

<u>Les fonctions</u>	<u>La sélection(if...else....)</u>	<u>La récursivité</u>
def nomfonction(P1,P2,...,Pn): 	if cond: 	def nomrec(P1,P2,...,Pn): nomrec(P11,P21,...,Pn1)

<u>la boucle tant que</u>	<u>la boucle pour</u>	<u>la boucle pour</u>
while cond: 	for i in range(vi,vf,pas): 	for x in T: # T est une variable composée

<u>Les exceptions</u>	<u>La POO</u>
try : except : 	Class Nomclasse:

<u>Les listes(L)</u>	
<u>Méthode</u>	<u>Signification</u>
L.append(N)	permet d'ajouter l'élément 'N' en fin de liste
L.extend(L)	permet d'ajouter les éléments d'une liste 'L' à la liste à laquelle la méthode est appliquée
L.insert(pos,e)	permet d'insérer un élément 'e' à une position 'pos' donnée
L.remove(e)	permet de supprimer le premier élément de la liste (donc en partant de 0) dont la valeur est 'e'.
L.count(e)	renvoie le nombre d'occurrences de la valeur 'e'.
L.reverse()	permet d'inverser une liste.
<u>L.index(e)</u>	renvoie l'index du premier élément dont la valeur est 'e'. Si la valeur ne figure pas dans la liste, une erreur est renvoyée.
L.sort()	permet de trier une liste.
.....

<u>Les ensembles(A,B)</u>		
<u>Notation mathématique</u>	<u>Notation Python</u>	<u>Notation Python</u>
$A \subseteq B$	A<=B	A.issubset(B)
$B \subseteq A$	B<=A	A.issuperset(B)
$A \cup B$	A B	A.union(B)
$A \cap B$	A&B	A.intersection(B)
$A \setminus B$	A-B	A.difference(B)
$A \Delta B$	A^B	A.symmetric_difference(B)
$x \in A$		x in A
		A.add(x) #ajout d'élément
		A.remove(x) # suppression d'élément
		A.pop() #extraction d'un élément
		A.copy() # copie un ensemble dans un autre
		A.clear() #vide l'ensemble

Les dictionnaires(D)	
Méthode	Signification
D.get(cle, valdef)	renvoie la valeur associée à clé, valdef si clé absente
D.items()	renvoie une copie de la liste des paires (clé, valeur)
D.keys()	renvoie une copie de la liste des clés
D.values()	renvoie une copie de la liste des valeurs
.....

Les chaines de caractères(S)	
Méthode	Signification
S.upper	return une chaine en majuscules
S.lower	return une chaine en minuscules
S.isupper	return True si ch ne contient que des majuscules
S.islower	return True si ch ne contient que des minuscules
S.isalnum	return True si ch ne contient que des alphanumériques
S.isalpha	return True si ch ne contient que des alphabétiques
S.isdigit	return True si ch ne contient que des numériques
S.isspace	return True si ch ne contient que des espace
S.strip(chars)	supprime toute les combinaisons de chars (ou l'espace par défaut) au début et en fin
.....

Les fichiers(f)	
Méthode	Signification
f=open('...', 'r')	permet d'ouvrir ou de créer un fichier
f.read(n)	permet de lire n caractères
f.read()	permet de lire la totalité du fichier
f.readline()	permet de lire et renvoie une ligne de texte .
f.readlines()	permet de lire et renvoie la liste de toutes les lignes de texte.
f.write(chaine)	permet d'écrire une chaîne de caractères dans un fichier texte.
info=load(f)	permet de lire une information à partir d'un fichier binaire.
dump(info,f)	permet d'écrire une information dans un fichier binaire.
f.close()	permet de fermer un fichier .
.....

Les base de données	
Méthode	Signification
connect()	pour se connecter à une base de données.
cursor()	les curseurs sont utilisés pour les interaction avec la base de données (émission de requetes SQL)
execute()	qui permet l'évaluation d'une requete SQL
executescript("... ")	qui permet l'évaluation d'un script SQL
fetchall()	permet de récupérer toutes les valeurs
fetchone()	permet de récupérer une valeur
close()	permet de fermer une connexion
commit()	valide une transaction avec la base de données
rollback()	annule une transaction avec la base de données(cette méthode annule toutes les modifications apportées à la base de données depuis le dernier appel à commit())
.....

