

DUT MMI – IUT de Marne-la-Vallée
06/11/2013
M1202 - Algorithmique

Cours 3

Tableaux et boucles

Sources

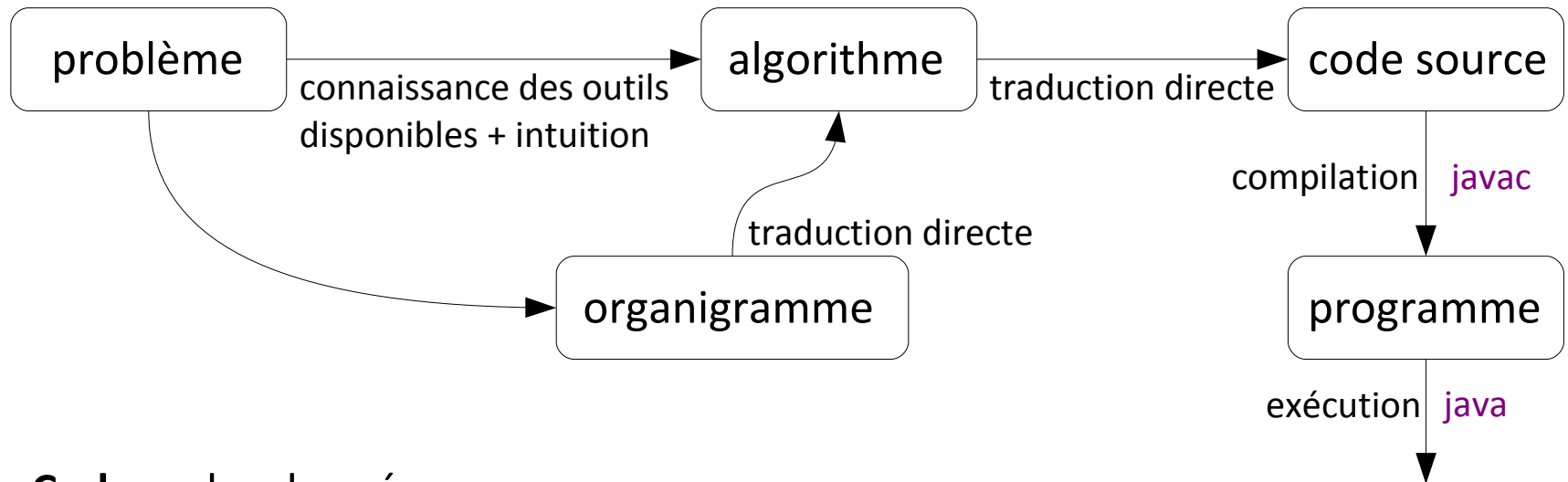
- *Le livre de Java premier langage*, d'A. Tasso
- Cours INF120 de J.-G. Luque
- Cours FLIN102 de l'Université Montpellier 2
- Cours de J. Henriet : <http://julienhenriet.olymp-network.com/Algo.html>
- <http://xkcd.com>, <http://xkcd.free.fr>

Plan du cours 3 – Tableaux et boucles

- Résumé des épisodes précédents
- Les tableaux
- Lecture du contenu d'un tableau
- Dessin d'un graphique à partir du contenu d'un tableau
- La boucle “for” / “pour tout”

Résumé des épisodes précédents

Du problème au programme pour le résoudre :



Codage des données :

- Pour chaque **type** de **variable** (entiers, flottants, chaînes de caractères, couleurs, booléens), une méthode de **codage** en binaire est choisie (en Java : `int`, `float`, `double`, `String`, `Color`, `boolean`, ...)
- Définition d'**opérations de base** pour chaque type de données (en Java : `+`, `-`, `*`, `/`, `%`, `&&`, `||`, `!`, ...)

Plan du cours 3 – Tableaux et boucles

- Résumé des épisodes précédents
- **Les tableaux**
- Lecture du contenu d'un tableau
- Dessin d'un graphique à partir du contenu d'un tableau
- La boucle “for” / “pour tout”

Les tableaux

Les tableaux sont des variables qui contiennent **plusieurs variables de même type**, stockées chacune dans une des cases du tableau.

