



🏠 Início > Engenharia de Software - Javascript



Espero que você aprenda muito com este exercício.

Aprofunde-se mais nos seus estudos com : [Exercícios Resolvidos](#)

Lista de exercícios de javascript resolvidos

Aprender javascript gratuitamente com esta lista de exercícios resolvidos.



#Índice

Exercícios

- | | |
|-------------------------------|--------------------------------|
| ✓ Exercício 4 | ✓ Exercício 6 |
| ✓ Exercício 1 | ✓ Exercício 7 |
| ✓ Exercício 2 | ✓ Exercício 8 |
| ✓ Exercício 3 | ✓ Exercício 9 |
| ✓ Exercício 5 | ✓ Exercício 10 |

Perguntas e Respostas

Motive-se

#Exercício 1

#2018 #FGV #MPE-AL


```
■ exports.myDateTime = function () {  
    return Date();  
};  
■ aniversarios.js  
■ const dataCorrente = require('./datas.js');  
  console.log(dataCorrente.myDateTime);
```

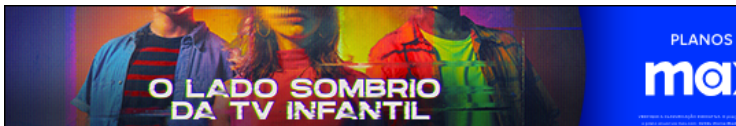
O código do enunciado do exercício retorna um objeto com a data corrente, sendo que a palavra-chave `exports` serve para disponibilizar propriedades e funções fora do arquivo do módulo.

Este exemplo hipotético acima, serviu para demonstrar que o trecho de código Javascript `exports.myDateTime` serve para permitir que a função `myDateTime` seja acessível para outros módulos, o que nos leva a **resposta correta Letra E**.



JavaScript (frequentemente abreviado como JS) é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multiparadigma (protótipos, orientado a objeto, imperativo e, funcional). JavaScript atualmente é a principal linguagem para programação client-side em navegadores web. É também bastante utilizada do lado do servidor através de ambientes como o node.js.

[Wikipedia](#)



#Exercício 2

#2018 #FGV #MPE-AL

Observe a linha de código NodeJS exibida a seguir.

```
var http = require('http');
```

Assinale a opção que melhor descreve esse comando.

- ☐ A Criar um objeto da classe `require`.
- ☐ B Criar um objeto `HTMLrequest`.
- ☐ C Disparar um request [HTML](#).
- ☐ D Incluir o módulo HTTP.
- ☐ E Invocar a função `require`, definida pelo usuário.

RESPONDER

#Resolução 2

Parte 1 da resolução

No sistema de módulos do [Node.js](#), cada arquivo é tratado com um módulo separado, ou seja, o comando `require('http')`, descrito no enunciado deste exercício de javascript, vai carregar o módulo `http`.

De acordo com a explicação acima, a opção que melhor descreve o comando `var http = require('http');` é a **Letra D (incluir o módulo http)**, que é a resposta da questão.



O Node.js pode ser definido como um ambiente de execução Javascript server-side, ou seja, com o Node.js é possível criar aplicações Javascript para rodar como uma aplicação standalone em uma máquina, não dependendo de um browser para a execução. A principal característica que diferencia o Node.JS de outras tecnologias, como PHP, Java, C#, é o fato de sua execução ser single-thread. Ou seja, apenas uma thread é responsável por executar o código Javascript da aplicação, enquanto que nas outras linguagens a execução é multi-thread.

Opus Software

#Exercício 3

#2018 #CESGRANRIO #Banco do Brasil

Considere o código Node.js apresentado a seguir:

```
var http = require('http');
var url = require('url');
http.createServer(function (req, res) {
  res.setHeader("charset", "UTF-8");
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write("<html><head><title>Query String</title></head><body>");
  var querystring = url.parse(req.url, true).query
  for(var i in querystring){
    res.write("<p>" + i + ": " + querystring[i] + "</p>");
  }
  res.end("</body>");
}).listen(8080);
```

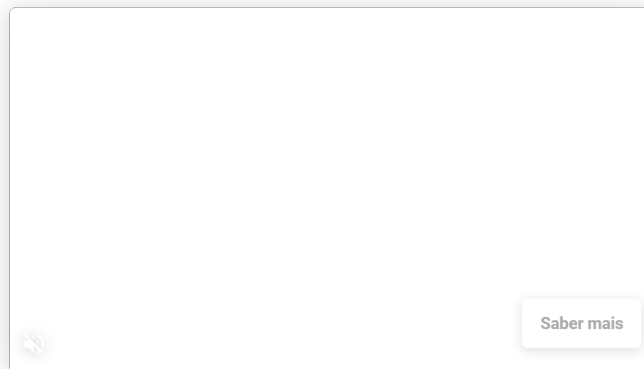
Considere, ainda, que a aplicação foi endereçada com a seguinte url (considere a url correta):

