```
Instruction boucle conditionnelle
                                                                                                                Instruction boucle itérative
                                                                              bloc d'instructions exécuté pour
   bloc d'instructions exécuté
fт
                                                                               chaque élément d'un conteneur ou d'un itérateur
   tant que la condition est vraie
boucles sans
                                                                                           for var in séquence:
      while condition logique:
                                                               Contrôle de boucle
                                                                                                                                         fini
                                                                                                 bloc d'instructions
                                              non
            bloc d'instructions
                                                       break
                                                                      sortie immédiate
                                                        continue
                                                                    itération suivante
                                                                                                                                                boucle
                                                                                        Parcours des valeurs d'un conteneur
    = 0 initialisations avant la boucle
                                                          ₫ bloc else en sortie
                                                                                        s = "Du texte" | initialisations avant la boucle
                                                           normale de boucle.
           condition avec au moins une valeur
                                                                                       cpt = 0
                                                                                                                                                g
   while i <= 100:
                                                                                    variable de boucle, affectation gérée par l'instruction for for c in s:
                              variable (ici i)
                                                             Algo: i=100
                                                                                                                                                variable
        s = s + i**2

i = i + 1
                         if c == "e":
                                                                                                                              Algo: comptage
  print("somme:",s)
                                                                                                  cpt = cpt + 1
                                                                                                                              du nombre de e
                                                                                       print("trouvé", cpt, "'e'")
                                                                                                                                                lа
                                                                                                                              dans la chaîne.
                                                                  Affichage
                                                                               boucle sur dict/set ⇔ boucle sur séquence des clés ; utilisation des
                                                                                                                                                modifier
 print("v=",3,"cm :",x,",",y+4)
                                                                               tranches pour parcourir un sous-ensemble d'une séquence
                                                                               Parcours des index d'un conteneur séquence
 éléments à afficher : valeurs littérales, variables, expressions
                                                                                                                                                pas 1
                                                                               □ changement de l'élément à la position
 Options de print:
                                                                               □ accès aux éléments autour de la position (avant/après)
                                                                                                                                                : ne l
 □sep=" "
                             séparateur d'éléments, défaut espace
                                                                               lst = [11, 18, 9, 12, 23, 4, 17]
 □ end="\n"
                             fin d'affichage, défaut fin de ligne
                                                                               perdu = []
                                                                                                                                                habitude
                                                                                                                         Algo: bornage des
 □ file=sys.stdout
                            print vers fichier, défaut sortie standard
                                                                               for idx in range(len(lst)):
                                                                                                                         valeurs supérieures à 15,
                                                                                    val = lst[idx]
                                                                     Saisie
                                                                                                                         mémorisation des
 s = input("Directives:")
                                                                                     if val > 15:
                                                                                                                         valeurs perdues.
                                                                                                                                                ponne
   input retourne toujours une chaîne, la convertir vers le type désiré
                                                                                         perdu.append(val)
                                                                               lst[idx] = 15
print("modif:",lst,"-modif:",perdu)
        (cf. encadré Conversions, en partie 1).
                             Opérations génériques sur conteneurs
len (c) → nb d'éléments
                                                                               Parcours simultané index et valeurs de la séquence :
min(c) max(c) sum(c)
                                       Note: pour dictionnaires et ensembles,
                                                                               for idx, val in enumerate(lst):
                                       ces opérations travaillent sur les clés.
sorted(c) → list copie triée
                                                                                 range ([début,] fin [,pas])
                                                                                                                       Séquences d'entiers
val in c → booléen, opérateur in de test de présence (not in d'absence)
enumerate(c) → itérateur sur (index, valeur)
                                                                                début défaut 0, fin non compris, pas signé et défaut 1
zip (c1, c2...) → itérateur sur tuples contenant les éléments de même index des c
                                                                                range (5) \rightarrow 0 1 2 3 4
                                                                                                            range (2, 12, 3) \rightarrow 25811
all (c) → True si tout élément de c évalué vrai, sinon False
                                                                                range (3,8) \rightarrow 3 4 5 6 7 range (20,5,-5) \rightarrow 20 15 10
any (c) → True si au moins un élément de c évalué vrai, sinon False
                                                                                range (len (s\acute{e}q)) \rightarrow séquence des index des valeurs dans séq
Spécifique aux conteneurs de séquences ordonnées (listes, tuples, chaînes, bytes...)
                                                                                arange fournit une séquence immutable d'entiers construits au besoin
reversed (c) → itérateur inversé c*5→ duplication c+c2→ concaténation
                                                                                                                       Définition de fonction
                                                                                nom de la fonction (identificateur)
c.index (val) \rightarrow position
                                  c.count (val) \rightarrow nb d'occurences
                                                                                           paramètres nommés
import copy
copy.copy (c) → copie superficielle (1er niveau) du conteneur
                                                                                def fct(x,y,z):
                                                                                                                                        fct
copy.deepcopy(c) → copie en profondeur (récursive) du conteneur
                                                                                       """documentation"""
                                                                                       # bloc instructions, calcul de res, etc.
modification de la liste originale
                                                  Opérations sur listes
                                                                                       return res ← valeur résultat de l'appel, si pas de résultat
                             ajout d'un élément à la fin
lst.append(val)
                                                                                calculé à retourner : return None
                             ajout d'une séquence d'éléments à la fin
lst.extend(seq)
                             insertion d'un élément à une position
lst.insert(idx, val)
                                                                                variables de ce bloc n'existent que dans le bloc et pendant l'appel à la
lst.remove(val)
                             suppression du premier élément de valeur val
                                                                                fonction (penser "boîte noire")
1st. pop ([idx]) \rightarrow valeur supp. & retourne l'item à l'index (sinon le dernier)
                                                                                Avancé: def fct(x,y,z,*args,a=3,b=5,**kwargs):
lst.sort() lst.reverse() tri / inversion de la liste sur place
                                                                                  *args \rightarrow nb variables d'arguments positionnels (tuple), a=3 \rightarrow valeurs
                                                                                 par défaut, **kwargs → nb variable d'arguments nommés (dict).
