

Master 1 Humanités numériques – Algorithmique et programmation

TD 5 : Chaînes de caractères et listes

J. Darmont – <http://eric.univ-lyon2.fr/~jdarmont/>

Les exercices sont à formuler en langage algorithmique textuel, en Blockly, puis en Python. Tester l'exécution dans ces deux derniers cas.

Exercice 1 : Manipulation de chaîne de caractères

Soit la chaîne de caractères « Bienvenue dans le M1 Humanités numériques ! ». La stocker dans une variable ch. Puis, stocker dans une variable et afficher (tester à chaque étape) :

1. la taille t de la chaîne ch ;
2. le premier et le dernier caractère c1 et c2 de la chaîne ch ;
3. la sous-chaîne sch1 partant du début de ch jusqu'au 9^e caractère ;
4. la sous-chaîne sch2 allant du 11^e au 20^e caractère de ch ;
5. la sous-chaîne sch3 partant du 22^e caractère jusqu'à la fin de ch ;
6. la chaîne chM qui est la chaîne ch mise en minuscules.

Exercice 2 : Recherches dans une chaîne de caractères

Soit la chaîne de caractères « L'objectif de la mention de master Humanités Numériques est de préparer aux fonctions de niveau cadre qui nécessitent de comprendre de manière technique et critique les données et les documents numériques, les outils qui les produisent, leurs représentations, leurs analyses et leurs usages. Il s'agit d'articuler les problématiques et les méthodes des SHS, de la statistique et de l'informatique, pour élaborer, analyser, interpréter socialement et techniquement des corpus numériques. », que vous pouvez copier/coller depuis le site de l'ICOM/Lyon 2¹ (objectifs, 2^e §). La stocker dans une variable nommée texte.

1. Chercher la position p de la première occurrence du mot « info » dans texte. Afficher p.
2. Remplacer dans texte toutes les occurrences du mot « numérique » par « digital ». Afficher texte après modification (hum !).

¹ <http://icom.univ-lyon2.fr/fr/nos-formations/master/humanites-numeriques/presentation-678085.kjsp>

Exercice 3 : Manipulation de liste de mots

1. Créer une liste nommée `listeMots` contenant les valeurs suivantes : « Humanités », « digitales ».
2. Écrire une procédure nommée `afficher` permettant d'afficher tous les éléments e d'une liste quelconque (nommée `liste`), ainsi que la taille de liste. Appeler la procédure pour `listeMots` après l'initialisation de texte. Tester.
3. Ajouter en 1^{re} position de `listeMots` la chaîne « M1 », puis remplacer le mot « digitales » (qui se trouve toujours en fin de liste) par « numériques » dans `listeMots`. Appeler de nouveau la procédure `afficher` pour `listeMots`. Tester.
4. Trier `listeMots` par ordre alphabétique. Appeler de nouveau la procédure `afficher` pour `listeMots`. Tester.
5. Chercher la position `p` de la première occurrence du mot « M1 » dans `listeMots`. Afficher `p`. Tester.
6. Créer une liste nommée `listeNombres` avec quelques valeurs numériques au choix. Appeler la procédure `afficher` pour `listeNombres`. Est-ce possible en algorithmique ? Tester avec Blockly et Python.

Exercice complémentaire

Copier/coller la procédure `afficher` dans un fichier séparé de nom `listeMod.py` et le sauvegarder.

Correction Exercice 1

Algorithme manipChaîne

Var ch, c1, c2, sch1, sch2, sch3, chM : Chaîne

Var t : Entier

Début

ch ← "Bienvenue dans le M1 Humanités numériques !"

