

# Les tableaux à une dimension

# Les tableaux

- Un tableau (en anglais **array**) est une variable qui permet de stocker un ensemble de valeurs. Chaque valeur possède un **indice**.
- Les indices commencent à **zéro**

**Rq:** puisque JavaScript n'est pas un langage fortement typé, on peut trouver des valeurs de **différents types** dans un tableau.

# Déclaration de tableau(1)

- La première méthode : déclarer un tableau en indiquant les valeurs qu'il contient de la manière suivante:

```
var tab=[ "Lundi", "Mardi", "Mercredi",..., "Dimanche" , 7 , 20];
```

- Les indices d'un tableau commencent toujours à 0
- Pour accéder à un élément d'un tableau, on utilise son **indice** entre []
- Exemple : `alert(tab[2]) ;` //affiche "Mardi"

# Déclaration de tableau (2)

- Il existe une deuxième façon de déclarer un tableau en utilisant le mot clé **Array()**:
- **Exemple :**

```
var tab= new Array ("Lundi","Mardi","Mercredi",...,"Dimanche",7 , 20);
```

# Déclaration d'un tableau(3)

- Syntaxe de la déclaration de base est :

```
var tab=new Array();           //A majuscule
```

```
var tab=new Array(taille ); // si on connaît à la taille
```

- On peut par la suite procéder à l'affectation de valeurs:  
tab[0]="Lundi";  
tab[1]="Mardi";  
tab[2]="Mercredi";  
....  
tab[6]="Dimanche";

# La taille d'un tableau

- Pour connaître le nombre des éléments d'un tableau (la taille) on utilise la propriété **length**.
- *Exemple :*

```
var tab= new Array ("Lundi","Mardi","Mercredi");  
  
var nbr=tab.length; //la valeur retournée est 3
```

# Parcourir un tableau

- On peut parcourir notre tableau avec la boucle **for** comme suit:

```
for(var i=0 ; i< tab.length ; i++){  
  alert(tab[i]);  
}
```

Ce code va nous afficher tous les éléments du tableau.

**Rq:** on peut utiliser aussi les autres boucles.

# Parcourir les éléments d'un tableau

- On peut parcourir les éléments d'un tableau à l'aide de l'opérateur **in** comme dans l'exemple :

```
var T=[25,40,7,6]
for(var x in T){
    alert(T[x])
}
```

- x** ici est un indice du tableau et non pas une valeur du tableau.



# Modifier un élément d'un tableau

- On peut modifier la valeur d'un tableau avec une simple affectation:

```
var tab = new Array ("Lundi","Mardi","Mercredi");  
  
tab[1]="Samedi"  
  
//[ "Lundi", "Samedi", "Mercredi" ]
```

# Saisir les éléments d'un tableau

- Remplir un tableau par des valeurs numériques:

```
var T=[]

for(var i=0;i<3 ; i++){
  var x=parseInt(prompt("Saisir un nombre"))
  T[i]=x
  // T.push(x)
}
```

- Si on veut remplir le tableau par des valeurs chaines, on a pas besoin de faire la conversion de x en entier.

# Ajouter des éléments à un tableau

Il existe des méthodes spéciales qui permettent d'insérer un élément dans un tableau déjà rempli ou vide:

- **push()**: une méthode qui permet d'ajouter un ou plusieurs éléments à la fin d'un tableau.
- **unshift()**: une méthode qui permet d'ajouter un ou plusieurs éléments début d'un tableau.

Exemple:

```
var monTab=["Mohamed","Karim"];  
monTab.push("Siham","Laila");  
  
// ["Mohamed","Karim","Siham","Laila"];
```

```
var monTab=[34,76,12];  
monTab.unshift(5,15);  
  
// [5,15,34,76,12]
```

# Supprimer des éléments d'un tableau

Il existe des méthodes spéciales qui permettent de supprimer un élément d'un tableau:

- **shift()**: méthode qui permet de **supprimer le premier** élément du tableau et renvoie cet élément.

```
var monTab=["Mohamed","Karim", "Siham","Laila"];  
monTab.shift();  
  
// ["Karim","Siham","Laila"];
```

- **pop()**: méthode qui permet de **supprimer le dernier élément** du tableau.

```
var monTab=["Mohamed","Karim", "Siham","Laila"];  
monTab.pop();  
  
// ["Mohamed","Karim","Siham"];
```

# Trier et inverser les éléments d'un tableau

- **reverse()**: c'est une méthode qui permet d'inverser les éléments d'un tableau.

```
var monTab=["Mohamed","Karim", "Siham","Laila"];  
monTab.reverse();  
  
// ["Laila", "Siham","Karim","Mohamed"];
```

- **sort()**: c'est méthode qui trie les éléments d'un tableau. Le tri est effectué sur les éléments du tableau **convertit en chaînes de caractères**.

```
var monTab=["Mohamed","Karim", "Siham","Laila"];  
monTab.sort();  
  
// ["Karim","Laila", "Mohamed", "Siham"];
```

# Récupérer l'indice d'un élément dans la liste

- `monArray.indexOf(valeur)` : méthode qui retourne le premier indice d'une valeur passée en paramètre

- Exemple:

```
var T=[20,50,13,8]
var indice=T.indexOf(50)
alert(indice)
```

- `monArray.lastIndexOf(valeur)` : méthode qui retourne le dernier indice d'une valeur passée en paramètre

```
var T=[20,50,13,8,20]
var indice=T.lastIndexOf(20)

alert(indice)    //affiche 4
```

# Supprimer un élément du tableau à partir de son indice

- `monArray.splice(i, Nb)`: méthode permettant de supprimer l'élément qui a l'indice `i` dans le tableau `monArray`.
- `Nb` est facultatif et signifie le nombre d'élément à supprimer après l'indice `i`.
- Si on veut supprimer l'élément une fois, on doit mettre la valeur `1`

```
var T=[20,50,13,8,20]
var indice=T.indexOf(20)

T.splice(indice,1)
alert(T) //Affiche 50,13,8,20
```

- Si on ne met pas la valeur de `NB`, il va supprimer à partir de l'indice `i` jusqu'à la fin du tableau

# La concaténation de plusieurs tableaux

- **MonArray.Concat(Tab1,Tab2,...)** : permet de concaténer les valeurs des tableaux Tab1, Tab2 ... dans un autre tableau MonArray
- Exemple:

```
var t1=[15,20]  
var t2=[30,8]  
var t3=t1.concat(t2)  
  
alert(t3)
```