Par exemple,

Un **tableau d'entiers** :

4
5
1
23
8
9

Un **tableau de chaînes de caractères** :

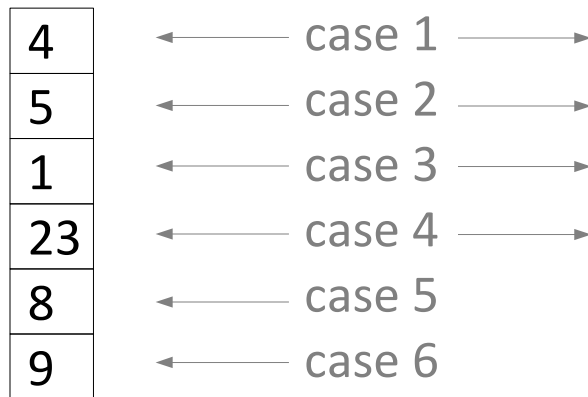
"chaine1"
"chaine2"
"blabla"
"toto"

Les tableaux

Les tableaux sont des variables qui contiennent **plusieurs variables de même type**, stockées chacune dans une des cases du tableau.

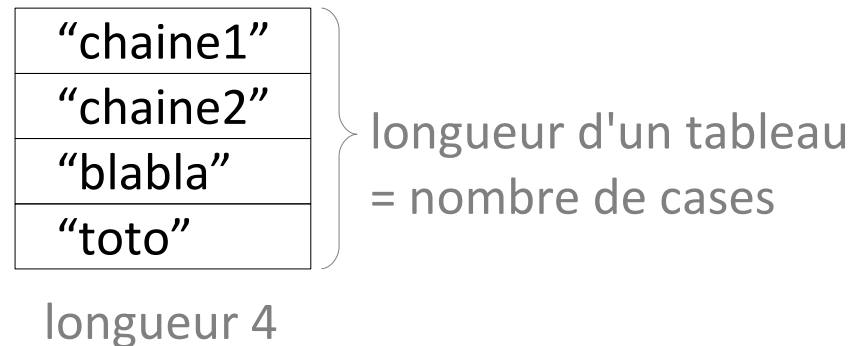
Par exemple,

Un **tableau d'entiers** :



longueur 6

Un **tableau de chaînes de caractères** :



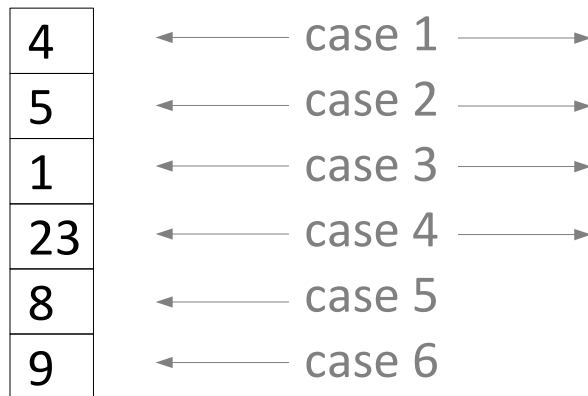
Les tableaux

en pseudo-code

Les tableaux sont des variables qui contiennent **plusieurs variables de même type**, stockées chacune dans une des cases du tableau.

Par exemple, Variables : *tableau1*, un tableau d'entiers,
tableau2, un tableau de chaînes de caractères

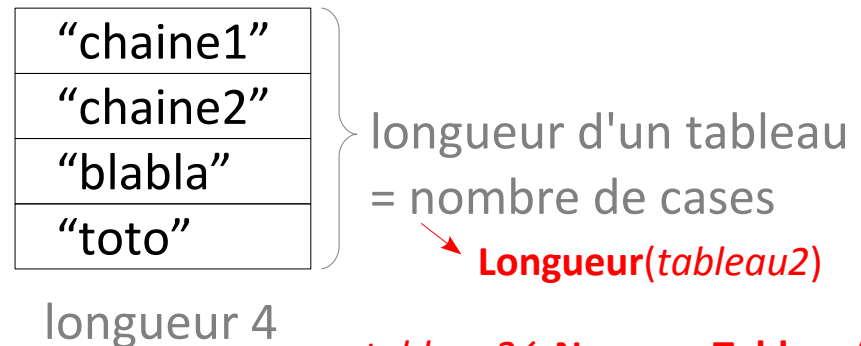
Un tableau d'entiers :



longueur 6

```
tableau1 ← NouveauTableau(6)
Case(tableau1,1) ← 4
Case(tableau1,2) ← 5
Case(tableau1,3) ← 1
Case(tableau1,4) ← 23
Case(tableau1,5) ← 8
Case(tableau1,6) ← 9
```

