

1. Cours

Donner la fonction *filtre* et *transforme* vue en cours.

Ecrire la classe `LinkedList` ayant pour seule méthode `append` (ajout en fin de liste), on prendra soin de définir la classe `LinkedListNode`.

2. Exercice

Voici la liste des villes et leur nombres d'habitants par milliers.	<pre>const villes = [{id: "Evry", h: 20}, {id: "Lyon", h: 89}, {id: "Point", h: 1}, {id: "Paris", h: 12300},];</pre>
---	---

Pour chaque code donner le schéma

<pre>for (let ville of villes){ ville.nbH = ville.h*1000; }</pre>	<pre>const nb = villes.map(({h,id}) => { return {m: `\${h/villes.length}`,id} })</pre>
---	--

3. Exercice

Donner le code avec reduce qui permet d'obtenir le nombres de d'habitants.

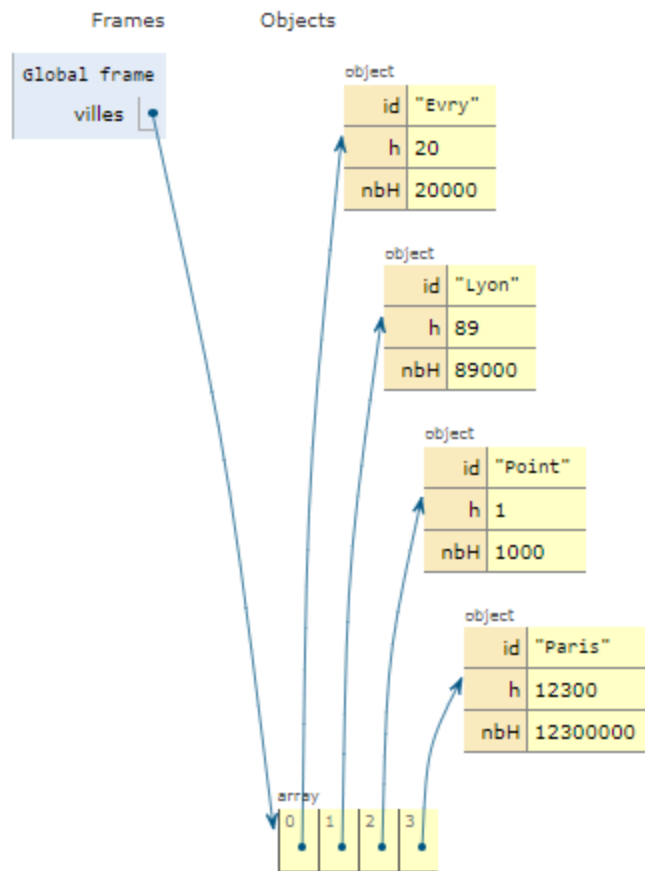
Donner le code avec reduce qui permet d'obtenir la ville qui a le plus grand nombre d'habitants.

Donner le code avec reduce qui permet d'obtenir un objet avec comme propriétés le plus grand nombre et plus petit nombre d'habitants.

