



## API de JS

### FUNÇÃO

---

1. Monte um exemplo básico de função que retorna a string "Hello World", depois faça a chamada desta função.
2. Quais são os possíveis locais de realizar a chamada de uma função.
3. Faça uma página contendo um botão que ao ser clicado ele exibe uma caixa de diálogo com seu nome.
4. Faça uma página contendo um link que ao ser clicado ele exibe uma caixa de diálogo com seu nome.
5. Crie uma função JS que realize a soma de dois números e faça sua chamada passando os argumentos "1" e 1.
6. Crie uma página com um botão que ao ser acionado ele realize a soma de dois números capturados.
7. Crie dois campos de texto e um botão, como ilustrado na Figura 1. Ao clicar no botão o texto do primeiro campo será copiado para o segundo. (**Dica:** `document.getElementById("AlgumaID").value`)

Figura 1 - Página que copia texto.

8. Faça uma página semelhante à Figura 1, porém a ação de copiar o texto do primeiro campo para o segundo deve ser executado com o evento de teclado, como, por exemplo, ao digitar. (**Dica:** veja essa [referência](#))
9. Explique a diferença do evento `keypress` e `keyup` usando a questão anterior.
10. Crie uma calculadora semelhante à Figura 2.

Figura 2 - Calculadora Javascript.

11. Diga qual é a utilidade de uma função anônima, e onde ela é utilizada. E diga se esse recurso possui alguma relação com *closure*? Justifique sua resposta.
12. Como podemos estruturar a *Application Programming Interface* (API) nativa de JS? E liste algumas extensões ou *frameworks* do JS para Web.

### OBJETO

---

13. Explique qual o significado da expressão `"document.write();" e "alert();" e`.

14. Demonstre as maneiras que o JS permite criar objetos e acessar seus elementos. Depois explique se é possível adicionar algum elemento no objeto após sua declaração. (**Dica:** prototype)
15. Qual é a diferença entre a definição de um objeto e sua instância. Mostre um exemplo que para utilizar o objeto é necessário criar a instância através do operador “new” e outro que não seja necessário.
16. Crie um objeto pessoa com as propriedades nome e idade, e o método dadosEmTabelaHTML, utilizando uma das três formas de criação de objeto.
17. Crie um objeto calculadora, através do método construtor, que possui as propriedades operando1 e operando2, e o método soma. Depois, adicione outro método fora da definição do construtor, como por exemplo, um método potência. (**Dica:** prototype)

## ARRAY

---

18. Faça o código necessário para definir diversos arrays com os elementos indicados abaixo. Use a função document.write para exibir o conteúdo do array.
  - a) Conjunto de 6 números inteiros.
  - b) Conjunto com 4 nomes de pessoas.
  - c) Conjunto com os nomes de 4 times de futebol, onde o índice de cada um corresponde à sua posição na tabela de classificação do Campeonato Brasileiro.
  - d) Um array associativo com 3 elementos correspondentes a um usuário: nome, altura e idade.
19. Crie um *array* e exiba seus valores e índices numa tabela em HTML.
20. Altere a questão anterior para que alguns índices sejam textos.
21. Faça um programa que leia um vetor A contendo 10 números inteiros. Depois, calcule e exiba o maior elemento e o índice onde ele se encontra.
22. O *array* pode ser comparado a uma fila, que adiciona e remove elemento no início ou fim de sua estrutura. Demonstre isto através de um exemplo.
23. Faça uma página que exiba uma lista HTML ordenada com os itens do *array* **nomes**. Use primeiro a estrutura de controle **for** e depois use o **for in** para iterar sobre o *array*.

```
var nomes = new Array(
    'Cristovão Colombo',
    'Pedro Álvares Cabral',
    'Dom Pedro de Alcântara',
    'Marechal Deodoro da Fonseca',
    'Joaquim José da Silva Xavier',
    'Edson Arantes do Nascimento'
);
```

24. Copie o *array* da questão anterior e exiba mais duas listas. Uma com os nomes ordenados em ordem alfabética crescente e outra na ordem inversa. (**Obs.:** use as métodos sort e reverse).
25. Defina um *array* associativo onde os índices são siglas e os valores são seus significados. Em seguida, exiba-os usando uma lista de definição HTML (<dl>, <dt> e <dd>).

### Exemplos:

WWW → World Wide Web

PHP → PHP Hypertext Preprocessor

HTTP → HyperText Transfer Protocol

26. Exiba uma lista HTML não ordenada com *links* para os sites cujos nomes e endereços encontram-se no array **sites**. Detalhe a url do site não deve aparecer na lista, apenas o nome com o link. (**Obs.:** Lembre-se de acrescentar o protocolo, `http://`, para o link funcionar adequadamente.)

```
var sites = new Array();
sites[0] = new Array('Meio Bit', 'www.meiobit.com');
sites[1] = new Array('Gizmodo Brasil', 'www.gizmodo.com.br');
sites[2] = new Array('Techtudo', 'www.techtudo.com.br');
```

27. Dado o array **heróis**, exiba os seus dados em uma tabela, conforme o exemplo.

Nome	Herói	Cidade
Bruce Wayne	Batman	Gothan City
Peter Parker	Homem Aranha	New York
Clark Kent	Superman	Metropolis

```
var heroes = [
  {'nome':'Bruce Wayne', 'heroi':'Batman', 'cidade':'Gothan City'},
  {'nome':'Peter Parker', 'heroi':'Homem Aranha', 'cidade':'New York'},
  {'nome':'Clark Kent', 'heroi':'Super Homem', 'cidade':'Metropolis'}
];
```

28. Copie a questão anterior e adicione um *link* em cada nome com uma consulta do nome do herói no google. Para isto, use o *link* `https://www.google.com.br/search?q=<heroi>`, no qual `<heroi>` deve ser o nome do herói.