`http://example.com:8080/a.b?c=d&e=f&g`

O resultado obtido será:

- ☐ A site: example.com
port: 8080
c: d
e: f
g:
- ☐ B example.com: 8080
c: d
e: f
g:
- ☐ C example.com: 8080
c: d
e: f
- ☐ D c: d
e: f
- ☐ E c: d
e: f
g:

RESPONDER



Saber mais



Uma engine Javascript(programa ou interpretador de código Javascript) pode ser implementada como um interpretador ou como um compilador(Just-in-time) de bytcodes. A engine V8(construída pelo Google e usada pelo runtime do Node.js) compila código Javascript em código de máquina ao invés de realizar a interpretação, ou seja, não produz bytecode ou qualquer código intermediário, que acaba garantindo uma altíssima performance.

[Exercícios Resolvidos](#)

#Exercício 4

#2018 #FUMARC #COPASA

Analise o seguinte código escrito na linguagem Javascript:

```
var frutas = new Array("banana", "laranja", "limao");
frutas.shift();
frutas.splice(1, 1, "uva");
for(var i=0; i<frutas.length; i++)
    document.write(frutas[i] + " | ");
```

O resultado correspondente apresentado como saída é:

- ☐ A banana | laranja | limao | uva |
- ☐ B banana | uva | laranja | limao |
- ☐ C laranja | uva |
- ☐ D laranja | uva | limao

RESPONDER

#Resolução 4



Este exercício foi solucionado pelo colaborador **João Ricardo de Oliveira Alves**

▪ [Linkedin](#)

Entre em contato conosco: exerciciosresolvidosbr@gmail.com

#Resumo

- As alternativas "A", "B", "D" e "E" estão erradas e a alternativa "C" está correta, pois após cada operação realizada no array de frutas o resultado correspondente apresentado durante a execução do [algoritmo](#) javascript tem como saída: laranja | uva |
- Veja também a [resolução completa e detalhada](#).



As bibliotecas são usadas pelos nossos códigos, enquanto os frameworks é que costumam utilizar os nossos códigos Javascript. Uma biblioteca é uma coleção de códigos voltados a resolver um determinado tipo de problema, enquanto que um framework possui várias funcionalidades prontas que normalmente já possuem um fluxo de trabalho ou estrutura a serem seguidos, ou seja, um framework algo bem mais abstrato do que uma biblioteca.

Treinaweb



#Exercício 5

#2019 #COVEST-COPSET #UFPE

Considere a execução do código JavaScript abaixo, compatível com o ECMAScript 6, e a respectiva numeração das linhas de código na coluna à esquerda, e assinale a alternativa correta.

```
var a = 5;  
var b = 10;  
if(a === 5){  
  let a = 4;  
  var b = 1;  
  console.log(a);  
  console.log(b);  
}  
console.log(a);  
console.log(b);
```

Após a execução do código, desconsiderando os caracteres de quebra de linha da função `console.log()`, o console apresentará:

- ☐ A as saídas: 4, 1, 5 e 1.
- ☐ B as saídas: 4, 1, 4 e 1.
- ☐ C as saídas: 4, 1, 4 e 10.
- ☐ D uma mensagem de erro referente à linha 3.
- ☐ E uma mensagem de erro referente à linha 4.

RESPONDER

#Resolução 5

Parte 1 da resolução

Primeiramente devemos entender que existe uma diferença entre as declarações de variáveis em Javascript, usando `var` e `let`. Veja abaixo a diferença entre cada tipo de declaração:

- `let` vs `var`
 - `let` permite que você declare variáveis limitando seu escopo no bloco, instrução, ou em uma expressão na qual ela é usada. Isso é inverso da keyword `var`, que define uma variável globalmente ou no escopo inteiro de uma função, independentemente do escopo de bloco. Variáveis declaradas com `let` são "içadas" para começo do bloco em que elas são definidas (isso é conhecido também pelo termo, `hoisting`).
 - Declarações de variáveis, onde quer que elas ocorram, são processadas antes que qualquer outro código seja executado. O escopo de uma variável declarada com `var` é seu contexto atual em execução, o qual é a função a qual pertence ou, para variáveis declaradas fora de qualquer função, o escopo é o global. Como as declarações de variáveis (e declarações em geral) são processadas antes de qualquer código seja executado, declarar uma variável em qualquer lugar no código é equivalente a declarar no início. Isso também significa que uma variável pode aparecer para ser usada antes dela ser declarada. Esse comportamento é chamado de "`hoisting`", a variável é movida para o início da função ou do código global.

