

# Tableaux unidimensionnels [tb]

## Résumé de cours

Université de Haute Alsace

Unisciel 

algotprog

 UNIVERSITÉ  
HAUTE-ALSACE

Version 18 mai 2018

## Table des matières

|                                      |          |
|--------------------------------------|----------|
| <b>1 Python - Résumé de cours</b>    | <b>1</b> |
| 1.1 Définitions . . . . .            | 1        |
| 1.2 Notations . . . . .              | 2        |
| 1.3 Tableaux et paramètres . . . . . | 2        |
| 1.4 Parcours de tableaux . . . . .   | 2        |

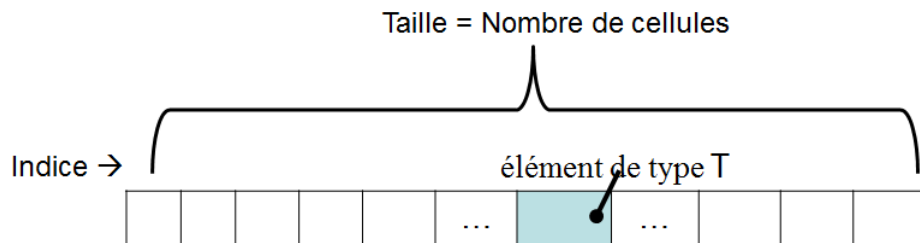
## 1 Python - Résumé de cours

### 1.1 Définitions



#### Tableau

(Sous-entendu unidimensionnel ou linéaire) Collection **homogène indicée** sur  $\mathbb{N}$ , c.-à.d. une séquence d'éléments de **même type** portant tous le même nom et se distinguant les uns des autres par un **indice**.



#### Taille logique vs physique

Dite aussi taille **effective** (le nombre d'éléments effectivement utilisés) que l'on oppose à la taille **physique** (la taille maximale du tableau).

## 1.2 Notations



### Déclaration/Création d'un tableau

```
import numpy as np
nomTab = np.zeros(taille,TypeElement)
```



### Déclaration/Création d'un tableau-liste

```
nomTab = taille * [valInitTypeElement]
```



### Déclaration et initialisation

```
import numpy as np
nomTab = np.array([val1,...,valN],dtype=TypeElement,order='C')
```



### Accès indiciel

```
tab[ k ]
```

## 1.3 Tableaux et paramètres



### Tableau et paramètres

```
TMAX = ...
... ssprg(tab,n,...)
```

## 1.4 Parcours de tableaux

La répétitive **Pour** est le moyen le plus simple de parcourir complètement un tableau. Le parcours partiel est le plus souvent basé sur une répétitive conditionnelle (**TantQue** ou **Répéter**). Les boucles imbriquées combinent généralement une répétitive **Pour** externe avec une répétitive conditionnelle interne.