

Énoncé : [soldes](#)

Soit un budget de 85€ et Le panier suivant

'Caleçons:6:3' // 'produit,prix,quantité'

'Chaussettes:5:2'

'T-shirt:14:2'

'Pantalons:31:3'

'Chemise:23:2';

Affichage :

Vous devez retirer 3 Pantalons

Vous devez retirer 1 Chemise

<u><i>Idee générale : Mettre Les données dans un tableau !</i></u>
--

L'utilisation d'un tableau permet de manipuler plus facilement l'ensemble des informations.

```
1. const p1 = 'Caleçons:6.99:2',
2. p2 = 'Chaussettes:5.99:5',
3. p3 = 'T-shirt:14.99:3',
4. p4 = 'Pantalons:31.99:2',
5. p5 = 'Chaussures:23.99:1';
6.
7. const listeAchats = [p1,p2,p3,p4,p5];
```

Ainsi pour calculer le total des achats, nous utilisons une fonction qui va boucler sur l'ensemble des produits.

```
8. function total(listeAchats){
9.
10.     let total = 0;
11.
12.     for (const achat of listeAchats) {
13.         total += prixachat;
14.     }
15.
16.     return total;
17. }
18.
```

```
19. console.log(total(ListeAchats));
```

Lig : 13 Le problème reste à déterminer Le prixachats.

Utilisation de Split

Un achat est de La forme 'Caleçons:6:3'

Split permet de créer un tableau avec chaque information séparée.

```
'Caleçons:6:3'.split(":")
```

```
> ["Caleçons", "6", "3"]
```

Voici comment récupérer pour chaque achat les informations de prix et de quantité nécessaire au calcul

```
1. for (const achat of ListeAchats) {  
2.  
3.     let details = achat.split(":");  
4.  
5.     prix = Number(details[1]);  
6.     quantité = Number(details[2]);  
7.  
8. }
```

Examinons [ce code](#)

Il devient facile de calculer le total des achats

```
1. function total(ListeAchats) {  
2.  
3.     let total = 0;  
4.  
5.     for (const achat of ListeAchats) {  
6.         let details = achat.split(":");  
7.         total = total + Number(details[1]) * Number(details[2]);  
8.     }  
9.  
10.    //console.log(total)  
11.  
12.    return total;  
13. }
```

Voici le reste du code pour retirer les achats afin de limiter nos achats à un montant max.

Code

```
/**
```

```
*
```

```
'Caleçons:6:3'
```

```
'Chaussettes:5:2'
```

```
'T-shirt:14:2'
```

```
'Pantalons:31:3'
```

```
'Chaussures:23:2';
```

```
*
```

```
*/
```

```
1. const p1 = 'PantaLon:31:3',
2.   p2 = 'Chaussettes:5:2',
3.   p3 = 'T-shirt:14:2',
4.   p4 = 'Caleçon:6:3',
5.   p5 = 'Chaussure:23:2';
6.
7. const listeAchats = [p1, p2, p3, p4, p5];
8.
9. function total(listeAchats) {
10.
11.   let total = 0;
12.
13.   for (const achat of listeAchats) {
14.     let details = achat.split(":");
15.     total = total + Number(details[1]) * Number(details[2]);
16.   }
17.
18.   //console.log(total)
```

```

19.
20.  return total;
21.}
22./**
23. *
24. * @param {Array} listeAchats Array of items
25. */
26.function positionMax(listeAchats) {
27.
28.  let pos = 0,
29.      max = 0;
30.
31.  for (let index = 0; index < listeAchats.length; index++) {
32.    let details = listeAchats[index].split(":");
33.    if (max < Number(details[1]) && Number(details[2]) > 0) {
34.      pos = index;
35.      max = Number(details[1]);
36.    }
37.
38.  }
39.  return pos;
40.
41.}
42.
43.function retireArticle(listeAchats, pos) {
44.
45.  let details = listeAchats[pos].split(":"),
46.      typeArticle = details[0];
47.  if (details[2] > 0) {
48.    details[2]--;
49.    //console.log(`retire de la liste ${details[0]}`);
50.  } else {
51.    console.log("error");
52.  }
53.  details = details.join(":");
54.  listeAchats[pos] = details;
55.  return typeArticle;
56.}
57.
58.function limitBudget(listeAchats, budget) {

```

```

59.
60.  let achatsARetire = [];
61.
62.  while (total(listeAchats) > budget) {
63.    achatsARetire.push(retireArticle(listeAchats,
        positionMax(listeAchats)));
64.  }
65.  return achatsARetire;
66.}
67.
68.function afficheLimitBudget(listeAretirer){
69.  const o = listeAretirer.reduce((a, x) => {
70.    if (!a[x]) a[x] = 0;
71.    a[x]=a[x]+1;
72.    return a},{});
73.
74.  Object.keys(o).forEach(function(c) {
75.    console.log(`Vous devez retirer ${o[c]} ${c}${o[c]>1?'s':' '} `);
76.  });
77.}
78.
79.
80.console.log(listeAchats);
81.afficheLimitBudget(LimitBudget(listeAchats, 85));
82.console.log(listeAchats);

```

Autre code pour éviter la répétition ! Lig.47

Code

```

/**
 *
 'Caleçons:6:3'
 'Chaussettes:5:2'
 'T-shirt:14:2'
 'Pantalons:31:3'

```

```
'Chaussures:23:2';
```

```
*
```

```
*/
```

```
1. console.time("you")
2. const p1 = 'Pantalon:31:3',
3.    p2 = 'Chaussettes:5:2',
4.    p3 = 'T-shirt:14:2',
5.    p4 = 'Caleçon:6:3',
6.    p5 = 'Chaussure:23:2';
7.
8. const listeAchats = [p1, p2, p3, p4, p5];
9.
10. function total(listeAchats) {
11.
12.     let total = 0;
13.
14.     for (const achat of listeAchats) {
15.         let details = achat.split(":");
16.         total = total + Number(details[1]) * Number(details[2]);
17.     }
18.     return total;
19. }
20.
21. function positionMax(listeAchats) {
22.
23.     let pos = 0,
24.         max = 0;
25.
26.     for (let index = 0; index < listeAchats.length; index++) {
27.         let details = listeAchats[index].split(":");
28.         if (max < Number(details[1]) && Number(details[2]) > 0) {
29.             pos = index;
30.             max = Number(details[1]);
31.         }
32.
33.     }
34.     return pos;
35.
36. }
```

```

37.
38.function retireArticle(listeAchats, pos, ecart) {
39.
40.  let details = listeAchats[pos].split(":"),
41.
42.  typeArticle = details[0],
43.  prix = Number(details[1]),
44.  quantite = Number(details[2]),
45.  nb;
46.
47.  nb = Math.min(Math.ceil((ecart)/(prix)), quantite);
48.  details[2] -=nb;
49.
50.  listeAchats[pos] = details.join(":");
51.  return [typeArticle,nb,quantite*prix];
52.}
53.
54.function limitBudget(listeAchats, budget) {
55.
56.  let achatsARetire = [];
57.  let t = total(listeAchats);
58.
59.  while ( t > budget) {
60.
61.    let [typeArticle,nb,montant] =
      retireArticle(listeAchats,positionMax(listeAchats),t-budget);
62.
63.    achatsARetire.push({typeArticle,nb});
64.
65.    t-=montant;
66.
67.  }
68.  return achatsARetire;
69.}
70.
71.console.log(limitBudget(listeAchats, 85));
72.
73.console.timeEnd("you");

```