Parte 2 da resolução

O segundo conceito que devemos aprender para chegarmos a solução do problema é: O Javascript permite redeclarar uma variável com o mesmo nome, sendo que irá prevalecer o último valor atribuído à variável de mesmo nome, conforme demonstrado no exemplo abaixo.

```
var a = 1;  
var a = 2;  
var b = 1;  
console.log(a+b);
```

O comando `console.log(a+b)`; irá imprimir 3 pois a soma realizada foi do último valor atribuído a variável "a" com o valor atribuído a variável "b", ou seja, dois mais um resultando em três.

Parte 3 da resolução

Para chegarmos a **resposta Letra A**, o console apresentará as saídas 4, 1, 5 e 1. Perceba que o programa Javascript descrito no enunciado da questão possui dois escopos, o do próprio programa e o escopo relacionado ao bloco do comando `if`.

- O primeiro comando `console.log(a);` imprimirá o valor 4, pois a variável `let a = 4` dentro do escopo do comando `if` terá precedência sobre o que foi declarado no escopo do programa.
- O segundo comando `console.log(b);` imprimirá o valor 1, pois foi atribuído 1 a variável do contexto do programa, pelo comando `var b = 1;`
- O terceiro comando `console.log(a);` imprimirá o valor 5, pois fora do escopo do bloco `if` a declaração `let a = 4;` não é usada, mas sim a declaração do escopo do programa `var a = 5;`
- O quarto e último comando `console.log(b);` imprimirá o valor 1, pois primeiro prevalecerá a última declaração `var b = 1;` e a variável `b` é do escopo do programa e não do bloco `if`.

Para exemplificar a última impressão do programa do enunciado do exercício, perceba que o programa simplificado abaixo imprimirá 1 pois `var b = 1;` apesar de estar sendo atribuído dentro do escopo do `if` pertencente ao escopo do programa.

```
var a = 5;
if(a === 5){
  var b = 1;
}
console.log(b);
```



Funções que operam sobre outras funções ou as recebendo como parâmetro ou as retornando são chamadas de Higher-Order Functions, tipos de funções que estão em todos os lugares no ecossistema do JavaScript.

Medium Tabeless

#Exercício 6

#2019 #FCC #SANASA Campinas

Considere a página [HTML](#) abaixo, que contém código JavaScript.

```
<body>
  <div id="caixa">
    <p id="p1">Primeiro Parágrafo.</p>
    <p id="p2">Segundo Parágrafo.</p>
  </div>
  <script>
    var para = document.createElement("p");
    var node = document.createTextNode("Terceiro Parágrafo.");
    var element = document.getElementById("caixa");
```

```
____I____;  
</script>  
</body>
```

Sabendo que com HTML DOM, JavaScript pode acessar e mudar os elementos de uma página web, para que o Terceiro Parágrafo seja adicionado ao contêiner identificado como caixa, a lacuna I deve ser corretamente preenchida por

- ☐ A document.getElementById("caixa").add(para)
- ☐ B element.append(para.append(node))
- ☐ C element.innethTML(para.add(node))
- ☐ D element.addNode(para.addContent(node))
- ☐ E element.appendChild(para.appendChild(node))

RESPONDER



Criada pela Microsoft, TypeScript não é uma linguagem nova, mas um superset (ou superconjunto) do JavaScript. O TypeScript fornece recursos para o suporte de uso da [Programação Orientada a Objetos](#), que tem como base quatro princípios fundamentais: encapsulamento, herança, abstração e polimorfismo.

[Medium Tabeless](#)

#Exercício 7

#2019 #FCC #TRF - 4ª REGIÃO

Considere o fragmento de programa JavaScript abaixo.

```
var str = "123456789";  
var p = /^[^5-7]/g;  
var resultado = str.match(p);
```

A variável resultado vai conter

- ☐ A 1,2,3,4,8,9
- ☐ B 5,6,7
- ☐ C 5,7
- ☐ D 1,2,3,4,6,8,9

E 3

RESPONDER



No desenvolvimento de aplicações Javascript, constantemente são usados plugins, pacotes e módulos de terceiros. Gerenciar e organizar esse código é trabalho do Package Manager (Gerenciador de Pacotes) ou Gerenciador de Dependências. NPM é o gerenciador de pacotes padrão do Node.js, que hoje também é aceito como o padrão para o JavaScript, e o Yarn é a mais nova opção em package managers, idealizado pelo Facebook.