{1}

t ← Longueur(ch)

Écrire("t = ", t)

{2}

c1 ← ch[1]

c2 ← ch[t]

Écrire("c1 = ", c1, " c2 = ", c2)

{3}

sch1 ← SousChaîne(ch, 1, 9)

Écrire("sch1 = ", sch1)

{4}

sch2 ← SousChaîne(ch, 11, 20)

Écrire("sch2 = ", sch2)

{5}

sch3 ← SousChaîne(ch, 22, t)

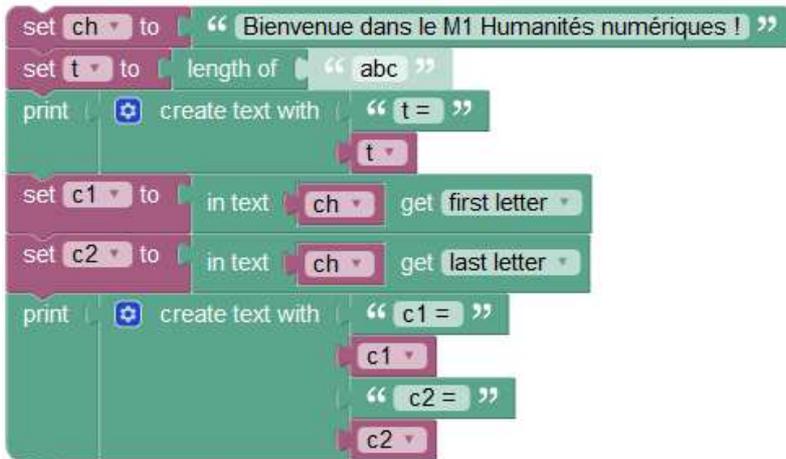
Écrire("sch3 = ", sch3)

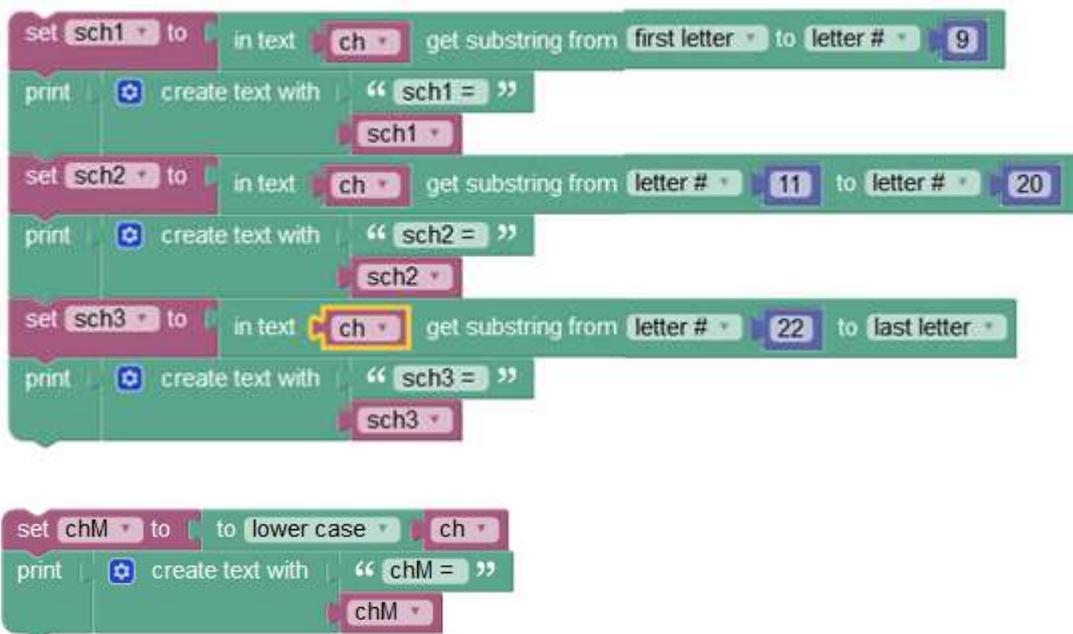
{6}

chM ← Minuscules(ch)

Écrire("chM = ", chM)

