

POO sous Python

Exercices concernant le contenu des deux fichiers ppt du Rappel

Enseignant : GHANEM

Les fonctions principales pour chaque type

Fonctions pour les entiers (int)

Fonction	Description
<code>int("25")</code>	Convertit une chaîne en entier
<code>abs(-5)</code>	Retourne la valeur absolue
<code>pow(2, 3)</code>	Calcule 2 puissance 3
<code>divmod(10, 3)</code>	Retourne (quotient, reste)
<code>bin(10)</code>	Convertit en binaire
<code>hex(255)</code>	Convertit en hexadécimal

Fonctions pour les flottants (float)

Fonction	Description
<code>float("3.14")</code>	Convertit en nombre décimal
<code>round(3.14159, 2)</code>	Arrondit à 2 décimales
<code>abs(-2.5)</code>	Retourne la valeur absolue
<code>max(1.5, 2.7)</code>	Retourne le plus grand
<code>min(1.5, 2.7)</code>	Retourne le plus petit
<code>int(3.99)</code>	Convertit en entier (troncature)

Fonctions pour les complexes (complex)

Fonction	Description
<code>complex(2, 3)</code>	Crée $2+3j$
<code>z.real</code>	Retourne la partie réelle
<code>z.imag</code>	Retourne la partie imaginaire
<code>z.conjugate()</code>	Retourne le conjugué
<code>abs(z)</code>	Retourne le module

Fonctions pour les chaînes (str)

Fonction	Description
s.upper()	Convertit en majuscules
s.lower()	Convertit en minuscules
s.strip()	Supprime les espaces
s.split(",")	Sépare en liste
s.replace("a", "b")	Remplace a par b
s.find("mot")	Cherche la position
len(s)	Retourne la longueur
s.startswith("A")	Vérifie si commence par A

Fonctions pour les listes (list)

Fonction	Description
<code>liste.append(5)</code>	Ajoute un élément
<code>liste.extend([6,7])</code>	Ajoute plusieurs éléments
<code>liste.insert(0, "a")</code>	Insère à une position
<code>liste.remove(5)</code>	Supprime un élément
<code>liste.pop()</code>	Supprime le dernier
<code>liste.sort()</code>	Trie la liste
<code>liste.reverse()</code>	Inverse l'ordre
<code>len(liste)</code>	Retourne la longueur

Fonctions pour les tuples (tuple)

Fonction	Description
<code>tuple([1,2,3])</code>	Convertit en tuple
<code>t.count(5)</code>	Compte les occurrences
<code>t.index(3)</code>	Trouve la position
<code>len(t)</code>	Retourne la longueur
<code>t + (4,5)</code>	Concatène des tuples
<code>max(t)</code>	Retourne le maximum
<code>min(t)</code>	Retourne le minimum

Fonctions pour les dictionnaires (dict)

Fonction	Description
<code>d.keys()</code>	Retourne les clés
<code>d.values()</code>	Retourne les valeurs
<code>d.items()</code>	Retourne les paires
<code>d.get("clé")</code>	Retourne la valeur
<code>d.pop("clé")</code>	Supprime une clé
<code>d.update({"a":1})</code>	Ajoute/modifie
<code>len(d)</code>	Retourne le nombre
<code>"clé" in d</code>	Vérifie si clé existe

Fonctions pour les ensembles (set)

Fonction	Description
<code>s.add(5)</code>	Ajoute un élément
<code>s.remove(5)</code>	Supprime un élément
<code>s.union(t)</code>	Retourne l'union
<code>s.intersection(t)</code>	Retourne l'intersection
<code>s.difference(t)</code>	Retourne la différence
<code>len(s)</code>	Retourne la taille
<code>5 in s</code>	Vérifie si présent
<code>s.clear()</code>	Vide l'ensemble

Fonctions universelles

Fonction	Description
<code>type(x)</code>	Retourne le type
<code>len(x)</code>	Retourne la longueur
<code>max(x)</code> exemple : <code>max (5,7,8)</code>	Retourne le maximum
<code>min(x)</code>	Retourne le minimum
<code>sorted(x)</code> <code>sorted([5,7,0,8])</code>	Retourne trié
<code>sum(x)</code> <code>sum([5,7,0,8])</code>	Calcule la somme
<code>any(x)</code>	True si un élément vrai
<code>all(x)</code>	True si tous vrais

- **Exercice 1**
- Créez une variable temperature avec la valeur 23.5 et affichez-la

- **Solution de l'exercice 1**

- temperature = 23.5

- print(temperature)