[CodeCasts](#)

#Exercício 8

#2019 #UFRR #UFRR

Imagine o seguinte cenário: Marcelo lhe deve um dinheiro e promete pagar em parcelas mensais de R\$ 70; com o intuito de prever qual seria o valor devido após o decorrer de 12 meses, você resolve escrever algumas linhas de código (JavaScript). Levando em consideração que o valor devido por Marcelo era de R\$ 1400, qual seria o total devido após a execução do trecho a seguir?

```
let total_divida = 1400;
function atualizar_divida(valor_recebido){
  total_divida -= valor_recebido;
  return total_divida;
}
for(var i=0;i<12;i++){
  atualizar_divida(70);
}
```

- ☐ A Marcelo lhe deveria 2240 reais pois o loop (for) está incrementando o total da dívida.
- ☐ B Marcelo teria pago 840 reais e lhe deveria ainda 560.
- ☐ C Marcelo lhe deveria ainda 1400 reais, pois o código possui erro semântico.
- ☐ D A declaração da função atualizar_divida está errada.
- ☐ E O trecho apresenta um erro de sintaxe na linha 3.

RESPONDER

#Exercício 9

#2018 #Gestão Concurso #EMATER-MG

JavaScript manipula mensagens de erro de uma maneira que muda de acordo com o navegador usado.

Analise o código JavaScript abaixo.

```
<html>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <script type="text/javascript">
      document.write("Hello World")
    </body>
</html>
```

No navegador Mozilla Firefox, dado o código acima, qual é a mensagem que aparece no console de erro?

- ☐ A Unterminated string constant.
- ☐ B SyntaxError: Unexpected EOF.
- ☐ C SyntaxError: unterminated string literal.
- ☐ D Uncaught SyntaxError: Unexpected token ILLEGAL.

RESPONDER

#Exercício 10

#2018 #IF-RS #IF-RS

Analise a página Web a seguir, escrita em (HTML) e com a linguagem JavaScript.

```
<!DOCTYPE html>
<body>
<script language="Javascript">
function classificacao()
{
  return function(obj1, obj2){
    var a = (obj2["nota"] - obj1["nota"]);
    return (a !== 0) ? a : (obj1["idade"] - obj2["idade"]);
  }
}
var aprovados = [
  {candidato: "Renata Soares", nota: 7.8, idade: 29},
  {candidato: "Marcos Teixeira", nota: 7.8, idade: 26},
  {candidato: "Priscila Gomes", nota: 7.8, idade: 30},
  {candidato: "João Oliveira", nota: 7.8, idade: 27},
  {candidato: "Adriana Telles", nota: 7.8, idade: 28}
];
aprovados.sort(classificacao());
document.write(aprovados[1].candidato);
</script>
</body>
</html>
```

Como pode ser observado, uma função foi criada para ordenar os candidatos do array "aprovados", considerando como critério de ordenação nota e idade. Ao executar este script no navegador Google Chrome, versão 64, pergunta-se: qual candidato ficou em segundo lugar?

- ☐ A Adriana Telles
- ☐ B João Oliveira
- ☐ C Marcos Teixeira
- ☐ D Priscila Gomes
- ☐ E Renata Soares

RESPONDER

Perguntas e Respostas

1. Javascript é uma linguagem de programação ?

+

**Imagine uma nova história para sua vida e acredite nela.***Paulo Coelho*

RESOLUÇÕES

#10/03/2024

[Um Analista de Informática levantou os requisitos para desen . . .](#)

#09/03/2024

[Maria deseja reformar sua casa antes de seu aniversário, mas . . .](#)

#05/03/2024

[O animal La cucaracha, em espanhol, corresponde a qual tradu . . .](#)


NOVIDADES

[Exercícios resolvidos sobre animais em inglês](#)[Exercícios de proposições simples e compostas com gabarito](#)[Exercícios de comparativo em inglês com gabarito](#)[Exercícios de algoritmos para iniciantes com respostas](#)[Lista de exercícios de HTML básico resolvidos](#)

exerciciosresolvidos.com.br

Acreditamos na educação gratuita e de qualidade. Nos ajude com sua opinião, sugestão ou crítica enviando um e-mail para:

exerciciosresolvidosbr@gmail.com

Copyright © 2024. Built with  Love