



Exercícios de fixação

1. Variáveis numéricas: Defina dois valores numéricos (Number) superiores a 2^{10} e eleve um ao outro. Qual foi o valor de saída? Depois faça o mesmo teste utilizando os mesmos valores, porém do tipo BigInt. Os resultados foram diferentes? (Valores Inteiros Big Int)
2. Defina uma string com 3 linhas. Primeiro utilize a definição de string em uma única linha e em seguida com a definição de string com suporte a múltiplas linhas. (Strings)
3. Utilize a técnica de template string para apresentar um texto do resultado da seguinte expressão: $((2 ** 3) * 2) / 4$. Verifique se a expressão foi executada. (Template Literals)
4. Escreva a seguinte string utilizando a codificação Unicode (*escape sequence*) para os caracteres acentuados: „JS suporta codificação Unicode”. Pesquise uma tabela de símbolos unicode na internet (ex. <https://www.rapidtables.com/code/text/unicode-characters.html>). (Strings)
5. Utilize o operador ternário para a comparação estrita entre 1 e „1” e depois faça o mesmo para comparação abstrata. (Operador Ternário)
6. Escreva um programa que transforme a string 17/07/2020 para 17-07-2020. (Wrapper String)
7. Escreva um programa que dado um endereço de email (padrão: usuário@servidor), apresente separadamente o identificador do usuário e o identificador do servidor de email. (Wrapper String)
8. Faça uma divisão que resulte em dízima periódica (ex. 10/3), e apresente o resultado com apenas 2 casas decimais. (Wrapper Number)
9. Considerando a estrutura if/else, experimente a definição de uma variável com escopo de bloco, fazendo seu acesso fora deste bloco para verificação dos resultados. (Desvio condicional, let/var/const)
10. Utilize a estrutura if/else para exibir os seguintes alertas de acordo com os níveis de magnitude em escala Richter: 0- 0.9 Micro, 2.0–3.9: Minor, 4.0–4.9: Light, 5.0–5.9 Moderate, 6.0–6.9 Strong, 7.0–7.9 Major, 8.0–8.9 Great. (Laços do-while)
11. Utilize a estrutura do-while para implementar o cálculo fatorial. Faça uma implementação com Number e outra com BigInt. Veja a partir de qual número o fatorial pode ser calculado apenas com BigInt. (Laços do-while)



12. Utilize uma operação bitwise para calcular o dobro de determinado número (Operadores Bitwise).
13. Implemente uma função que calcule a média de 4 números, e retorne o valor da média. Implemente o mesmo em um procedimento, sem valor de retorno, mas que o resultado seja atualizado em um de seus parâmetros. Escreva externamente a chamada da função o resultado da média (Parâmetros por referência).
14. Implemente uma função que calcule o somatório entre 5 valores. Faça uma implementação por função anônima. Implemente uma vez com a palavra reservada function e outra com a arrow syntax. Atribua a função a uma variável, e experimente a chamada à função por meio da variável. (Função como valor)
15. Refaça a atividade 11, sobre cálculo do fatorial, só que desta vez utilizando a técnica de recursividade. (Recursividade)
16. Defina um objeto com no mínimo 3 propriedades. Por meio do comparador "in" teste uma propriedade existente e uma outra não existente. (Objetos)
17. Utilize o método Object.is é para realizar duas comparações entre objetos, uma deve retornar o valor verdadeiro, e outra o valor falso (Comparando com o método Object.is).
18. Cria um objeto com ao menos 3 atributos. Imprima o valor de um atributo, acessando o mesmo como "chave". Em seguida delete um dos atributos e imprima o objeto para verificar o resultado.
19. Crie dois objetos com duas propriedades distintas cada. Agora crie um terceiro objeto copiando as propriedades dos outros dois por meio do operador spread (Operador spread)
20. Repita o exercício 19 utilizando o operação Object.assign (Object.assign)
21. Escreva uma função que receba como parâmetro um valor, e armazene o fatorial deste valor no atributo fatorial do objeto this. Experimente vincular objeto a está função por meio do operação call e por meio da operação bind. Teste os resultados (contexto).
22. Escreva uma função que receba na entrada um array de valores numéricos, e devolva o maior valor encontrado. A lista deve ser iniciada por meio do método push ou unshift.
23. Escreva uma função que receba um array de objetos e um objeto como parâmetro. Se o objeto existir na lista este deve ser removido, e a função deve retornar o índice que este se encontrava.



