

# INF1120 – Programmation 1

## Chaînes de caractères et entrées/sorties

Jacques Berger

# Objectifs

Comprendre les chaînes de caractères

Introduire l'écriture à l'écran

Introduire les lectures au clavier

# Prérequis

Types

# Chaînes de caractères

Une chaîne de caractères est un ensemble de caractères juxtaposés en mémoire

Un caractère spécial indique la fin de la chaîne : '\0'

# Chaînes de caractères

Une chaîne de caractères est un type de données très souvent utilisé

En Java, nous avons un type de données pour représenter les chaînes de caractères : String

# String

L'initialisation d'une string peut se faire avec l'opérateur d'affectation

```
String message = "Ceci est un message!";
```

# Concaténation

La concaténation de chaînes survient lorsqu'on place le contenu de 2 chaînes, l'une à la suite de l'autre, dans une nouvelle chaîne

En Java, la concaténation est faite avec l'opérateur +

```
String partie1 = "Une phrase ";  
String phrase = partie1 + "magique!";
```

# Concaténation

La concaténation peut mélanger des expressions de différents types

Dès qu'un des deux opérandes de l'opérateur + est une string, une concaténation est effectuée au lieu d'une addition



# Concaténation

```
int nombre = 53;  
String phrase = "J'ai récolté " + nombre + " pomme";  
if (nombre > 1) {  
    phrase += "s";  
}  
phrase += " hier.";  
System.out.println(phrase);
```

# Saut de ligne

On peut ajouter `\n` dans une chaîne pour provoquer un saut de ligne

```
System.out.println("Une ligne\nUne autre ligne");
```

# Formattage

Nous pouvons déclarer une string avec des paramètres de formattage aux endroits où l'on veut insérer une variable

On insère ensuite la variable à l'aide de la fonction `String.format`

```
int nombre = 12;  
String resultat = String.format("%d pommes", nombre);
```

# Formattage

%d pour insérer un nombre entier

%f pour insérer un nombre réel

%c pour insérer un caractère

%s pour insérer une string

# Formattage

```
int nombre = 53;
String pluriel = "";
if (nombre > 1) {
    pluriel = "s";
}
String phrase = "J'ai récolté %d pomme%s hier.";
System.out.println(String.format(phrase, nombre,
pluriel));
```

# Formattage

Alignement à droite : %xd où x est le nombre de caractères à utiliser pour afficher la variable

```
int petit = 4;  
int grand = 24566633;  
double pi = 3.1416;  
String format = "%10d\n%10d\n%10f";  
System.out.println(String.format(format, petit, grand,  
pi));
```

# Formattage

Pour spécifier le nombre de décimales d'un nombre réel : `%.xf` où `x` est le nombre de décimales

```
double pi = 3.1416;  
String format = "%.2f";  
System.out.println(String.format(format, pi));  
  
// Combiner l'alignement et les décimales  
String format2 = "%10.2f";  
System.out.println(String.format(format2, pi));
```

# Conversions

Pour transformer un int en String :

```
int nombre = 53;  
String chaine = Integer.toString(nombre);
```



# Conversions

Pour transformer une String en int :

```
String chaine = "435";  
int nombre = Integer.parseInt(chaine);
```

# System.out.println

La fonction `System.out.println` sert à afficher du contenu à la console

# Conversions

Transformer un double en String :

```
double pi = 3.1416;  
String chaine = Double.toString(pi);
```

# Conversions

Transformer une String en double :

```
String chaine = "144.75";  
double nombre = Double.parseDouble(chaine);
```

# Lecture

Un logiciel peut demander à l'utilisateur de lui fournir de l'information

On peut lire ce que l'utilisateur tappe au clavier dans la console

# Clavier

Pour cette lecture, on utilise une classe Clavier spécifiquement conçue pour ce cours

```
Clavier entree = new Clavier();
```

Cette classe est disponible sur Moodle

# Clavier

À partir du clavier, on peut appeler des fonctions pour lire différentes données

lireInt : Lire un entier

lireDouble : Lire un nombre réel

lireString : Lire une ligne (String)

# Plus loin...

Le livre de Java premier langage, 11ème édition  
Anne Tasso  
Eyrolles, 2016  
Chapitres 1 et 2