

Chaine de caractere

Langage C

string.h

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

ctype.h

- isupper(<c>)
- islower(<c>)
- isdigit(<c>) ∈ ('0', ..., '9')
- isalpha(<c>) ∈ ('a', 'b', ..., 'z', 'A', 'B', ..., 'Z')

isalnum(<c>) Si isalpha ou isdigit

isxdigit(<c>) Si <c> est ('0'... '9' ou 'A'... 'Z' ou 'a'... 'f')
hexadecimal

tolower(<c>)

toupper(<c>)

python

* Liste

- `L = []` (declaration)
- `list.append(x)` / ajout 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 inversée de `L` sans l'affecter

* Tuple

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

*Chaine

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