24. Dado um array como no mínimo 3 elementos, implemente por meio da desestruturação de array a atribuição de seus valores para variáveis.
25. Dada uma lista de objetos com a menos 3 atributos faça implemente uma função que construa o código SQL para inserção destes objetos. Considere que o nome da tabela poderá ser padronizado e todos seus atributos serão armazenados como texto (forEach, Object.entries)
26. Dado um array de valores numéricos calcule o somatório utilizando a estrutura for .. of ... (for of)
27. Dado um objeto com ao menos 4 propriedades, sendo uma destas outro objeto, gere uma e string com a notação JSON para este objeto. Em seguida, crie uma cópia deste objeto utilizando a string JSON .(JSON. parse JSON.stringify)
28. Utilizando as funções filter, map, e reduce, dado um array de valores numéricos, sendo estes dentre 0 e 10, calcule a média dos quadrados, considerando apenas os valores maiores que 7 (filter, map, reduce)
29. Defina uma classe utilizando prototypes. Instancie dois objetos desta classe, atribua valores diferentes e imprima estes valores na tela. (new)
30. Defina uma classe utilizando prototypes, esta classe deve ter um construtor parametrizável e um método. Depois defina uma subclasse, que deva chamar o construtor da super classe dentro de seu construtor, e também implementar um método adicional. Instancie um objeto da subclasse, e verifique se o construtor da super classe foi invocado, e se os métodos definidos pela superclasse e pela subclasse podem ser acessados a partir deste objeto (Herança com Prototype).
31. Reimplemente o exercício 18, mas ao invés de utilizar os prototypes, utilize a sintaxe de classes e herança introduzido pelo ES6 (Herança com classes).
32. Implemente uma estrutura de herança, onde a super classe contenha atributos privados. Na subclasse, implemente um método que precise fazer acesso a este atributo (métodos de acesso).
33. Implemente o cadastro de objetos de uma entidade, gerando Ids únicos, controlados por um atributo de classe (Atributos e métodos estáticos).
34. Dada uma super classe, implemente duas subclasses, sobre escrevendo um método comum. Inclua os dois objetos em uma lista, e execute uma operação de foreach para cada objeto



chamar seu método. Observe o comportamento distinto para cada objeto de acordo com sua classe (Polimorfismo).

35. Utilize a estrutura Map para armazenar objetos de configuração. Armazene informações como ip, login, e senha para uma conexão. Cada uma destas informações deve ser uma entrada separada no Map. Em seguida faça uma leitura exibindo o nome da chave e o objeto para cada item armazenado (Map).
36. Utilizando uma estrutura de Map inclua objetos que representam municípios brasileiros. Estes objetos devem conter atributos para nome do município e seu código IBGE (registre ao menos 3 objetos). Utilize uma estrutura de WeakMap para armazenar objetos que contenha para cada um dos municípios informações quanto ao tamanho da população e a densidade demográfica, utilize como chave objeto de município já criado. Faça uma exibição para cada município, apresentando seu nome, código IBGE, população e densidade demográfica (WeakMap).
37. Criei uma constante do tipo symbol em programa. Defina um objeto e atribua o valor de um symbol a um de seus atributos. Faça duas comparações, uma entre o valor o atributo e um novo symbol de mesmo valor, e uma segunda comparação com entre o atributo com o próprio Symbol atribuído. Verifique se houve diferença nas comparações (Symbols).
38. Dada uma lista de objetos faça uma varredura pelo método next().(Iterator)
39. Dado uma lista de objetos utilize a estrutura de generators implemente um recurso de paginação, trazendo apenas os próximos X elementos de uma lista quando solicitado. (Slide Generators)
40. Utilizando expressões regulares, verifique se uma string segue um padrão de formatação de data (dd/mm/aaaa) ou não (Expressões regulares).
41. Utilizando expressões regulares, extraia de um CNPJ cada uma das partes do seu número (regulares – grupos nomeados).
42. Dada duas datas, apresente a diferença entre elas em dias (Datas)
43. Implemente uma rotina que percorra uma lista, e ao achar um valor negativo deve lançar um objeto de Erro. Escreva uma função para chamar esta rotina, e esta instrução deve tratar o erro ocorrido, exibindo o índice e o valor relacionados ao erro (Tratamento de erros)
44. Implemente uma função de busca em que cada parte do array é pesquisada por uma thread distinta. A função que encontrar o valor deve imprimir o índice qual este se encontrava por meio da função resolve(). Utilize a função race para iniciar a busca(slide 346 - promises)



45. Utilizando a técnica de mixins como alternativa para herança múltipla, crie uma contato com método getEmail, uma classe usuário com método autenticar, uma classe funcionário com método listaAtividades, herdando os métodos de usuário e contato.
46. Crie um objeto com um conjunto de propriedades. Tente acessar uma propriedade inexistente utilizando operador *optional chaining*, e atribuir o resultado a uma variável. Em seguida imprima o valor da variável caso este seja diferente de undefined ou a mensagem „propriedade inexistente“, fazendo a comparação com o operador nullish coalescing (optional chaining e nullish coalescing).