

Raccourcis Clavier

(Notebook)

Exécuter la cellule	Ctrl + Enter
Exécuter et passer à la cellule suivante	Shift + Enter
Auto-complétion (vscode)	Ctrl + Espace
Auto-complétion (jupyter)	TAB

Types Primaires

Chaîne de caractères	str	<code>s = '3.0'</code> ou <code>s = "False"</code>
Nombre Entier	int	<code>i = 3</code>
Nombre Flottant	float	<code>f = 3.0</code>
Boolean	bool	<code>b = True</code> ou <code>b = False</code>
Liste	list	<code>l = ['3.0', False, 3]</code>
Liste immuable	tuple	<code>t = ('3.0', False, 3)</code>
Dictionnaire	dict	<code>d = {'pi': 3.14, 'faux': False}</code>
Ensemble	set	<code>e = {1, 4, 5, 10}</code>
Objet nul		None
Lire le type d'une variable		type(variable)

Liste

Instancier une liste	<code>l = [1, '2', False]</code>
Ajouter un élément	<code>l.append(value)</code>
Insérer un élément	<code>l.insert(i, value)</code>
Supprimer un élément	<code>l.remove(value)</code>
Trier la liste	<code>l.sort()</code>
Concatener deux listes l1 et l2	<code>l1 + l2</code>

Chaîne de caractères

Instancier une chaîne	<code>s = "texte"</code>
Concaténer deux chaînes	<code>s + 'chaîne2'</code>
Test si <i>s</i> commence par 'te'	<code>s.startswith('te')</code>
Test si <i>s</i> fini par 'te'	<code>s.endswith('te')</code>
Remplacer 'tex' par 'chan'	<code>s.replace('tex', 'chan')</code>
Formater	<code>"%i / %s / %.2f" % (i, s, f)</code>

Dictionnaire

Instancier une dictionnaire	<code>d = {'key1': v1, "key1": v2}</code>
Modifier un élément	<code>d['key'] = value</code>
Itérer sur un dictionnaire	<code>for key, value in d.items():</code>

Opérations Générique

Déclarer une fonction	<code>def nom(arg1, arg2 = default):</code> → ... → <code>return valeur</code>
Boucle: For	<code>for i in range(n):</code>
Boucle: While	<code>while condition :</code>
Condition: Si	<code>if condition :</code>
Condition: Sinon si	<code>elif condition :</code>
Condition: Sinon	<code>else:</code>
Afficher dans la console.	<code>print("description:", valeur)</code>
Commentaire	<code># Commentaire</code>

Opérateurs

Égal / Différent	<code>==</code> / <code>!=</code>
Supérieur / Supérieur ou Égal	<code>></code> / <code>>=</code>
Inférieur / Inférieur ou Égal	<code><</code> / <code><=</code>
Inversion booléenne	<code>not</code>
ET booléen / OU booléen	<code>and</code> / <code>or</code>
Test si nul	<code>is None</code>
a^b	<code>a ** b</code>

Commun aux list, tuple, str...

Longueur	<code>len(l)</code>
Supprimer le <i>i</i> ^{ème} élément	<code>del l[i]</code>
Tester si l'objet contient une valeur	<code>value in l</code>
Itérer sur la liste	<code>for element in l:</code>
Appliquer la fonction <i>f</i> à toute la liste	<code>[f(v) for v in l]</code>
Filtrer la liste des objets nuls	<code>[v for v in l if v is not None]</code>

Indexes

Les indexes commencent à 0 !

