

Memento Python

Variables : types, conversions

Chaînes de caractères	Listes
Concaténer deux chaînes "abc" et " def" : <code>"abc" + " def"</code>	Initialiser une liste L vide : <code>L = []</code>
Obtenir le caractère numéro i de la chaîne s : <code>s[i]</code> (comme pour une liste)	Ajouter l'élément x au bout de la liste L : <code>L.append(x)</code>
Convertir la chaîne de caractères "123" en un entier : <code>int("123")</code>	Ajouter l'élément x au rang i de la liste L : <code>L.insert(i,x)</code>
Convertir l'entier 123 en chaîne de caractères : <code>str(123)</code>	Supprimer la première occurrence de x dans la liste L : <code>L.remove(x)</code>
Convertir la liste L en chaîne de caractères, les éléments étant séparés par "-"; stocker le résultat dans chaîne : <code>chaîne = "-".join(L)</code>	Supprimer l'élément d'indice i dans la liste L : <code>L.pop(i)</code>
	Longueur d'une liste L : <code>len(L)</code>
	Trier une liste L et stocker le résultat dans L_triee : <code>L_triee = sorted(L)</code>

Calculs

4 opérations classiques : + - * / a exposant b : <code>a**b</code> Division entière de a par b : <code>a//b</code> Reste de la division euclidienne de a par b : <code>a%b</code> Définir z, nombre complexe : <code>z=complex(3,2)</code> En particulier, définir i : <code>i=complex(0,1)</code> Valeur absolue, module de z : <code>abs(z)</code> Partie réelle, partie imaginaire de z : <code>z.real</code> <code>z.imag</code> Argument de z (avec bibliothèque <code>cmath</code>) : <code>phase(z)</code>	Bibliothèque math Racine carrée de a : <code>sqrt(a)</code> Exponentielle de a : <code>exp(a)</code> Logarithme népérien de a : <code>log(a)</code> et <code>cos</code> , <code>sin</code> , <code>tan...</code> (angles en radians) Bibliothèque random Nombre aléatoire dans <code>[0;1[</code> : <code>random()</code> Nombre aléatoire entier entre a et b inclus : <code>randint(a,b)</code>
--	---

Tests

<code>if (condition):</code> <code>...</code> <code>else:</code> <code>...</code>	Opérateurs de comparaison : <code>==</code> <code>!=</code> <code><</code> <code>></code> <code><=</code> <code>>=</code> Opérateurs logiques : <code>and</code> <code>or</code>
--	---

Boucles

Boucle Pour	Boucle Tant que
<p>Avec exactement 10 itérations, le compteur variant de 0 à 9 :</p> <pre>for compteur in range(10): ...</pre> <p>Avec exactement 9 itérations, le compteur variant de 1 à 9 :</p> <pre>for compteur in range(1,10): ...</pre> <p>Parcours d'une liste non numérique :</p> <pre>for compteur in ["lundi", "mardi", "mercredi"]: ...</pre>	<pre>while (condition): ...</pre> <p>(les parenthèses autour de la condition sont optionnelles)</p>

Graphiques

En début de programme, charger la bibliothèque `matplotlib.pyplot` :

```
import matplotlib.pyplot as plt
```

Placer un point de coordonnées $(x;y)$, bleu, gros :		
<code>plt.plot(x,y,"bo")</code>	b	bleu
Tracer un segment entre deux points A et B , rouge :	g	vert
<code>plt.plot([x_A,x_B],[y_A,y_B],"r")</code>	r	rouge
Tracer les points dont les coordonnées sont données par les listes X_liste et Y_liste , en noir, en reliant les points :	c	cyan
<code>plt.plot(X_liste,Y_liste,"k-")</code>	m	magenta
Pour avoir un repère orthonormé :	y	jaune
<code>plt.axis("equal")</code>	k	noir
Pour une grille :	w	blanc
<code>plt.grid()</code>
Afficher la fenêtre graphique :		
<code>plt.show()</code>		