- **Exercice 2**
- Convertissez la chaîne "125" en entier et stockez-la dans nombre

- **Solution de l'exercice 2**

- `nombre = int("125")`

- `print(nombre)`

- **Exercice 3**
- Concaténez votre prénom et nom avec un espace entre les deux

- **Solution de l'exercice 3**

- `prenom = "Ahmed"`
- `nom = "Bennaceur"`
- `nom_complet = prenom + " " + nom`
- `print(nom_complet)`

- **Exercice 4**
- Créez un nombre complexe $z = 4 - 3j$ et affichez son conjugué

- **Solution de l'exercice 4**

- $z = 4 - 3j$

- `conjugue = z.conjugate()`

- `print(conjugue)`

- **Exercice 5**
- Convertissez l'entier 42 en float et affichez le résultat

- **Solution de l'exercice 5**
- `nombre_float = float(42)`
- `print(nombre_float)`

- **Exercice 6**

- Créez une liste mixte donnees contenant [10, "python", 3.14, True]

- **Solution de l'exercice 6**

- `donnees = [10, "python", 3.14, True]`
- `print(donnees)`

- **Exercice 7**

- À partir de nombres = [1, 2, 3, 4, 5], extrayez les éléments 2 et 3 dans une sous-liste

- **Solution de l'exercice 7**

- `nombre = [1, 2, 3, 4, 5]`
- `sous_liste = nombre[1:3] # Index 1 et 2 (2 et 3)`
- `print(sous_liste)`

- **Exercice 8**

- Créez une liste couleurs avec "rouge", "vert", "bleu", puis remplacez "vert" par "jaune"

- **Solution de l'exercice 8**

- `couleurs = ["rouge", "vert", "bleu"]`
- `couleurs[1] = "jaune"`
- `print(couleurs)`

- **Exercice 9**

- À partir de `liste1 = [1, 2]` et `liste2 = [3, 4]`, créez une liste `liste3` qui les combine

- **Solution de l'exercice 9**

- `liste1 = [1, 2]`
- `liste2 = [3, 4]`
- `liste3 = liste1 + liste2`
- `print(liste3)`

- **Exercice 10**
- Inversez l'ordre des éléments dans la liste lettres = ['a', 'b', 'c', 'd']

- **Solution de l'exercice 10**

- lettres = ['a', 'b', 'c', 'd']
- lettres_inversees = lettres[::-1]
- print(lettres_inversees)

- **Exercice 11**
- Créez un tuple coordonnees avec les valeurs (48.8566, 2.3522) et affichez la latitude

- **Solution de l'exercice 11**
- `coordonnees = (48.8566, 2.3522)`
- `latitude = coordonnees[0]`
- `print(latitude)`

- **Exercice 12**

- Créez un ensemble voyelles avec 'a', 'e', 'i', 'o', 'u' et ajoutez 'y'

- **Solution de l'exercice 12**

- `voyelles = {'a', 'e', 'i', 'o', 'u'}`
- `voyelles.add('y')`
- `print(voyelles)`

- **Exercice 13**

- À partir de $\text{set1} = \{1, 2, 3\}$ et $\text{set2} = \{3, 4, 5\}$, créez leur union

- **Solution de l'exercice 13**

- `set1 = {1, 2, 3}`
- `set2 = {3, 4, 5}`
- `union_set = set1.union(set2)`
- `print(union_set)`

- **Exercice 14**
- Vérifiez si la valeur 5 est présente dans l'ensemble nombres = {1, 3, 5, 7}

- **Solution de l'exercice 14**

- nombres = {1, 3, 5, 7}
- if 5 in nombres:
- print("5 est présent")
- else:
- print("5 n'est pas présent")

- **Exercice 15**

- Créez un dictionnaire capitales avec "Maroc":"Rabat", "Gabon":"Libreville", "Cameroun":"Yaoundé"

- **Solution de l'exercice 15**

- capitales = {
- "Maroc":"Rabat",
- "Gabon":"Libreville",
- "Cameroun":"Yaoundé"}
- print(capitales)

- **Exercice 16**

- À partir de `personne = {"nom": "Rym", "age": 30}`, ajoutez la clé "ville" avec valeur "Rabat"

- **Solution de l'exercice 16**

- `personne = {"nom": "Rym", "age": 30}`
- `personne["ville"] = "Rabat"`
- `print(personne)`

- **Exercice 17**

- Affichez toutes les clés du dictionnaire produit = {"nom": "laptop", "prix": 999, "stock": 5}

