

# chaine de caractères Langage C

## string.h

- strlen(s)
- strcpy(s, t)
- strcat(s, t) : ajouter t à la fin des
- strcmp(s, t) : compare s et t ( $t = s \geq 0, s - t > 0, s - t < 0$ )
- strncpy(s, t, n)
- strncat(s, t, n)
- strchr(s, c)

## <ctype.h

- isupper(<c>)
- islower(<c>)
- isdigit(<c>)  $\in ('0', \dots, '9')$
- isalpha(<c>)  $\in ('a', 'b', \dots, 'z', 'A', 'B', \dots, 'Z')$

isalnum(<c>) Si isalpha ou isdigit

isxdigit(<c>) Si <c> est  $'0' \dots '9'$  ou  $'A' \dots 'Z'$  ou  $'a' \dots 'z'$   
hexadecimal

ter-lower(<c>)

ter-upper(<c>)

# python

## \* Liste

- L = [] (declaration)
- liste.append(x) / ajoute de x à la liste
- Liste [debut : fin : pas] : pour afficher une tranche
- len(L)
- min(L)
- max(L)
- sum(L)
- L.count(elem) : compte le nombre de (elem) dans la liste
- elem in L, elem not in L : boolean (vrai ou faux)
- L.insert(indice, elem)
- L.extend(seq) ajout le contenu d'une sequence seq à L
- L.sort() trie la liste par ordre croissant
- L.remove(elem) : supprime le premier élément trouvé dans la liste L
- L.pop() : supprime et retourne le dernier élément de L
- L.pop(i) supprime et retourne l'élément à la position i de L
- L.reverse() : inverse l'ordre des éléments de la liste
- reversed(L) : affiche la liste inversé de L sans l'effectuer

## \* Tuples

- len(Tup)
- min(Tup)
- max(Tup)
- Tup.count(elem)
- tup.index(elem) : retourne la première occurrence de l'élément n'il existe  
                        risque d'erreur
- sep - join(tup) : convertit la tuple en une chaîne de caractères  
                        en ajoutant un séparateur entre chaque élém

## \* Chaine

- len(ch)
- ch.lower(): convertit en minuscule
- ch.upper(): --- majuscule
- ch.title(): convertit l'initial de chaque mot en majuscule
- ch.swapcase(): convertit minuscule → majus et tout majus → minis
- ch.capitalize(): convertit en majus <sup>Tout</sup> seulement la 1<sup>re</sup> lettre de la phrase
- ch.find(~~s~~): cherche la position de x dans ch  
<sup>sous chaine</sup>
- ch.count(sch): compte le nombre d'occurrences sch dans la chaine de caractères ch
- ch in / not in ch : vérifie
- ch.startswith(~~s~~): vérifie
- ch.split(): voir cours page 38