

Exercice

```
class BD{
  constructor(villes){
    this.villes = villes;
  }
  find (query, callback) {
    callback( -- ? -- )
  }
}
// création d'une base
villes = new BD([{ville:"vincennes",h:2},{ville:"paris",h:2},{ville:"roubais",h:1}]);
```

```
// les villes avec h=2 passe à h=30
villes.find({h:2},function(villes){
  for(let ville of villes){
    ville["h"]=30;
  });
});
```

```
// on ajoute la propriété capitale = true
villes.find({ville:"paris"},function(villes){
  for(let ville of villes){
    ville["capitale"]=true;
  });
});
```

Résultat :

```
villes = [{ville:"vincennes","h":30},{ville:"paris","h":30,"capitale":true},{ville:"roubais","h":1}]
```

Compléter le code -- ? -- pour filtrer les arguments du callback.

Aide : *for in* permet de boucler sur les clefs d'un objet.

```
let obj = {a:1, b:2};
for (let prop in obj) {
  console.log(`${prop} = ${obj[prop]}`); // a = 1 b = 2
}
```

Exercice

<p>Donner la fonction <code>itemList</code> qui permet d'afficher une liste de <code>li</code></p> <pre><li data-id="1558957481701"> <div class="view"><label>un</label></div> <li data-id="1558957483611"> <div class="view"><label>deux</label></div> <li data-id="1558957485443"> <div class="view"><label>trois</label></div></pre>	<p>à partir d'un tableau.</p> <pre>[{id: 1558957481701, title: "un"}, {id: 1558957483611, title: "deux"}, {id: 1558957485443, title: "trois"}]</pre>
--	--

Cours

<pre>class Prefix { constructor(pref) { this.pref= pref; } addPref(tab){ -- ? -- } upper(s){ return s[0].toUpperCase() + s.slice(1); } }</pre>	<pre>const T = ["dupont","durand"]; let pMme = new Prefix("M "); console.log(pMme.addPref(T));</pre>
--	---

Compléter le code `-- ? --` pour afficher `["M Dupont", "M Durand"]`

Cours

Donner la fonction *filtrer* et *transformer* vue en cours.

Cours

Donner la fonction qui donne le mot le plus long.

```
const motL = "anticonstitutionnelement myélosaccoradiculographie
cyclopentanoperhydrophénanthrène intergouvernementalisation";
```