

chapitre 5: Les Tableaux

Les tableaux en JS sont Hétérogènes (différents types dans un même tableau) et dynamiques (taille évolutive). Ils peuvent aussi contenir des trous (undefined).

Construction

Littéral

```
1 | let tab = [1,2,3];
```

Constructeur

```
1 | let tab = new Array(1,2,3); //[1,2,3]
2 | let tab2 = new Array(3) = [,,,];
```

Tableau vide

```
1 | let tab = [];
2 | let tab = new Array();
```

Tableau avec une taille x

```
1 | let tab = new Array[x];
2 | let tab = [,,,,,]; // attention, il y aura autant d'éléments que de virgules
   | dans ce tableau (tableau vide)
```

Ne pas confondre

```
1 | New Array("3"); // tableau contenant 3 : ["3"]
2 | new Array(3); // tableau avec 3 trous
```

Accéder à un élément

```
1 | tab[indice];
```

Un accès en lecture hors borne donne **undefined**. Tandis que un accès en écriture hors borne écrit dans le tableau et modifie sa taille.

Il n'est cependant pas possible d'écrire dans un tableau sans l'avoir déclaré au préalable.

Verifier existence élément

```
1 | INDICE in tab //true ou false
```

Un tableau contenant des *undefined* n'est pas un tableau vide

taille d'un tableau

```
1 tab.length
2 tab["length"]
3 tab.length = 23 // change la taille du tableau
```

diminuer la taille du tableau engendre la perte des valeurs stockées et l'augmenter génère des trous à la fin

Il est important de noter qu'un trou est différent de undefined

Tableau associatifs

```
1 tab["clé"] = valeur
```