1. Les commandes SQLite

1.1 Les Commandes de manipulation de tables :

La structure	
CREATE TABLE nom_Table(...)	création d'une table
DROP TABLE nom_Table	suppression d'une table
ALTER TABLE nom_Table ADD...	modification du schéma (ajouter une colonne)
ALTER TABLE nom_Table RENAME TO ...	modification du schéma (renommer une table)
Les données	
INSERT INTO nom_Table VALUES (...)	Ajout d'une ligne
INSERT INTO nom_Table SELECT ...	Ajout de lignes
DELETE FROM nom_Table WHERE ...	suppression d'une ligne
UPDATE nom_Table SET ... WHERE ...	Modification d'un attribut
SELECT attributs FROM ...	Chercher des attributs d'un enregistrement
SELECT * FROM...	Chercher tous les attributs d'un enregistrement
ORDER BY [DESC]	Ordonner l'affichage selon un attribut

SELECT attributs FROM table WHERE conditions

- Les conditions peuvent être constituées :
 - Opérateurs de comparaison : =, <, >, <=, etc...
 - Connecteurs logiques : AND, OR, NOT, XOR et parenthèses.
 - Prédicat ensemblistes : EXISTS(), IN(), Between
 - Expressions régulières : LIKE chaîne de caractère, avec :
 - '-' remplace n'importe quel caractère
 - '%' remplace n'importe quelle chaîne de caractère
- On peut aussi ajouter : ORDER BY attribut [DESC]

Select liste des champs /*

From liste des tables

Where condition portant sur les champs

Group by liste des champs regroupement

Having conditions regroupement

Order by liste critères de

Limit v_i, nbr

1.2 Rappel(Opérateurs ensemblistes)

Opération	Syntaxe(algèbre relationnelle)	Syntaxe(SQL)
L'union	$A \cup B$	SELECT * FROM A UNION SELECT * FROM B
L'intersection	$A \cap B$	SELECT * FROM A INTERSECT SELECT * FROM B
La différence	$A \setminus B$	SELECT * FROM A where not exists(SELECT * FROM B Where A _i =B _i)
La Projection	$\Pi_{p_1, \dots, p_2} (A)$	SELECT p ₁ , ..., p ₂ FROM A
La Sélection	$\sigma_{p='...'} (A)$	SELECT * FROM A where p='...'
Le produit cartésien	$A \times B$	SELECT * FROM A ,B
La jointure Conditionnelle	$A \bowtie_{p='...'} B$	SELECT * FROM A ,B Where p='...'
La jointure naturelle	$A \bowtie B$	SELECT * FROM A ,B Where A.num=B.num <i>Ou bien</i> SELECT * FROM A join B on A.num=B.num <i>Ou bien</i> SELECT * FROM A natural join B
La division	$A \div B$	SELECT Xi FROM A R where not exists(SELECT Yj FROM B where not exists(Select * from A where A.Xi=R.Xi and A.Yj=B.Yj))

1.3 Les principales fonctions d'agrégation :

COUNT	pour compter le nombre d'enregistrements distincts d'une table
AVG	valeur moyenne d'une colonne de type numérique d'une table
MIN	valeur minimale d'une colonne de type numérique d'une table
MAX	valeur maximale d'une colonne de type numérique d'une table
SUM	somme d'une colonne de type numérique d'une table

1.4 Accéder à un SGBD depuis un programme

1 Schéma classique d'accès à un SGBD à partir d'un langage de programmation quelconque :

1. Récupérer une connexion à la base
2. Exécuter une (ou des) requête(s) en utilisant cette connexion
3. Récupérer le résultat des requêtes (et le traiter)
4. Fermer la connexion

Les base de données-python	
Méthode	Signification
connect()	pour se connecter à une base de données.
cursor()	les curseurs sont utilisés pour les interaction avec la base de données (émission de requetes SQL)
execute()	qui permet l'évaluation d'une requete SQL
executescript("... ")	qui permet l'évaluation d'un script SQL
fetchall()	permet de récupérer toutes les valeurs
fetchone()	permet de récupérer une valeur
close()	permet de fermer une connexion
commit()	valide une transaction avec la base de données
rollback()	annule une transaction avec la base de données(cette méthode annule toutes les modifications apportées à la base de données depuis le dernier appel à commit())
.....

Exemples :

Affichage à l'écran	Enregistrer dans un fichier	Création d'une vue
<pre>from sqlite3 import * def afficher(db,req): conn= connect(db) cur=conn.cursor() cur.execute(req) res= cur.fetchall() for ligne in res: print(ligne) conn.close()</pre>	<pre>def enregistrer (db,req,fichier): conn= connect(db) cur=conn.cursor() cur.execute(req) res= cur.fetchall() f=open(fichier,"w") for ligne in res: f.write(str(ligne)+"\n") f.close() conn.close()</pre>	<pre>def creervue(db,req): conn= connect(db) cur=conn.cursor() cur.execute(req) conn.close()</pre>