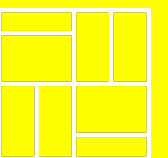


CONCATENAÇÃO

Concatenar é uma termo de programação que significa "juntar" ou "colocar junto".



Concatenação

ARRAYS

Anteriormente aprendemos que para juntar um ou mais arrays a um já existente, usamos a função *concat*

```
var array1 = ['ola'];  
var array2 = ['de novo'];
```

```
array1.concat(array2);  
// [ 'ola', 'de novo' ]
```

ARRAYS e OBJECTS

É possível concatenar um objecto a um array, e não ao contrário, visto que a função *concat* é relativa a arrays.

```
var array1 = ['ola'];  
var object1 = { nome: 'isto é um nome!' };
```

```
array1.concat(object1);  
// [ 'ola', { nome: 'isto é um nome!' } ]
```

STRINGS

Para colocar strings juntas em JavaScript, usamos o operador (+)

```
var variavel = 'ola' + ' ' + 'de novo';  
// 'olá de novo'
```

```
var texto = 'de novo';  
var variavel = 'ola' + ' ' + texto;  
// 'olá de novo'
```

STRINGS e NÚMEROS

Para colocar strings e números juntos em JavaScript, usamos o operador (+).

```
var variavel = 'ola' + 5;  
// 'ola5'
```

```
var variavel = '5' + 5;  
// '55'
```

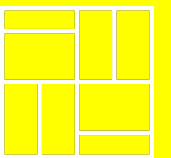
Exercício

CONCATENAÇÃO

- Juntem as strings: 'Isto é um teste' e 'tu vais passar!';
- Formem uma string com strings e números para obter o resultado:
'Isto é 1 string com 2 números'

PROPRIEDADES GLOBAIS

- Propriedades globais retornam um valor simples;
- Não têm propriedades ou métodos.



Propriedades globais

UNDEFINED

O valor global undefined representa um valor indefinido.

```
var variavel = undefined;  
var outraVariavel;
```

NAN

A propriedade global NaN é um valor especial que significa Not-A-Number (não é um número).

```
var variavel = 3 * 'a';  
// NaN
```

NULL

O valor null representa um valor nulo ou "vazio".

```
var variavel = null;
```

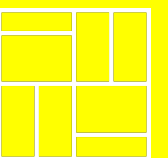
INFINITY

A propriedade global Infinity é um valor numérico que representa o infinito.

```
var variavel = 2**1024  
// Infinity
```

FUNÇÕES GLOBAIS

Funções globais são funções que são chamadas globalmente ao invés de num objeto.



FUNÇÕES GLOBAIS

PARSEINT(argumento)

Analisa um *argumento* string (convertendo-o para uma string primeiro caso necessário) e retorna um inteiro.

```
parseInt("3");  
// 3
```

PARSEFLOAT(argumento)

Analisa um *argumento* (convertendo-o para uma string primeiro caso necessário) e retorna um número decimal.

```
parseFloat("3.3");  
// 3.3
```

ISNAN(valor)

Determina se o *valor* é NaN ou não. Se necessário, o parâmetro é primeiro convertido a um número.

```
isNaN("ahahahaha");    // true  
isNaN("233333");        // false
```

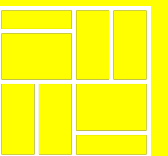
ISFINITE(valor)

Determina se o *valor* transmitido é um número finito. Se necessário, o parâmetro é primeiro convertido a um número.

```
isFinite(2**10);         // true  
isFinite(2**1024);       // false
```

DATE

Cria uma instância JavaScript de Date que representa um único momento no tempo.



Criar um Date

NEW DATE()

Cria a data do momento exacto do sistema.

```
new Date();  
// Sun Oct 16 2022 10:46:52 GMT+0100
```

NEW DATE(valor)

Cria a data a partir do *valor* inteiro representando o número de milissegundos desde 1 de Janeiro de 1970 00:00:00 UTC

```
new Date(1000);  
// Thu Jan 01 1970 01:00:01 GMT+0100
```

NEW DATE(dataString)

Um valor do tipo String que representa uma data.
A string deverá estar uma formato reconhecido pelo método Date.parse()

```
new Date('2014-12-23');  
// Tue Dec 23 2014 00:00:00 GMT+0000
```

```
new Date('2014-25-23');  
// Invalid Date
```

```
new Date('10 06 2023');  
// Fri Oct 06 2023 00:00:00 GMT+0100
```

NEW DATE(ano, mês, dia, hora, minuto, segundo, milissegundo)

Cria a data a partir de todos os valores inteiros.

Nota: Os meses vão de 0 até 11.

```
new Date(1990, 1, 10);  
// Sat Feb 10 1990 00:00:00 GMT+0000
```

Funções de leitura

GETDATE()

Retorna o dia do mês da data especificada.

```
new Date('2014-12-23').getDate();  
// 23
```

GETMONTH()

Retorna o mês da data especificada.

```
new Date('2014-12-23').getDate();  
// 11
```

GETDAY()

Retorna o dia da semana da data especificada.
Os valores da semana começam em 0 para Domingo.

```
new Date('2014-12-23').getDay();    // terça-feira  
// 2
```

GETHOURS()

Retorna as horas da data especificada.

```
new Date('December 25, 1995 23:15:30').getHours();  
// 23
```

GETFULLYEAR()

Retorna o ano da data especificada com 4 dígitos.

```
new Date('2014-12-23').getFullYear();  
// 2014
```

GETMINUTES()

Retorna os minutos da data especificada.

```
new Date('December 25, 1995 23:15:30').getMinutes();  
// 15
```

Funções de escrita

SETDATE(dia)

Altera o *dia* do mês da data especificada.

```
var data = new Date('2014-12-23');  
data.setDate(25);  
data.getDate();  
// 25
```

SETMONTH(mês)

Altera o *mês* da data especificada.

```
var data = new Date('2014-12-23');  
data.setMonth(10);  
data.getMonth();  
// 10
```

SETFULLYEAR(ano)

Altera o *ano* da data especificada com 4 dígitos.

```
var data = new Date('2014-12-23');  
data.setFullYear(1990);  
data.getFullYear();  
// 1990
```

SETHOURS(horas)

Altera as *horas* da data especificada.

```
var data = new Date('December 25, 1995 23:15:30');  
data.setHours(10);  
data.getHours();  
// 10
```

Exercício

DATE

- Criem um Date com a data 6 de Fevereiro de 2020.
- Com a data criada obtenham o dia da semana e o ano;
- Alterem a data para 6 de Março de 2021.