- **Solution de l'exercice 17**

- `produit = {"nom": "laptop", "prix": 999, "stock": 5}`
- `cles = produit.keys()`
- `print(cles)`

- **Exercice 18**

- Supprimez la clé "note" du dictionnaire etudiant = {"nom": "Jihane", "age": 20, "note": 15}

- **Solution de l'exercice 18**

- `etudiant = {"nom": "Jihane", "age": 20, "note": 15}`
- `del etudiant["note"]`
- `print(etudiant)`

- **Exercice 19**

- Calculez le résultat de 17 divisé par 3 et affichez le quotient entier et le reste

- **Solution de l'exercice 19**

- `quotient = 17 // 3`
- `reste = 17 % 3`
- `print(f"Quotient: {quotient}, Reste: {reste}")`

- **Exercice 20**
- Calculez 5 à la puissance 3 en utilisant l'opérateur **

- **Solution de l'exercice 20**

- `resultat = 5 ** 3`

- `print(resultat)`

- **Exercice 21**
- Utilisez l'opérateur modulo pour vérifier si 28 est pair

- **Solution de l'exercice 21**

- if `28 % 2 == 0`:
- `print("28 est pair")`
- else:
- `print("28 est impair")`

- **Exercice 22**
- Calculez la moyenne de 15, 18 et 20

- **Solution de l'exercice 22**

- `moyenne = (15 + 18 + 20) / 3`

- `print(moyenne)`

- **Exercice 23**

- Demandez à l'utilisateur son âge et affichez "Vous avez X ans" où X est l'âge saisi

- **Solution de l'exercice 23**

- `age = input("Quel est votre âge ? ")`
- `print(f"Vous avez {age} ans")`

- **Exercice 24**
- Utilisez `print` avec `sep="-"` pour afficher "2023", "09", "15" séparés par des tirets

- **Solution de l'exercice 24**
- `print("2023", "09", "15", sep="-")`

- **Exercice 25**
- Affichez "Bonjour" et "Monde" sur deux lignes différentes en un seul print

- **Solution de l'exercice 25**
- `print("Bonjour\nMonde")`

- **Exercice 26**
- Utilisez format pour afficher "Le résultat est 12.3456" avec seulement 2 décimales

- **Solution de l'exercice 26**
- nombre = 12.3456
- `print("Le résultat est {:.2f}".format(nombre))`

- **Exercice 27**
- Vérifiez si un nombre est positif, négatif ou nul et affichez le message correspondant

- **Solution de l'exercice 27**

- nombre = -5
- if nombre > 0:
- print("Positif")
- elif nombre < 0:
- print("Négatif")
- else:
- print("Nul")

- **Exercice 28**
- Vérifiez si une année est bissextile (divisible par 4 mais pas par 100, ou divisible par 400)

- **Solution de l'exercice 28**

- `annee = 2024`
- `if (annee % 4 == 0 and annee % 100 != 0) or (annee % 400 == 0):`
- `print(f"{annee} est bissextile")`
- `else:`
- `print(f"{annee} n'est pas bissextile")`

- **Exercice 29**
- Affichez "OK" si temperature est entre 18 et 25 degrés, sinon "Hors plage"

- **Solution de l'exercice 29**
- temperature = 22
- if 18 <= temperature <= 25:
- print("OK")
- else:
- print("Hors plage")

- **Exercice 30**
- Vérifiez si une chaîne est vide et affichez "Vide" ou "Non vide"

- **Solution de l'exercice 30**

- `texte = "Bonjour"`
- `if texte == "":`
- `print("Vide")`
- `else:`
- `print("Non vide")`

- **Exercice 31**
- Affichez tous les nombres de 1 à 10 à l'aide d'une boucle for avec range

- **Solution de l'exercice 31**

- for i in range(1, 11):

- print(i)

- **Exercice 32**
- Parcourez la liste `pays = ["Sénégal", "Cote d'ivoire", "Gabon"]` et affichez chaque élément

- **Solution de l'exercice 32**

- `pays = ["Sénégal", "Cote d'ivoire", 88.6 , "Gabon"]`
- `for p in pays:`
- `print(p)`

- **Exercice 33**
- Calculez la somme de tous les éléments de nombres = [5, 10, 15, 20]

- **Solution de l'exercice 33**

- `nombre = [5, 10, 15, 20]`
- `somme = 0`
- `for n in nombre:`
 - `somme += n`
- `print(somme)`

- **Exercice 34**
- Affichez chaque caractère de la chaîne "python" sur une ligne différente