Lecture / Écriture du <i>i</i> ^{ème} élément	<code>v = l[i] / l[i] = v</code>
Accéder au <i>i</i> ^{ème} dernier élément	<code>l[-i]</code>
Accéder du 2 ^{ème} au 4 ^{ème} éléments	<code>l[1:4]</code>
Accéder aux 3 premiers éléments	<code>l[:3]</code>
Accéder aux 3 derniers éléments	<code>l[-3:]</code>
Accéder à un élément sur deux	<code>l[::2]</code>
Inverser l'ordre des éléments	<code>l[::-1]</code>

Tableau (Matrices)

```
import numpy as np
```

Création de tableaux

Convertir une liste en tableau	<code>arr = np.array([0,1,2])</code>
Créer un tableau NxM de 0	<code>arr = np.zeros((N,M))</code>
Créer un tableau NxM de 1	<code>arr = np.ones((N,M))</code>
Créer une matrice identité NxN	<code>arr = np.eye(N)</code>
Vecteur: <code>[j, j+i, ..., k]</code>	<code>arr = np.arange(j, k, i)</code>
Intervalle de j à k avec N points	<code>arr = np.linspace(j, k, N)</code>
Charger depuis un fichier	<code>arr = np.load('path')</code>
Sauvegarder vers un fichier	<code>np.save('path', arr)</code>

Propriétés des tableaux

Type des données	<code>arr.dtype</code>
Nombre de dimensions	<code>arr.ndim</code>
Taille sur chaque dimension	<code>arr.shape</code>

Calculs sur les tableaux

Valeur minimale / maximale	<code>arr.min()</code> / <code>arr.max()</code>
Index de la valeur minimale	<code>arr.argmin()</code>
Index de la valeur maximale	<code>arr.argmax()</code>
Valeur moyenne	<code>arr.mean()</code>
Écart-type	<code>arr.std()</code>
Somme de tous les éléments	<code>arr.sum()</code>
Produit de tous les éléments	<code>arr.prod()</code>
Test si tout le tableau est vrai	<code>arr.all()</code>

Conversion de tableaux

Modifier le type de données	<code>arr.astype(np.dtype)</code>
Modifier les dimensions	<code>arr.reshape((d1, d2, ...))</code>
Permuter les axes	<code>arr.transpose(a1, a2, ...)</code>
Transformer en vecteur	<code>arr.flatten()</code>
Dupliquer le tableau	<code>arr.copy()</code>

Graphique

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

Créer la zone du tracé

Créer un graph	<code>fig, ax = plt.subplots()</code>
Créer plusieurs sous-graphs	<code>fig, axs = plt.subplots(ny, nx)</code>
Afficher le graph	<code>fig.show()</code>
Choisir la taille du graph	<code>fig.set_size_inches(width, height)</code>

Tracé le graphique

Tracer une courbe	<code>ax.plot(x, y, label="nom", color='red', ...)</code>
Tracer un nuage de point	<code>ax.scatter(x, y, label="nom", color='r', ...)</code>
Tracer un histogramme	<code>ax.bar(x, height, label="nom", color='r', ...)</code>
Trace un polygone	<code>ax.fill(x, y, color='r', ...)</code>
Tracer une ligne horizontale	<code>ax.hlines(y, xmin, xmax)</code>
Tracer une ligne verticale	<code>ax.vlines(x, ymin, ymax)</code>
Afficher une annotation	<code>ax.annotate("texte", (xpos, ypos))</code>
Affiche une image	<code>ax.imshow(array, cmap="gray")</code>

Configurer l'apparence du graphique

Spécifier un titre au graph	<code>ax.set_title("Titre")</code>
Spécifier un titre global	<code>fig.suptitle("Titre de la figure")</code>
Spécifier un titre à l'axe X	<code>ax.set_xlabel("Axe X")</code>
Spécifier un titre à l'axe Y	<code>ax.set_ylabel("Axe Y")</code>
Désactiver les axes	<code>ax.axis("off")</code>
Afficher une grille	<code>ax.grid(color='r', linewidth=2)</code>
Affiche la légende	<code>ax.legend()</code>

Manipulation d'images

Charger une image	<code>img = plt.imread("fichier.jpg")</code>
Accès à une région	<code>img[y_min : y_max, x_min : x_max]</code>
Accès à un canal	<code>img[:, :, canal]</code>