

# Mémento Python

## Commentaire

<code># ceci est un commentaire</code>	Un commentaire n'est pas pris en compte
--	---

## Affectation d'une variable

<code>a = 10</code>	L'entier 10 est stocké dans la variable a
<code>t = "la distance est "</code>	la phrase est stockée dans la variable t

## Passage d'un type de donnée à un autre

<code>a = int(" 56 ")</code>	La phrase est transformée en un entier qui est stocké dans a
<code>b = float("2.4")</code>	b stocke la valeur décimale 2,4
<code>t = str(15)</code>	L'entier 15 est transformé en phrase et stocké dans t

## Entrée et sortie de données

<code>print("var")</code>	Affiche la phrase « var »
<code>print(var)</code>	Affiche le contenu de la variable « var »
<code>adr = input("Votre adresse ? ")</code>	Affiche <b>Votre adresse ?</b> et attend que l'utilisateur entre un texte. La réponse est stockée dans la variable « var »

## Structures conditionnelles

<code>if temperature &lt; 0 :     print("Habillez-vous chaudement")</code>	if (condition vraie) alors : instructions à réaliser
<code>if moyenne &gt;= 10 :     print("Vous avez la moyenne") else :     print("Vous n'avez pas la moyenne")</code>	if (condition vraie) alors : instructions à réaliser Sinon : autres instructions à réaliser
<code>a = 2 if a == 2 :     print("a vaut ", a) a = a + 1</code>	Pour tester si a contient 2, on écrit <code>a == 2</code> et non <code>a = 2</code> . Affiche <code>a vaut 2</code> car le test est vrai. La dernière ligne ne dépend pas du if et sera exécutée quelque soit la valeur de a.

## Structures itératives

<pre>for i in range(3,6) :     print(i)</pre>	Pour <b>i</b> allant de 3 inclu à 6 exclu : afficher la valeur de <b>i</b>
<pre>while age &lt; 10 :     print(age)     age = age + 1</pre>	Tant que <b>age</b> est strictement inférieur à 10 : afficher <b>age</b> ajouter 1 à <b>age</b>

## Expressions mathématiques

<pre>b = a ** 2</pre>	$b = a^2$
<pre>from math import sqrt b = sqrt(a)</pre>	Importe la fonction <b>sqrt</b> du module <b>math</b> $b = \sqrt{a}$ racine carrée ↔ <b>square root</b>
<pre>import random b = random.randint(1,10)</pre>	Importe <b>randint</b> du module <b>random</b> . Tire au hasard un entier entre 1 et 10 et affecte le résultat à la variable <b>b</b> .

## Tracé d'une trajectoire avec matplotlib

<pre>import matplotlib.pyplot as plt plt.plot(X, Y, 'ro')</pre>	<b>X</b> et <b>Y</b> sont contiennent respectivement la liste des coordonnées $x$ et $y$ des points de la trajec- toire.
---	--

## Tracé de vecteurs avec matplotlib

<pre>plt.arrow(X,Y,Vx,Vy,head_width=0.2,           head_length=0.5,color="red")</pre>	<b>X</b> et <b>Y</b> sont définis ci-dessus. <b>Vx</b> et <b>Vy</b> sont des listes contenant les coordonnés $x$ et $y$ du vecteur vitesse. Trace le vecteur de coordonnées $(v_x, v_y)$ à partir du point de coordonnées $(x, y)$ .
---	--

## Légende et axes avec matplotlib

<pre>plt.axis('equal') ou si bug, plt.gca().set_aspect('equal')</pre>	Permet d'avoir la même unité en $x$ et en $y$
<pre>plt.xlim(xmin, xmax) plt.ylim(ymin, ymax)</pre>	Fixe les limites souhaitées en $x$ et en $y$
<pre>plt.text(x,y,"texte", fontsize=12)</pre>	Ecrit <b>texte</b> au point de coordonnées $(x, y)$
<pre>plt.xlabel("x (unité)") plt.ylabel("y (unité)")</pre>	Pour écrire les grandeurs sur les axes
<pre>plt.show()</pre>	Pour afficher la figure. A mettre à la fin du programme.