- **Solution de l'exercice 34**

- for lettre in "python":

- print(lettre)

- **Exercice 35**
- Affichez les nombres de 10 à 1 en ordre décroissant avec while

- **Solution de l'exercice 35**

- `i = 10`
- `while i >= 1:`
- `print(i)`
- `i -= 1`

- **Exercice 36**
- Utilisez `while` pour trouver le premier nombre divisible par 7 et 3 après 50

- **Solution de l'exercice 36**

- nombre = 51
- while True:
- if nombre % 7 == 0 and nombre % 3 == 0:
- print(f"Trouvé: {nombre}")
- break
- nombre += 1

- **Exercice 37**
- Créez une boucle qui s'arrête quand l'utilisateur tape "stop"

- **Solution de l'exercice 37**

- while True:
- entree = input("Tapez 'stop' pour arrêter: ")
- if entree.lower() == "stop":
- break

- **Exercice 38**
- Créez une fonction `maximum(a, b)` qui retourne le plus grand des deux nombres

- **Solution de l'exercice 38**

- `def maximum(a, b):`
 - `if a > b:`
 - `return a`
 - `else:`
 - `return b`
- `print(maximum(5, 8))`

- **Exercice 39**
- Créez une fonction `est_majeur(age)` qui retourne `True` si $\text{age} \geq 18$

- **Solution de l'exercice 39**

- `def est_majeur(age):`

- `return age >= 18`

- `print(est_majeur(20))`

- **Exercice 40**

- Créez une fonction `liste_premiers(n)` qui retourne les nombres de 1 à n

- **Solution de l'exercice 40**
- `def liste_premiers(n):`
 - `return list(range(1, n+1))`
- `print(liste_premiers(5))`

- **Exercice 41**
- Créez une fonction `compte_caracteres(texte)` qui retourne la longueur d'une chaîne

- **Solution de l'exercice 41**
- `def compte_caracteres(texte):`
 - `return len(texte)`
- `print(compte_caracteres("python"))`

- **Exercice 42**
- Vérifiez si "python" est présent dans la chaîne "J'apprends python"

- **Solution de l'exercice 42**
- `texte = "J'apprends python"`
- `if "python" in texte:`
- `print("Présent")`
- `else:`
- `print("Absent")`

- **Exercice 43**
- Remplacez tous les espaces par des underscores dans "hello world python"

- **Solution de l'exercice 43**

- `texte = "hello world python"`
- `nouveau_texte = texte.replace(" ", "_")`
- `print(nouveau_texte)`

- **Exercice 44**
- Séparez la chaîne "nom,prenom,age" en une liste utilisant la virgule comme séparateur

- **Solution de l'exercice 44**

- `chaine = "nom,prenom,age"`
- `liste = chaine.split(",")`
- `print(liste)`

- **Exercice 45**
- Convertissez "bonjour tout le monde" en majuscules

- **Solution de l'exercice 45**
- `texte = "bonjour tout le monde"`
- `texte_maj = texte.upper()`
- `print(texte_maj)`

- **Exercice 46**
- Comptez combien de fois le nombre 5 apparaît dans [2, 5, 8, 5, 1, 5]

- **Solution de l'exercice 46**

- `liste = [2, 5, 8, 5, 1, 5]`
- `count = liste.count(5)`
- `print(f"Le nombre 5 apparaît {count} fois")`

- **Exercice 47**
- Triez la liste [34, 12, 89, 5, 23] en ordre croissant

- **Solution de l'exercice 47**

- `nombres = [34, 12, 89, 5, 23]`

- `nombres.sort()`

- `print(nombres)`

- **Exercice 48**
- Créez une liste carres contenant les carrés des nombres de 1 à 5

- **Solution de l'exercice 48**
- `carres = [i**2 for i in range(1, 6)]`
- `print(carres)`

- **Exercice 49**
- Doublez chaque élément de la liste [1, 2, 3, 4, 5] pour obtenir [2, 4, 6, 8, 10]

- **Solution de l'exercice 49**

- `liste = [1, 2, 3, 4, 5]`
- `liste_doublee = [x * 2 for x in liste]`
- `print(liste_doublee)`

- **Exercice 50**

- Fusionnez deux listes [1, 2, 3] et ['a', 'b', 'c'] en [(1, 'a'), (2, 'b'), (3, 'c')]

- **Solution de l'exercice 50**

- `liste1 = [1, 2, 3]`
- `liste2 = ['a', 'b', 'c']`
- `resultat = list(zip(liste1, liste2))`
- `print(resultat)`