Un tableau de chaînes de caractères :



```
tableau2 ← NouveauTableau(4)
Case(tableau2,1) ← "chaine1"
Case(tableau2,2) ← "chaine2"
Case(tableau2,3) ← "blabla"
Case(tableau2,4) ← "toto"
```

Plus court :

```
tableau1 ← {4,5,1,23,8,9}
tableau2 ← {"chaine1","chaine2","blabla","toto"}
```

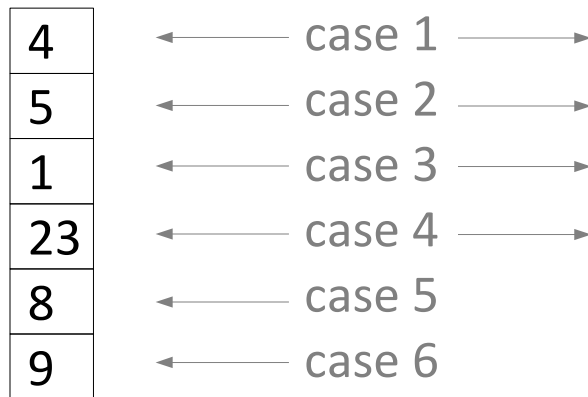

Les tableaux

en Java

Les tableaux sont des variables qui contiennent **plusieurs variables de même type**, stockées chacune dans une des cases du tableau.

Par exemple, `int[] tableau1; String[] tableau2;`

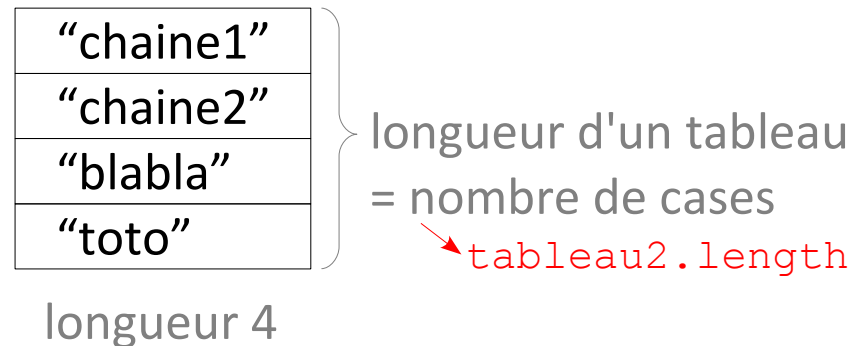
Un tableau d'entiers :



longueur 6

```
tableau1=new int[6];  
tableau1[0]=4;  
tableau1[1]=5;  
tableau1[2]=1;  
tableau1[3]=23;  
tableau1[4]=8;  
tableau1[5]=9;
```

Un tableau de chaînes de caractères :



```
tableau2=new String[4];  
tableau2[0]="chaine1";  
tableau2[1]="chaine2";  
tableau2[2]="blabla";  
tableau2[3]="toto";
```

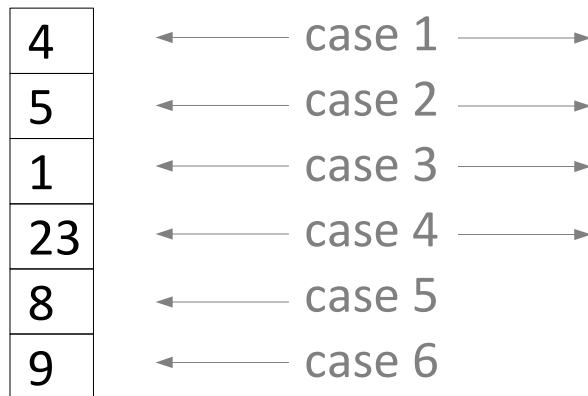
Les tableaux

en Java

Les tableaux sont des variables qui contiennent **plusieurs variables de même type**, stockées chacune dans une des cases du tableau.

