.w3schools.com/js/js_date_methods_set.asp

Desafiozinho extra rápido!!! ☺

Crie uma página onde o usuário possa escolher em um elemento select um número de 1 a 12 e a página mostre o nome do mês correspondente em um alert() usando as funções de da Data do Javascript, lembrando que a contagem de mês no Javascript é de 0 a 11.