## STRING

29. Pegue uma *string* e a exiba toda maiúscula, depois exiba apenas a primeira letra em maiúsculo.
30. Crie uma função que receba um número e a quantidade de dígitos que o número deve possuir. Se o número não conferir com a quantidade de dígitos ele deve preencher o início do número com zero. Tipo, se o número informado for 5 e quantidade de dígitos for 2, logo a função deve retornar 05.
31. Use a variável string e efetue as substituições necessárias para deixar todas as ocorrências da palavra “não” em negrito.

```
var string = "Vou não, posso não, quero não, minha mulher não deixa não, não vou não, quero não.";
```

32. Dada uma *string* com um endereço de e-mail qualquer, faça o código necessário para identificar o nome de usuário e o domínio do e-mail. (**Obs:** use a função `split`.)

**Exemplo:** luiz.chaves@ifpb.edu.br

**Usuário:** luiz.chaves

**Domínio:** ifpb.edu.br

33. Pegue o texto "Luiz, 1; Carlos, 2; Chaves, 3", que representa nome dos alunos e seus períodos, e transforme numa tabela em HTML.
34. Altere a questão da calculadora para aceitar apenas número, e caso seja digitado alguma letra deve-se apenas apagar a letra.
35. Pegue este [texto](#) contendo as cidades da Paraíba e monte um caixa de seleção (`<select>`).

36. Dada uma *string* com o nome de uma pessoa, mostre esse nome seguindo o formato conforme o exemplo:

“Ronaldo Nazário de Lima” → “**LIMA**, Ronaldo Nazário de”

37. Adicione um método chamado *reverse* no objeto *String* que retornará seu conteúdo na ordem inversa.
38. Adicione um método no objeto *Array* do JS chamado *upperCaseString* que verifica se o *array* possui alguma *string* e os coloca em maiúsculo.

## DATE

---

39. Faça uma página que mostre uma saudação ao usuário de acordo com a hora atual, seguindo a tabela abaixo.

Hora	Mensagem
6 ... 11	Bom dia!
12 ... 17	Boa tarde!
18 ... 23	Boa noite!
0 ... 6	Vá estudar!

40. Exiba a hora no formato hh:mm:ss.
41. Exiba a data no formato dd/mm/aaaa.
42. Exiba a hora no formato hh:mm:ss que seja atualizado a cada 1 segundo. (**Obs:** utilize o método *setTimeout()*).
43. Dado o array abaixo, verifique se cada uma das datas é uma data válida.

```
var datas = array("31/04/2011", "29/02/2011", "29/02/2012", "31/09/2011");
```

44. Utilize o objeto *Date* e calcule quantos dias faltam para o natal.
45. Exiba a mensagem "boa semana" ou "bom final de semana".
46. Dado a data de nascimento de uma pessoa, mostre a sua idade com precisão de anos, meses e dias.
47. Exiba todos os dias do mês atual em uma lista HTML. Deixe o programa flexível de modo que seja fácil mudar o mês do calendário.
48. Reescreva a questão acima usando agora uma tabela HTML para montar um calendário de um determinado mês. As colunas da tabela devem começar no domingo e terminar no sábado. Ao final, mostre o calendário de todos os meses do ano.
49. Crie uma entrada de texto em HTML que impeça a entrada de datas inválidas.
50. Crie uma entrada de texto em HTML que impeça a entrada de datas anteriores a atual.
51. Veja se o HTML5 possui alguma alternativa para as duas últimas questões sem usar JS.

## NUMBER

---

52. Como é possível exibir um número decimal no formato exponencial.
53. Faça que um número decimal tenha duas casas decimais.

54. Através de um número decimal adicione o carácter de separação de casa decimal e outro de casa de milhar. E ao receber o número 10500.10 ele possa retornar 10.500,10.

## MATH

---

55. Faça uma página que inicialize dois vetores: um com perguntas e outro com respostas. Cada vez que esta página for requisitada, mostre aleatoriamente uma pergunta e uma resposta.
56. Monte uma página que possua entrada de texto para calcular a área de um círculo e retângulo.
57. Tente obter uma equação de segundo grau e calcule o valor de x para y igual a zero.

## DESAFIO

---

58. Faça uma página contendo dois campos de texto que copia o valor do primeiro para o segundo convertendo maiúsculo para minúsculo e vice versa.
59. No mundo real estamos cercados por inúmeros identificadores: CPF, RG, Título de Eleitor, etc. E em algum momento eles são utilizados para informar campos de formulário para as mais diversas finalidades. Para evitar erros no preenchimento desses dados nos formulários a grande maioria deles trabalha com o recurso do dígito verificador, que é um recurso matemático para se gerar algum(ns) dígito(s). Leia esse [link](#) para saber mais informações. Então baseado nisso, veja essa [explicação](#) de como ocorre o cálculo do dígito verificador do CPF e faça uma função validaCPF.