Par exemple, `int[] tableau1; String[] tableau2;`

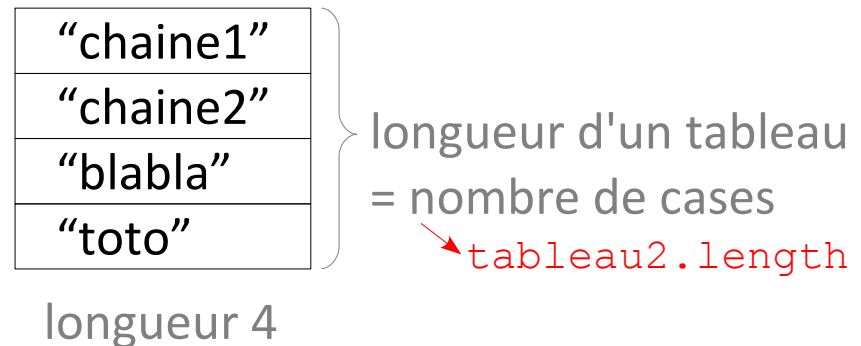
Un tableau d'entiers :



longueur 6

```
tableau1=new int[6];  
tableau1[0]=4;  
tableau1[1]=5;  
tableau1[2]=1;  
tableau1[3]=23;  
tableau1[4]=8;  
tableau1[5]=9;
```

Un tableau de chaînes de caractères :



```
tableau2=new String[4];  
tableau2[0]="chaine1";  
tableau2[1]="chaine2";  
tableau2[2]="blabla";  
tableau2[3]="toto";
```

Attention, cases du tableau `t` numérotées de 0 à `t.length-1` en Java.

Les tableaux

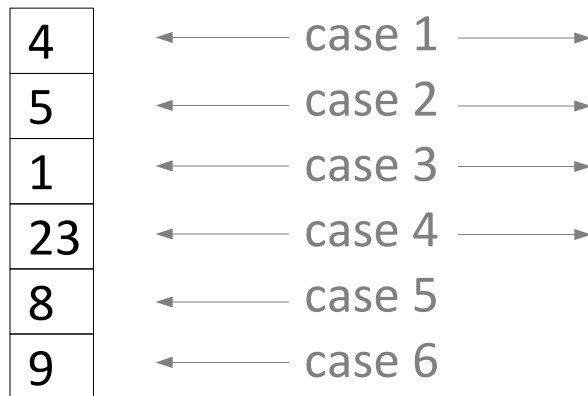
en Java

Les tableaux sont des variables qui contiennent **plusieurs variables de même type**, stockées chacune dans une des cases du tableau.

Par exemple,

```
int[] tableau1; String[] tableau2;
```

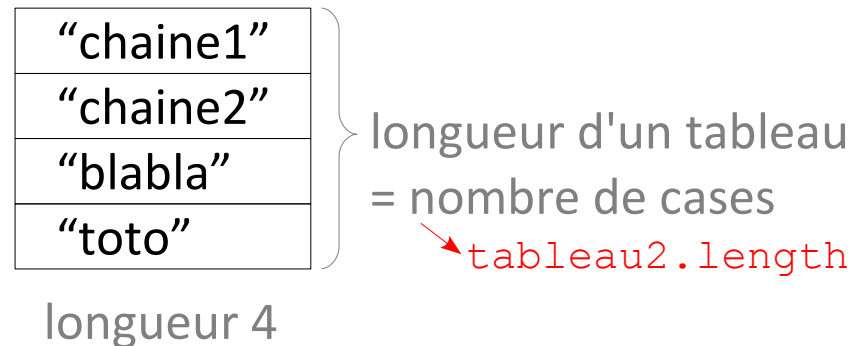
Un tableau d'entiers :



longueur 6

```
tableau1=new int[6];  
tableau1[0]=4;  
tableau1[1]=5;  
tableau1[2]=1;  
tableau1[3]=23;  
tableau1[4]=8;  
tableau1[5]=9;
```

Un tableau de chaînes de caractères :



```
tableau2=new String[4];  
tableau2[0]="chaine1";  
tableau2[1]="chaine2";  
tableau2[2]="blabla";  
tableau2[3]="toto";
```

Plus court (**déclaration + initialisation**) :

```
int[] tableau1 = {4,5,1,23,8,9};  
String[] tableau2 = {"chaine1","chaine2","blabla","toto"};
```

Les tableaux

Pour lire le contenu d'un tableau...
il faut une **boucle pour aller lire chaque case** !

