Aide-mémoire Python - v2.4

1. **Fonctions**
   - int(x) : convertit x, de type float ou str, en entier
   - float(x) : convertit x, int ou str, en réel
   - str(x) : convertit x, int ou float, en str
   - list(x) : convertit x en list
   - tuple(x) : convertit x en tuple
   - help(x) : aide sur x
   - dir(x) : liste des attributs de x
   - type(x) : type de x
   - print(x) : imprime x à l'écran
   - input(x) : imprime le string x et lit le string qui est introduit au clavier
   - round(x) : valeur arrondi du float x
   - len(s) : longueur de la séquence s
   - range([start], stop, [step]) : retourne une suite arithmétique d'entiers
2. **Modules :**
   - math : accès aux constantes et fonctions mathématique (pi, sin(), sqrt(x), exp(x),floor(x) (valeur plancher), ceil(x) (valeur plafond), ...) : exemple : math.ceil(x)
   - copy : copy(s), deepcopy(s) : "shallow" et "deepcopy" de s
   - pickle :
     - dumps(v) : transforme v en une représentation,
     - loads(r) : reconstitue l'objet
     - dump(v,f) : transforme et écrit dans le fichier f
     - load(f) : 'reconstitue à partir de la représentation lue de f
   - shelve
     - db = open() : créer un fichier comme objet de type shelve
     - db.close() : fermeture
3. **Opérations et méthodes sur les séquences (str, list, tuples) :**
   - min(s),max(s) : élément minimum, maximum
   - sum(s) : (ne fonctionne pas pour les string) : somme de tous les éléments (valeur numérique)
   - s.index(value, [start, [stop]]) : premier indice de value dans s[start :stop]
   - s.count(sub [,start [,end]]) : le nombre d'occurrences sans chevauchement de sub dans s[start :end]
   - map(f,s) : créer une liste où chaque élément i de s est remplacé par f(i)
   - filter(f,s) : créer une séquence du même type que s avec les éléments i de s tel que f(i) ou i.f() est vrai
   - reduce(f,s) : applique f() deux à deux aux éléments consécutifs de s (de gauche à droite) et renvoie le résultat
4. **Méthodes sur les str :**
   - s.lower() : met en minuscule
   - s.upper() : met en majuscule
   - s.islower(), s.isdigit(), s.isalnum(),s.isalpha(), s.isupper() : vrai si dans on a (respectivement) des minuscules, des chiffres, des car. alphanumériques, alphabétiques, majuscules
   - s.find(sub [,start [,end]]) : premier indice de s où le sous string sub est trouvé dans s[start :end]
   - s.replace(old, new[, co]) : retourne une copie de s en remplaçant toutes les (ou les co premières) occurrences de old par new.
   - s.format(...) : sert au formatage (en particulier) d'output
   - s.capitalize() : met la première lettre en majuscule
   - s.strip() : copie de s en retirant les "blancs" en début et fin
   - s.join(t) : créer un str qui est le résultat de la concaténation des éléments de t chacun

séparé par le str s
  - s.split([sep [,maxsplit]) : renvoie une liste d'éléments séparés dans s par le caractère sep (par défaut "blanc") ; au max maxsplit séparations sont faites
5. **Opérateurs et méthodes sur les listes s :**
  - s.append(v) : ajoute un élément valant v à la fin de la liste
  - s.extend(s2) : rajoute à s tous les éléments de la liste s2
  - s.insert(i,v) : insert l'objet v à l'indice i
  - s.pop() : supprime le dernier élément de la liste et retourne l"élément supprimé
  - s.remove(v) : supprime la première valeur v dans s
  - s.reverse() : retourne la liste, le premier et dernier élément échange leurs places, le second et l'avant dernier, et ainsi de suite
  - s.sort(cmp=None, key=None, reverse=False) : trie s
  - del s[i], del s[i :j] : supprime un ou des éléments de s
  - zip (a,b,c) : construit une liste de des triples dont le ième élément reprend le ième élément de chaque séquence a,b,c
  - it=iter(s) : créé un itérateur
  - it.next() : élément suivant de l'itérateur s'il existe, exception StopIteration sinon
6. **Méthodes sur les dict :**
  - d.clear() : supprime tous les élément de d
  - d.copy() : "shallow" copie de d
  - d.fromkeys(s,v) : créer un dict avec les clés de s et valeur v
  - d.get(k [,v]) : renvoie la valeur d[k] si elle existe v sinon
  - d.has_key(k) : vrai si d[k] existe
  - d.items() : liste des items (k,v) de d
  - d.iteritems(), d.iterkeys(), d.itervalues() : cf items,keys,values, mais retourne un itérateur (pas de création de liste)
  - d.keys() : liste des clés
  - d.pop(k [,v]) : enlève d[k] et renvoie sa valeur ou v
  - d.popitem() : supprimer un item (k,v) et retourne l'item sous forme de tuple
  - d.setdefault(k [,v]) : d[k] si elle existe sinon v et rajoute d[k]=v
  - d.update(s) : s est une liste de tuples que l'on rajoute à d
  - d.values() : liste des valeurs de d
7. **Méthodes sur les fichiers :**
  - f=open('fichier') : ouvre 'fichier' en lecture
  - f=open('fichier','w') : ouvre 'fichier' en écriture
  - f=open('fichier','a') : ouvre 'fichier' en écriture en rajoutant après les données déjà présentes
  - f.read() : retourne le contenu du fichier f
  - f.readline() : lit une ligne
  - f.readlines() : renvoie la liste des lignes de f
  - f.write(s) : écrit la chaîne de caractères s dans le fichier f
  - f.close() : ferme f
8. **Exceptions :**
  -
```
try:
    ...
    raise ...
    ...
except:
    ...
else:
    ...
finally:
    ...
```