

# TD 1

## Prise en main des types de variables

Il s'agit de découvrir le stockage simple de variables, et comment on caractérise le type de la variable (entier, chiffre à virgule, caractère), en distinguant la notion algorithmique et son implémentation dans un langage informatique (ici le javascript).

---

### Variable entière

Notation Algorithmique

$a \leftarrow 5$

Traduction en langage informatique (javascript)

`var a=5 ;`

---

### Variable nombre à virgule

Notation Algorithmique

$a \leftarrow 5,0$

**Remarque :** en français, c'est la virgule qui est le séparateur des chiffres, pas en notation informatique (basée sur les règles d'écriture anglaises)

Traduction en langage informatique (javascript)

`var a = 5.0 ;`

**Remarque :** même si cela semble implicite, il faut spécifier un chiffre après la virgule (le point) pour explicitement indiquer que l'on veut utiliser un nombre à virgule.

---

### Variable de type caractère

Notation Algorithmique

$c \leftarrow \langle \text{NantesUniversité} \rangle$

Traduction en langage informatique (javascript)

`var c = "NantesUniversité"`

**Remarque :** c'est l'utilisation des guillemets (notation anglaise) qui permet de distinguer les nombres des lettres.

---

## Opérations sur les variables de même type

### *Variables de même type (entier)*

#### Notation Algorithmique

```
a ← 5  
b ← 3  
c ← a + b
```

#### Traduction en langage informatique (javascript)

```
var a = 5;  
var b = 3;  
var c = a+b;
```

**Remarque :** opération Identique pour addition, soustraction, multiplication. Pour la division, il y a normalement conversion automatique des chiffres en nombres à virgule.

### *Variables de même type (nombre à virgule)*

#### Notation Algorithmique

```
a ← 5,0  
b ← 3,0  
c ← a / b
```

#### Traduction en langage informatique (javascript)

```
var a = 5;  
var b = 3;  
var c = a+b;
```

---

## Erreur(s) à ne pas commettre

#### Notation Algorithmique

```
a ← 5  
c ← « NantesUniversité »
```

#### Traduction en langage informatique (javascript)

```
var a = 5;  
var c = "NantesUniversité";  
var d = a+c;
```

**Remarque :** Dans le cadre d'un mélange de variable « nombre » + « texte », il y a la plupart du temps un **transtypage**, c'est à dire une conversion de toutes les variables dans un seul type, par défaut en type « caractère » en javascript.

## Listes, tableaux, tableaux associatifs

Les listes permettent de regrouper plusieurs valeurs uniques de manière ordonnée. Pour plus de simplicité, on considère une liste (ou un tableau) comme une suite d'éléments de même type (soit caractère, soit nombre).

**Objectifs** : comprendre comment récupérer un élément de la liste (à partir de son index, mais aussi en recherchant un mot dans par son nom).

### Notation Algorithmique

L = [28,5,52,44,163,1002,1e3,35,-43,0,,46]

### Traduction en langage informatique (javascript)

```
var l = [28,5,52,44,163,1002,1e3,35,-43,0,,46];
```

**Remarque** : il y a des éléments numériques en exposant (notion d'écriture des nombres : 1e3 veut dire 1000) et un élément n'est pas spécifié (case vide entre la valeur 0 et la valeur 46).

A voir aussi : [https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global\\_Objects/Array](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global_Objects/Array)

---

### Accéder au premier élément de la liste

```
var a = c[0]; (attention la numérotation des éléments commence à zéro)
```

---

### Déterminer le nombre d'éléments de la liste

```
var h = l.length;
```

---

### Accéder au dernier élément d'une liste

```
var fin = l[l.length - 1];
```

**Remarque** : comme la numérotation commence à 0, il faut compter le nombre d'éléments dans la liste, puis enlever 1 à ce nombre pour accéder au dernier élément.

---

### Changer un élément de la liste

**Exemple** : changer le 4ème élément dans la liste pour la valeur 256.

```
l[3] = 256;
```

**Remarque** : Attention à la notation de l'indice dans le tableau (3 au lieu de 4). La notation informatique commence à la valeur 0 et non 1. Pour accéder à l'élément n il faut donc utiliser l'index n-1.

---

## Ordonner la liste par valeur croissante

```
var listetree = l.sort() ;
```

Attention par défaut le tri est fait par ordre **alphanumérique** (donc pas numérique comme on pourrait le penser intuitivement).

Le résultat obtenu est donc :

```
[-43,0,1000,1002,163,28,35,44,46,5,52,]
```

Noter que 5 et 52 sont positionnés ensemble (ils commencent pas 5), et que l'élément vide est mis à la fin.

Pour avoir un tri numérique (comme attendu) il faut spécifier la méthode de tri avec une fonction dédiée.

### Algorithme

```
function compare(a, b) {  
    // Exemple issu de  
https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global\_Objects/  
Array/sort  
    if (a est inférieur à b selon les critères de tri)  
        return -1;  
    if (a est supérieur à b selon les critères de tri)  
        return 1;  
    // a doit être égal à b  
    return 0;  
}
```

### Réponse javascript

```
l.sort((a, b) => a - b);
```

cf [https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global\\_Objects/Array/sort](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global_Objects/Array/sort)

---

## Trouver l'élément le plus grand de la liste

```
Math.max()
```

L'idée est de montrer qu'il existe des fonctions déjà existantes liées aux listes ... max, min, ... Il s'agit de montrer rapidement que l'on peut avoir des éléments déjà présents, et d'expliquer aux étudiants comment les trouver.

---

## Trouver l'élément le plus petit de la liste

```
Math.min()
```

# Évaluation de l'impact des approches algorithmiques

**Problème 1 : quel tri de liste est le plus efficace ?**

**Réflexion 2 : comment rechercher un motif spécifique dans un génome complet ?**