Si le tableau a été prévu trop court au début, **impossible de changer sa longueur**... il faut une boucle pour le recopier dans un tableau plus grand !

Possibilité de créer des **tableaux de tableaux**...

Manipulation et expériences en TD/TP...

Plan du cours 3 – Tableaux et boucles

- Résumé des épisodes précédents
- Les tableaux
- Lecture du contenu d'un tableau
- Dessin d'un graphique à partir du contenu d'un tableau
- La boucle “for” / “pour tout”

Affichage du contenu d'un tableau d'entiers

Algorithme **AfficheTableau**

```
public static AfficheTableau( tableau1) {
```

Affichage du contenu d'un tableau d'entiers

Algorithme **AfficheTableau**

Variable d'entrée : tableau d'entiers *tableau1*

Variable : entier *i*

Début

$i \leftarrow 1$

Tant que $i < \text{Longueur}(\text{tableau1})+1$ faire :

Affiche(Case(*tableau1*,*i*))

$i \leftarrow i+1$

Fin TantQue

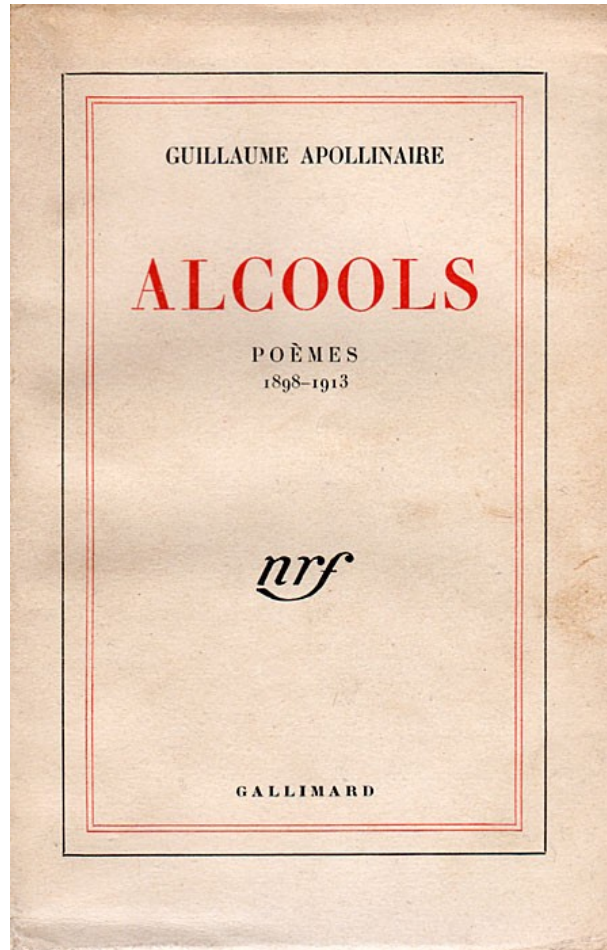
Fin

```
public static void AfficheTableau(int[] tableau1){
    //Afficher les cases du tableau tableau1
    int i;
    i = 0;
    while (i<tableau1.length){
        System.out.println(tableau1[i]);
        i = i+1;
    }
}
```

Plan du cours 3 – Tableaux et boucles

- Résumé des épisodes précédents
- Les tableaux
- Lecture du contenu d'un tableau
- Dessin d'un graphique à partir du contenu d'un tableau
- La boucle “for” / “pour tout”

Graphique du nombre d'apparitions des mots d'un texte



J'ai cueilli ce brin de bruyère
L'automne est morte souviens-t'en
Nous ne nous verrons plus sur terre
Odeur du temps brin de bruyère
Et souviens-toi que je t'attends