   Opérations sur dictionnaires
                                             Opérations sur ensembles
                                                                                 r = fct(3, i+2, 2*i)
                                                                                                                            Appel de fonction
                      d.clear()
d[cl\acute{e}] = valeur
                                        Opérateurs :
                                                                               stockage/utilisation une valeur d'argument
                      del d[clé]
                                        | → union (caractère barre verticale)
d[cl\acute{e}] \rightarrow valeur
                                                                               de la valeur de retour par paramètre
d. update (d2) { mise à jour/ajout des couples
                                        \& \rightarrow intersection
                                                                                                              Avancé:
                                                                                                                                           fct
                                                                               - ^ → différence/diff. symétrique
d.keys()
d.values()
d.items()

d.items()

d.sec couples

→ vues itérables sur les
clés / valeurs / couples
                                                                                                               *séauence
                                                                                la fonction avec les paren-
                                        < <= > =\rightarrow relations d'inclusion
                                                                                thèses qui fait l'appel
                                        Existent aussi sous forme de méthodes.
d.pop(clé[,défaut]) \rightarrow valeur
                                                                               s.startswith (prefix[,débul[,fin]]) Opérations sur chaînes
                                        s.update(s2) s.copy()
d.popitem() \rightarrow (clé, valeur)
                                        s.add(clé) s.remove(clé)
                                                                               s.endswith(suffix[,début[,fin]]) s.strip([caractères])
d.get(clé[,défaut]) \rightarrow valeur
                                        s.discard(clé) s.clear()
                                                                               s.count(sub[,d\acute{e}but[,fin]]) s.partition(sep) \rightarrow (avant,sep,apr\grave{e}s)
d.setdefault (clé[,défaut]) →valeur
                                        s.pop()
                                                                               s.index(sub[,début[,fin]]) s.find(sub[,début[,fin]])
                                                                  Fichiers
                                                                               s.is...() tests sur les catégories de caractères (ex. s.isalpha())
stockage de données sur disque, et relecture
                                                                               s.upper() s.lower() s.title() s.swapcase()
     f = open("fic.txt", "w", encoding="utf8")
                                                                               s.casefold() s.capitalize() s.center([larg,rempl])
variable
               nom du fichier
                                 mode d'ouverture
                                                          encodage des
                                                                               s.ljust([larg,rempl]) s.rjust([larg,rempl]) s.zfill([larg])
                                 □ 'r' lecture (read)
                                                          caractères pour les
               sur le disque
                                                                               s.encode (codage)
                                                                                                      s.split([sep]) s.join(séq)
fichier pour
                                 □ 'w' écriture (write)
                                                          fichiers textes:
les opérations (+ chemin...)
les operations (+ cnemin...) | 'a' ajout (append) | utf8 | a | Cf. modules os,os.path.pathlib | ...'+' 'x' 'b' 't' | latin1
                                                                               directives de formatage
                                                                                                              valeurs à formater
                                                                  ascii
                                                                                "modele{} {} {}".format(x,y,r)—
en écriture
                                🖠 lit chaîne vide si fin de fichier
                                                                  en lecture
                                                                                "{sélection: formatage!conversion}"
                                                       → caractères suivants
                                f.read([n])
f.write("coucou")
                                      si n non spécifié, lit jusqu'à la fin!
                                                                                □ Sélection :
f.writelines(list lignes)
                                                                                                            "{:+2.3f}".format(45.72793)
                                f.readlines ([n]) \rightarrow list lignes suiv.
                                                                                  2
                                                                                                            →'+45.728'
                                                                                                      Exemples
                                                       \rightarrow ligne suivante
                                f.readline()
                                                                                   nom
                                                                                                           "{1:>10s}".format(8, "toto")
                                                                                   0.nom
      🖢 par défaut mode texte 🕇 (lit/écrit str), mode binaire b possible
                                                                                                           \rightarrow '
                                                                                                                      toto'
                                                                                   4[clé]
      (lit/écrit bytes). Convertir de/vers le type désiré!
                                                                                                           "{x!r}".format(x="L'ame")
                                                                                   0[2]
                                                                                                          →'"L\'ame"'
f.close() ½ ne pas oublier de refermer le fichier après son utilisation!
                                                                                □ Formatage :
f.flush() écriture du cache
                                  f.truncate([taille]) retaillage
                                                                                <u>car-rempl.</u> <u>alignement signe larg.mini précision~larg.max</u> <u>type</u>
\slashed lecture/écriture progressent séquentiellement dans le fichier, modifiable avec :
                                                                                     <> ^{\circ} = + - espace 0 au début pour remplissage avec des 0
                                   f.seek (position[,origine])
                                                                                entiers : b binaire, c caractère, d décimal (défaut), o octal, x ou X hexa...
 Très courant : ouverture en bloc gardé (ferme- with open (...) as f:
                                                                                flottant : e ou E exponentielle, f ou F point fixe, g ou G approprié (défaut),
 ture automatique) et boucle de lecture des
                                                 for ligne in f :
                                                                                chaîne : s ...
                                                                                                                                 % pourcentage
                                                    # traitement de ligne
 lignes d'un fichier texte:
                                                                                □ Conversion : s (texte lisible) ou r (représentation littérale)
```