Fin





```
ch = "Bienvenue dans le M1 Humanités numériques !"
```

```
#1
```

```
t = len(ch)
```

```
print("t =", t)
```

```
#2
```

```
c1 = ch[0]
```

```
c2 = ch[t - 1]
```

```
print("c1 =", c1, " c2 =", c2)
```

```
#3
```

```
sch1 = ch[: 9]
```

```
print("sch1 =", sch1)
```

```
#4
```

```
sch2 = ch[10 : 20]
```

```
print("sch2 =", sch2)
```

```
#5
```

```
sch3 = ch[21 : ]
```

```
print("sch3 =", sch3)
```

```
#6
```

```
chM = ch.lower()
```

```
print("chM =", chM)
```

Correction Exercice 2

Algorithme rchChaîne

Var texte : Chaîne

Var p : Entier

Début

texte ← "L'objectif de la mention de master Humanités Numériques est de préparer aux fonctions de niveau cadre qui nécessitent de comprendre de manière technique et critique les données et les documents numériques, les outils qui les produisent, leurs représentations, leurs analyses et leurs usages. Il s'agit d'articuler les problématiques et les méthodes des SHS, de la statistique et de l'informatique, pour élaborer, analyser, interpréter socialement et techniquement des corpus numériques."

{1}

p ← ChercherPosition(texte, "info")

Écrire("p = ", p)

{2}

texte ← Remplacer(texte, "numérique", "digital")

Écrire(texte)

Fin



texte = "L'objectif de la mention de master Humanités Numériques est de préparer aux fonctions de niveau cadre qui nécessitent de comprendre de manière technique et critique les données et les documents numériques, les outils qui les produisent, leurs représentations, leurs analyses et leurs usages. Il s'agit d'articuler les problématiques et les méthodes des SHS, de la statistique et de l'informatique, pour élaborer, analyser, interpréter socialement et techniquement des corpus numériques."

#1

p = texte.find("info")

print("p =", p)

#2

texte = texte.replace("numérique", "digital")

print(texte)

Correction Exercice 3

Procédure afficher (liste : Liste de chaînes)

Var e : Chaîne

Début

Pour e dans liste faire

Écrire(e)

Fin pour

Écrire(Longueur(liste))

Fin

Algorithme manipListe

Var listeMots : Liste de chaînes

Début

```
listeMots ← ("Humanités", "digitales")  
afficher(listeMots)  
Ajouter(listeMots, 1, "M1") {Ajout en première position}  
listeMots(Longueur(ListeMots)) ← "numériques" {Modification en dernière position}  
afficher(listeMots)  
Trier(listeMots)  
afficher(listeMots)  
p ← ChercherPosition(listeMots, "M1")  
Écrire("p = ", p)
```

Fin

The image shows a sequence of Scratch code blocks implementing the algorithm. The blocks are as follows:

- to afficher with: liste**
 - for each item e in list liste
 - do print e
 - print length of liste
- set listeMots to create list with " Humanités " and " digitales "
- afficher with: liste listeMots
- in list listeMots insert at first as " M1 "
- in list listeMots set last as " numériques "
- afficher with: liste listeMots
- set listeMots to sort alphabetic ascending listeMots
- afficher with: liste listeMots
- set p to in list listeMots find first occurrence of item " M1 "
- print create text with " p = " and p
- set listeNombres to create list with 1, 3.14, and -20
- afficher with: liste listeNombres

```
def afficher(liste):
    for e in liste:
        print(e)
    print(len(liste))

listeMots = ["Humanités", "digitales"]
afficher(listeMots)

listeMots.insert(0, "M1")           # Ajout en première position
listeMots[len(listeMots) - 1] = "numériques" # Modification en dernière position
afficher(listeMots)
listeMots.sort()                   # Tri
afficher(listeMots)
p = listeMots.index("M1")          # Recherche
print("p =", p)

listeNombres = [1, 3.14, -20]
afficher(listeNombres)
```