Graphique du nombre d'apparitions des mots d'un texte

Un tableau
de chaînes de
caractères *Mots*

j	1
ai	1
cueilli	1
ce	1
brin	2
de	2
bruyère	2
l	1
automne	1
est	1
morte	1
souviens	2
t	2
en	1
nous	2
ne	1
verrons	1
plus	1
sur	1
terre	1
odeur	1
du	1
temps	1
et	1
toi	1
que	1
je	1
attends	1

J'ai cueilli ce brin de bruyère
L'automne est morte souviens-t'en
Nous ne nous verrons plus sur terre
Odeur du temps brin de bruyère
Et souviens-toi que je t'attends

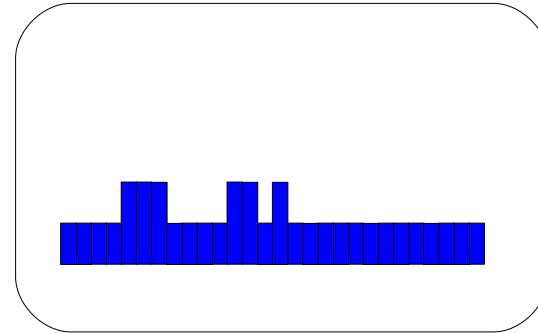
Un tableau d'entiers *NbApparitions*

Graphique du nombre d'apparitions des mots d'un texte

Un tableau
de chaînes de
caractères *Mots*

j	1
ai	1
cueilli	1
ce	1
brin	2
de	2
bruyère	2
l	1
automne	1
est	1
morte	1
souviens	2
t	2
en	1
nous	2
ne	1
verrons	1
plus	1
sur	1
terre	1
odeur	1
du	1
temps	1
et	1
toi	1
que	1
je	1
attends	1

Résultat voulu :

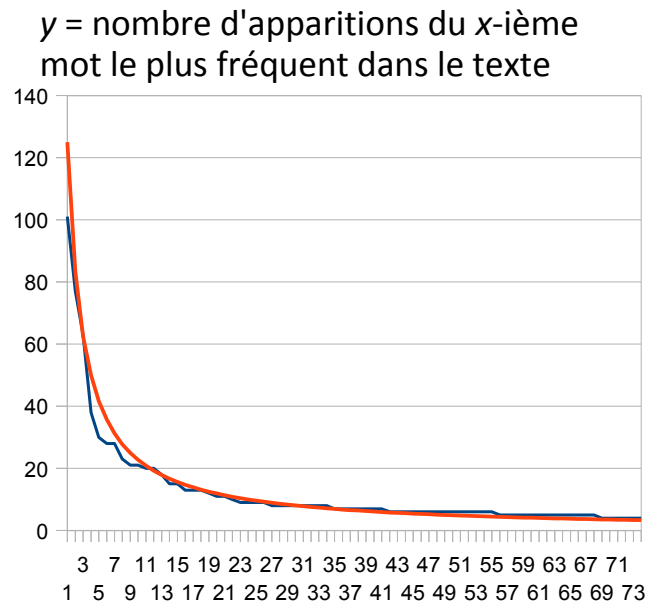


Un tableau d'entiers *NbApparitions*

Le nombre d'apparitions d'un mot dans un texte

La "minute mathématique"

La loi de Zipf prédit la courbe du nombre d'apparitions des mots les plus fréquents d'un texte.

