. 35 - C: Quando declaramos o 'config2', na primeira propriedade passamos '... config1', que faz uma cópia de 'config1'. Em seguida 'opcoes: {... config1.opcoes}' cria uma nova cópia do objeto interno 'opcoes'.
Quando atribuímos o valor 200 à 'config.opcoes.zoom', apenas config2.opcoes é alterado, enquanto 'config1.opcoes.zoom' permanece 100.

- 31 - B: o primeiro elemento da matriz foi ignorado [1,2], enquanto [, y] pega o segundo array, mas ignora o primeiro elemento (3), e y recebe 4.
E o '...resto' pega o restante dos elementos após o segundo [[5,6]]

- 32 - A: Nessa desestruturação, nós definimos explicitamente quais propriedades queremos extrair. Então primeiro entramos na propriedade 'dados'. Dentro de 'preferências', extraímos apenas a propriedade 'time'. E por fim, a pegamos o restante das propriedades dos dados (nome: "Ana").
Obs.: o 'resto' aqui não saiu aninhado dentro de preferências, mas sim no nível de dados.

- 33 - A: Aqui, o parâmetro {a, ...resto} já faz desestruturação. 'a' recebe 1, 'resto' recebe uma nova cópia das outras propriedades {b: 2, c: 3}.
Importante: o operador 'rest' cria um novo objeto, não uma referência ao original.
Então quando atribuímos 99 ao 'resto.c', estamos alterando apenas o objeto resto. Enquanto o objeto original 'dados' não é modificado.

- 34 - A: A função recebe dois parâmetros, porém na chamada é passado apenas um. Logo, o segundo irá utilizar valor padrão.
Durante a chamada, fizemos o • array de forma desestruturada onde x recebe o primeiro valor (1), e '...resto' recebe o restante dos elementos [2,3].
y irá receber resto [2,3] e 1.

onde a propriedade 'cidade' é extraída de dentro de endereço.

- 24 - B: Pois quando há chaves repetidas, o valor do objeto que vem depois, sobrescreve o anterior. como b existe em base(2) e em extra(3), o valor final de b será 3.

- 25 - B: A função recebe um objeto como parâmetro. x e y tem valores padrões. Na chamada, x recebe 5, como y não foi passado, recebe o padrão 20. Então 5+20 = 25

- 26 - C: o operador spread espalha os elementos de arr1 e arr2, os valores são inseridos em um novo array e o número 5 é adicionado ao final.

- 27 - A; A propriedade 'tema' é extraída e renomeada para 'modo'. ... opções recebe o restante das propriedades (font e tamanho).

- 28 - o operador spread passa os elementos do array como argumento da função: soma(10, 20, 30, 40).
dentro da função: a=10, b=20, resto=[30, 40] ~~soma~~ cujo tamanho será 2. Então 10+20+2 = 32.

- 29 - A; hs recebe o primeiro elemento do array ('js').
o segundo elemento (python) é ignorado.
hs recebe o terceiro elemento ('sql')
Nesse caso, extraímos apenas a propriedade 'habilidades'.

- 30 - B: obj2 recebe uma cópia de obj1. através do uso do operador spread. Logo, obj2.a é uma cópia do valor 1, e obj2.b aponta ~~para~~ para o mesmo objeto que obj1.b. Realizamos a atribuição do valor 99 ao objeto compartilhado. Por isso o log irá exibir 99.

- 14 - B: A função reverse vai alterar a ordem dos elementos da lista e retornar.
- 15 - C: A função 'find' vai procurar dentro dos dados, a primeira ocorrência de valor maior que 2.
- 16 - Resultado 37. Primeiro filtramos o array para obtermos apenas os números ímpares (1, 3, 5). Na sequência vamos percorrer o array e modificar cada um dos elementos, multiplicando cada um deles por 3. obteremos então (3, 9, 15). O reduce vai realizar um somatório de todos os elementos, começando com 10 como valor inicial

- 17 - Iremos percorrer o array com For, selecionar apenas os números divisíveis por 3. Após isso calcular a metade de cada um. E por fim, somar todos os metades.
- 18 - O array original é modificado pois o SPLICE altera o próprio array, removendo dos elementos a partir do índice 1.

- 19 -
- 20 - Find retorna o primeiro elemento que satisfaz a condição
  Filter retorna um novo array com todos os elementos que satisfazem a condição do filtro.
  Some retorna um booleano que informa se existe pelo menos um elemento que satisfaz a condição.
- 21 - A: nome recebe "Ana", e "resto" agrupa todos as outras propriedades do objeto usuario, exceto nome.

- 22 - A: a recebe o primeiro elemento (1), o segundo elemento é ignorado (porque vazio). b recebe o terceiro elemento (3), e ...c recebe o restante do array (4, 5)

- 23 - A: Está sendo feita uma desestruturação aninhada

Atividade 4 Fases

Aluno: João Marcos

* Fase 1

- 1 - B: Função push adiciona um elemento ao vetor de 4 posições.
- 2 - C: Função pop vai remover o último elemento da lista.
- 3 - A: Função shift vai remover o primeiro elemento da lista.
- 4 - B: Função unshift vai adicionar um elemento ao início da lista
- 5 - B: Pois a função slice(1) está uma fatia do array até a posição 1 e atribuindo a const b.
- 6 - B: ~~Uset~~ Nesse caso, a função 'splice' vai manipular o array, removendo dois elementos a partir da posição 1.
- 7 - C: A função indexOf vai retornar o número da posição do elemento especificado.
- 8 - B: Nesse exemplo, o Map vai percorrer e modificar cada elemento da lista de Tens, e retornando uma nova lista com os valores pela metade.
- 9 - C: A função filter vai percorrer o vetor e filtrar os elementos baseado na condição $n > 6$ e retorna um vetor apenas com os números que atendam essa condição.
- 10 - D: O reduce irá realizar um somatório de todos os elementos do array e retornar para const soma.
- 11 - A: Pois estamos impulsionando a verificação da existência do nome 'Lucas' dentro do array 'names'.
- 12 - C: A função 'join' está unindo os elementos do vetor, baseado no hífen.
- 13 - C: A função 'concat' está concatenando o vetor (4,5) passado como parâmetro ao array inicial contendo [1,2,3].