



Atribuição Programação Web I

Aluna: Kemelyn Ramos de Figueiredo.

Tarefa 1

Questão 1. A saída será a alternativa d) b), pois na linha 2 é adicionado mais um elemento na lista que já tem 2 elementos, e na linha 3 é retornada a ~~lista~~ a quantidade de elementos.

Questão 2. Alternativa c. O pop() executado na 2ª linha remove o último elemento da lista.

Questão 3. Alternativa a. O shift() executado na 2ª linha remove o primeiro elemento da lista.

Questão 4. Alternativa d; O unshift(0) executado na 2ª linha adiciona o elemento passado por parâmetro na posição 0 da lista.

Questão 5. Alternativa d. Na linha 2 é criada uma variável que recebe a lista da linha 1, no que a função valer(y) é utilizada, então essa é nova lista só que receberá os valores a partir da posição 1 da lista inicial.

Questão 6. Alternativa d. O splice(4,2) na linha 2, remove da lista os elementos da posição 4 e 2.

Questão 7. Alternativa e. O `reduce('e')` retorna a versão do elemento passado por parâmetro.

Questão 8. Alternativa d. Na linha 2, o `map` percorre toda a lista e aplica a função passada por parâmetro em cada elemento da lista, gerando uma nova lista.

Questão 9. Alternativa e. Na linha 2, o `filter` retorna uma nova lista onde os elementos são maiores que 6.

Questão 10. Alternativa cd. Na linha 2, está sendo criada uma nova lista ~~com~~ com a redução dos elementos da lista da linha 1, fazendo a soma dos elementos.

Questão 11. Alternativa ia. O `includer()` retorna um booleano de verificação ~~se~~ feita na lista. Como o nome 'Lucas' está presente, ele retorna true.

Questão 12. Alternativa e. O `join('-')` une todos os elementos da lista da linha 1, separados pelo caractere passado por parâmetro.

Questão 13. Alternativa e. O `concat([4,5])`, une a lista passada por parâmetro com a lista da linha 1.

Questão 14. Alternativa B. (O inverso) na linha 2
tem com que os elementos da lista fiquem na
ordem inversa.

Questão 15. Alternativa C. O find() utilizando na
linha 2, retorna o primeiro elemento da lista
que atende às condições passada por parâ-
metro no find().

Tarefa 2:

Questão 16. A primeira linha contém um array
com 5 elementos. Na linha 2 é criada uma nova
lista com o resultado das chamadas de função,
que inicialmente usa o filter para filtrar os
números ímpares, logo após isso, usa o map
multiplicando cada elemento por 3, e por
fim utiliza o reduce para somar todos os
elementos dessa nova lista, com o detalhe que
o seu acumulador inicia em 10, então o
resultado final será 10 + cada elemento da
nova lista.

Questão 17. Código no Github.

Questão 18. A modificação ocorre pois o splice
remove diretamente do array original, uma das
soluções seria usar o slice, já que ele cria
uma cópia apenas com os elementos desejados,
sem modificar o array original, outra solução
seria espalhar o array original, ~~ou~~ criando
uma cópia da lista original e usar o
splice na cópia, deixando a lista original
intacta.



Questão 19. Código no GitHub.

Questão 20. `find`: retorna o primeiro elemento que satisfaça a condição e o retorna.
`filter`: cria um novo Array com todos os elementos que satisfaçam a condição.
`some`: retorna um valor booleano, caso exista algum elemento que satisfaça a condição, retorna true, caso contrário, false.

Fase 3.

Questão 1. Alternativa a.

Questão 2. Alternativa a.

Questão 3. Alternativa a. Na linha 2 é feita a destruturização do objeto, fazendo com que retorne apenas a cidade do endereço.

Questão 4. Alternativa b.

Questão 5. Alternativa b. A função recebe como parâmetro um objeto desestruturado com valores já padronizados, com a chamada função na linha 21, é passado o valor de `X` e é referenciado como 5, e assim é executada a mesma, uma função.

Questão 6. Alternativa C. Na linha 3 é criada uma nova lista contendo o espalhamento da 1^o e 2^o lista, e o número 5.

Questão 7. Alternativa A. É feita a desestruturação do objeto, onde o conteúdo do item é guardado em modo, e o restante do objeto é espalhado na variável specs.

Questão 8. Alternativa B. É passado como parâmetro da função a lista já sendo desestruturada, os primeiros elementos serão a e b, e o restante da lista é espalhado na lista resto.

Questão 9. Alternativa A. É feita a desestruturação da chave "habilidades", onde h1 é o primeiro elemento, logo após há um espaço vazio para o segundo elemento e h3 é o terceiro elemento.

Questão 10. Alternativa B. Na linha 2 é usada o espalhamento para fazer uma cópia do objeto da linha 1. Na linha 3 é atribuído um valor na chave x e depois esse valor é mostrado no console.

Questão 11. Alternativa B.

Questão 12. Alternativa A.

Questão 13. Alternativa A. O primeiro valor escrito é retornado pela lista da linha 1, já o segundo valor é resultado da atribuição dentro da função, que altera o valor da chave e na cópia, não no Array original.

Questão 14. Alternativa a. É passado como parâmetro da função ~~seja~~ a desestruturação da lista da linha 1, onde x recebe o primeiro elemento, e os outros elementos são espalhados na lista resto, logo após é atribuída a y, na lista resto, através do espalhamento, em seguida x, fazendo com que essa nova lista em y tenha o primeiro elemento como último.

Questão 15. Alternativa C.