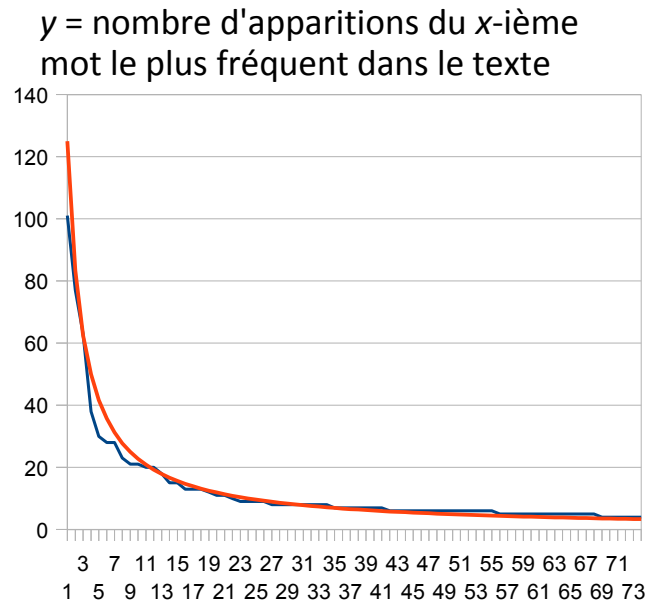


x = numéro du mot (1 pour le plus fréquent, 2 pour le 2^o plus fréquent...)

Le nombre d'apparitions d'un mot dans un texte

La "minute mathématique"

La loi de Zipf prédit la courbe du nombre d'apparitions des mots les plus fréquents d'un texte.



— Nombre réel d'apparition des mots
— Nombre estimé d'apparition des mots

$$y = 250 / (x + 1)$$

x = numéro du mot (1 pour le plus fréquent, 2 pour le 2^o plus fréquent...)

Fonctionne pour n'importe quel texte assez long...

Plan du cours 3 – Tableaux et boucles

- Résumé des épisodes précédents
- Les tableaux
- Lecture du contenu d'un tableau
- Dessin d'un graphique à partir du contenu d'un tableau
- La boucle “for” / “pour tout”

La boucle “for” / “Pour tout...”

Pour parcourir tous les entiers entre deux valeurs entières.

En pseudo-code :

Pour tout entier i de 1 à 42 faire :

...

Fin Pour

En Java :

```
for (int i=1; i<43; i++) {
```

```
    ...
```

```
}
```


La boucle “for” / “Pour tout...”

Exemple : **parcours des cases d'un tableau**

En pseudo-code avec Tant que :

Variables : tableau d'entiers *tab*, entier *i*

$i \leftarrow 1$

Tant que $i < \text{Longueur}(tab)+1$ faire :

[des choses avec la *i*-ième case du tableau **Case**(*tab*,*i*)...]

$i \leftarrow i+1$

Fin Tant que

En pseudo-code avec Pour :

Variables : tableau d'entiers *tab*, entier *i*

Pour *i* de 1 à **Longueur**(*tab*) faire :

[des choses avec la *i*-ième case du tableau **Case**(*tab*,*i*)...]

Fin Pour

La boucle “for” / “Pour tout...”

Pour parcourir tous les entiers entre deux valeurs entières.

En pseudo-code :

Pour tout entier *i* de 1 à 42 faire :

...

Fin Pour

En Java :

```
for(int i=1;i<43;i++) {  
    ...  
}
```

En Java avec while :

```
int i=1;  
while (i<43) {  
    ...  
    i++;  
}
```

La boucle “for” / “Pour tout...”

Pour parcourir tous les entiers entre deux valeurs entières.

En pseudo-code :

Pour tout entier i de 1 à 42 faire :

...

Fin Pour

En Java :

```
for(int i=1, i<43, i++) {  
    ...  
}
```

En Java avec while :

```
int i=1;  
while (i<43) {  
    ...  
    i++;  
}
```

La boucle “for” / “Pour tout...”

La boucle “for” / “Pour tout”

Une boucle pour **parcourir tous les entiers entre deux valeurs entières.**

En Java :

```
int compteur;  
compteur=1;  
while (compteur<mots.length+1) {  
    ...  
}
```

```
for (int compteur=1; compteur<mots.length+1; compteur++) {  
    ...  
    compteur=compteur+1  
    compteur+=1  
}
```

déclaration + initialisation *condition d'arrêt* *mise à jour*

The diagram illustrates the execution flow of a for loop. It shows the three parts of the for loop: 'déclaration + initialisation' (int compteur=1), 'condition d'arrêt' (compteur<mots.length+1), and 'mise à jour' (compteur++). The code snippet shows the loop body with 'compteur=compteur+1' and 'compteur+=1'. Arrows indicate the flow: from the update statement back to the condition, and from the condition back to the update statement, forming a cycle.