

Projeto Classificatório



Mateus Junio da Silva Miranda

01 de junho de 2022 – Sorocaba, SP

Questão 1

Descrevendo a solução para a equipe

Para corrigir os produtos repetidos, foi criada uma função que através dos métodos **reduce()** e **map()**, removiam os mesmos. O método reduce declarava dentro da variável newArray, a função que tinha como parâmetro map e o objeto com o valor inicial, e iterava com o elemento seguinte para validar o iterador ao redor de uma array; já o método map foi utilizado para percorrer toda uma estrutura de array, executar uma determinada ação e retornar o novo conteúdo

```
function removeProdutosRepetidos(i)
{
  var newArray = [...i.reduce((map, obj) => map.set(obj.product_id, obj), new Map()).values()];
  return newArray;
}
```

Para corrigir o estoque foi utilizado o método **filter()**. Este método filtrava o novo array para ignorar as posições que retornariam com o estoque zerado.

```
function removeZeroEstoque(i)
{
  newArray = i.filter((item) => item.product_stock_quantity !== 0);
  return newArray;
}
```

Função chamada na nova variável:

```
cart_products_sanitized = removeZeroEstoque(removeProdutosRepetidos(cart_products));
```

Respondendo o E-mail

Bom dia!

A nova variável 'cart_products_sanitized' foi criada. Dentro dela filtramos e percorremos todos os produtos com estoque zerado ou informações duplicadas. Assim, o estoque foi organizado novamente.

Att,

Questão 2

Descrevendo a solução para a equipe

Para corrigir o preço, foi necessário utilizar iniciar uma variável de posição element no objeto e converte-la para tipo number, através do método **parseFloat()**. Esta função que recebe um argumento de entrada e, se for o caso, converte este argumento do tipo string em ponto flutuante number.

```
//Função que corrige os números que possuem valor string, para number
function corrigirPrecos(i) {

  for(var element in i) {
    var priceErr = i[element].price;
    priceErr = i[element].price.replaceAll('R$ ', '');
    priceErr = parseFloat(priceErr);

    i[element].price = priceErr ;
  }
  return i;
}
```

Função chamada na nova variável:

```
var new_products_list = corrigirPrecos(products_list);
```

Respondendo o E-mail

Olá, como vai?

O problema ressaltado anteriormente, já foi resolvido. Os produtos da variável 'product_list' foram todos transportados para a nova variável 'new_products_list'.

Esta nova variável, recebeu como padrão o conteúdo da anterior, porém com uma função (processos de reformulação de código) atrelada à mesma.

Espero ter ajudado,

Abraços

Questão 3

Descrevendo a solução para a equipe

Para somar o valor dos itens do carrinho, foi criada a função `somarItens`. No início desta função, também foi necessário declarar um escopo semelhante a função do problema anterior, para corrigir os valores que estavam declarados como string, utilizando a função `parseFloat()`. Após isto, foi necessário utilizar a função **`reduce()`**. Esta função declarava dentro da variável `soma`, a função que tinha como parâmetro “a” o valor inicial acumulado, e iterava com b que seria o segundo elemento; feito isso, é ordenado que busque o próximo item do array e somar com a rodada anterior. Além disso, foi dado o valor 0 no final para que a soma fosse realizada a partir da posição 0.

```
//Função para somar o valor final do carrinho
function somarItens(i) {
  for(var element in i) {
    var priceErr = i[element].product_price;
    priceErr = parseFloat(priceErr);

    i[element].product_price = priceErr ;
  }
  var soma = i.reduce((a, b) => a + b.product_price, 0);
  return soma;
}
```

Para a função de validação de frete foi declarada estrutura de condição simples, para indicar a variável booleana `frete` se ela é verdadeira ou falsa.

```
//Função para validar se o frete grátis está incluso ou não
function validaFrete(i)
{
  var frete;
  if (i > 300)
  {
    frete = true;
  }
  else
  {
    frete = false;
  }
  return frete;
}
```

Função chamada na nova variável:

```
var cartPricesInfo = {totalPrice: somarItens(cart_products), freeShipping: validaFrete(somarItens(cart_products))}
```

Respondendo o E-mail

Olá, boa tarde!

A variável que vocês precisavam 'cartPricesInfo' foi declarada. Dentro da mesma há os valores totais do carrinho (totalPrice) e também, se as compras forem acima de R\$300 a posição de frete grátis (freeShipping) é dada como verdadeira ou falsa (true or false)

Espero ter ajudado, qualquer dúvida entre em contato conosco.

Abraços.