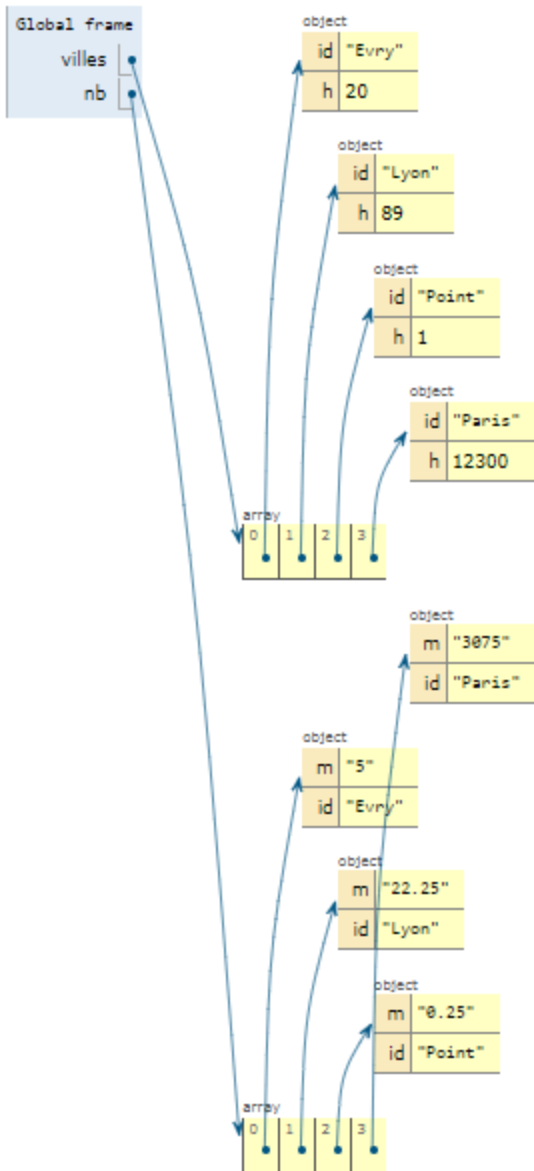
4. Exercice

Comment trouver les deux villes qui ont le plus grand nombre d'habitants (proposer deux solutions dont une avec un `reduce`)

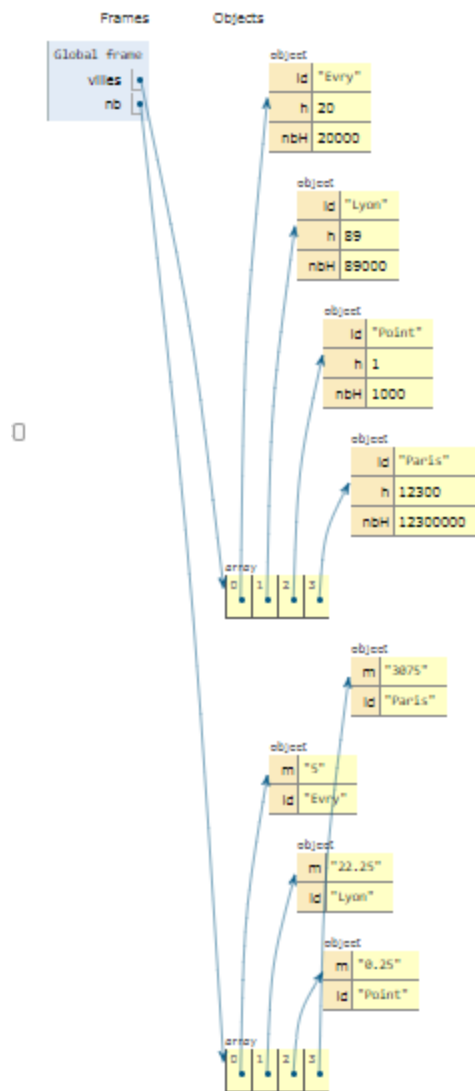
2 : Affichage :



Sii uniquement le code



Bilan ;



<https://goo.gl/8wTU18>

3 :

a) let nbH = villes.reduce((a,c) => a+=c.h,0)

b) let maxH = villes.reduce((a,c) => a.h>c.h?a:c);
console.log(maxH.id);

c) let maxH = villes.reduce((a,c) => {
 if (c.h<a.min) a.min = c.h;
 if (c.h>a.max) a.max = c.h
 return a;
}

```
},{min:Infinity,max:-Infinity} );
```

```
console.log(maxH);
```

<https://jsbin.com/xukagux/1/edit?html,js,console,output>

Exercice 4

```
let trier = villes.reduce((acc, cur) =>
  [cur, ...acc].sort((a,b) => b.h - a.h),[]);
console.log(trier.slice(0,2))
```

```
let [premier,second] = villes.reduce((acc, cur) =>
  [cur, ...acc].sort((a,b) => b.h - a.h).slice(0,2),[]);
console.log(premier,second)
console.log(`Résultats : ${premier.id} et ${second.id}`);
```

<https://jsbin.com/xukagux/2/edit?js,console>

Autre sol

```
const theTwo = function({first ,second }, cur){
  if (cur.h > first.h){
    return {first : cur,second : first}
  }
  if (cur.h > second.h)
    return {first,second : cur}

  return {first ,second}
}
```

```
let {first,second} = villes.reduce(theTwo,{first : villes[0],second : villes[0]});
```

```
console.log(`les deux villes sont ${first.id} et ${second.id}`);
```

<https://jsbin.com/xukagux/3/edit?js,console>