

# Interface avec Python

## 1. Élément python

Les éléments `python` permettent d'inclure un programme en Python dans la page et de récupérer les sorties (textes et figures SVG). L'attribut `type` précise l'action à effectuer

### 1.a. Attribut `type='code'`

C'est la valeur par défaut. L'élément `python` doit contenir le texte du code python à exécuter. Le code de tous les éléments `python` de la page est exécuté en une seule fois. L'attribut `visible` (valeur par défaut `"true"`) précise si le code doit apparaître dans la page.

La balise ouvrante de `python` doit être suivie de la première ligne de Python sans espace ni saut de ligne. De même, la balise fermante doit suivre immédiatement la dernière ligne.

Exemple : le fragment XML

```
<python type='code'>import math
a=[1,2,3,4]
b=a
for k in range(4):
    b[k] = math.cos(a[k])</python>
```

exécute le code correspondant et donne dans la page :

```
import math
a=[1,2,3,4]
b=a
for k in range(4):
    b[k] = math.cos(a[k])
```

### 1.b. Attribut `type='print'`

L'élément `python` doit contenir une expression Python qui est convertie en chaîne de caractères et affichée dans la page.

Exemple :

```
<python type="print">b</python>
```

```
print(b)
--> [0.5403023058681398, -0.4161468365471424, -0.9899924966004454, -0.653643620863611]
```

La fonction utilisée pour la conversion est `repr`.

### 1.c. Attribut `type='plot'`

Dans ce cas, l'élément `python` permet d'afficher dans la page une figure obtenue avec le module python [Mathplotlib](#).

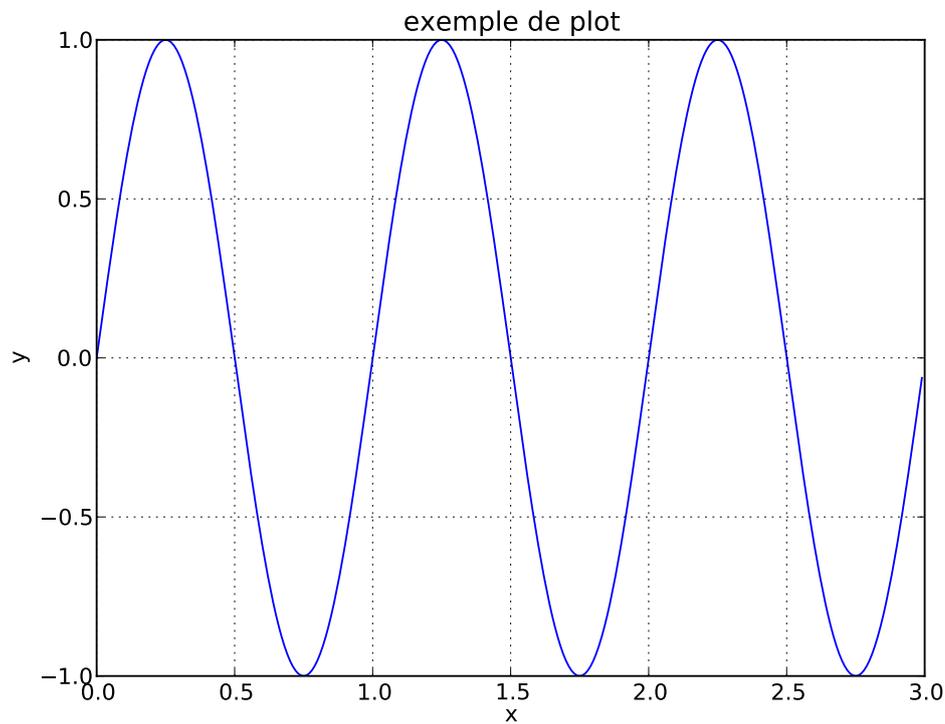
La figure doit être générée dans un élément `python` antérieur. L'élément d'attribut `type='plot'` a pour fonction d'effectuer l'exportation avec la fonction `savefig` du module `pylab`.

Exemple : le fragment XML

```
<python>from pylab import *
from math import sin
from numpy import *
xlabel('x')
ylabel('y')
title('exemple de plot')
grid(True)
t = arange(0.0,3.0,0.01)
s=sin(2*pi*t)
plot(t,s)</python>
<python type="plot" name="plot1.svg" width="600" height="400"/>
```

génère le résultat

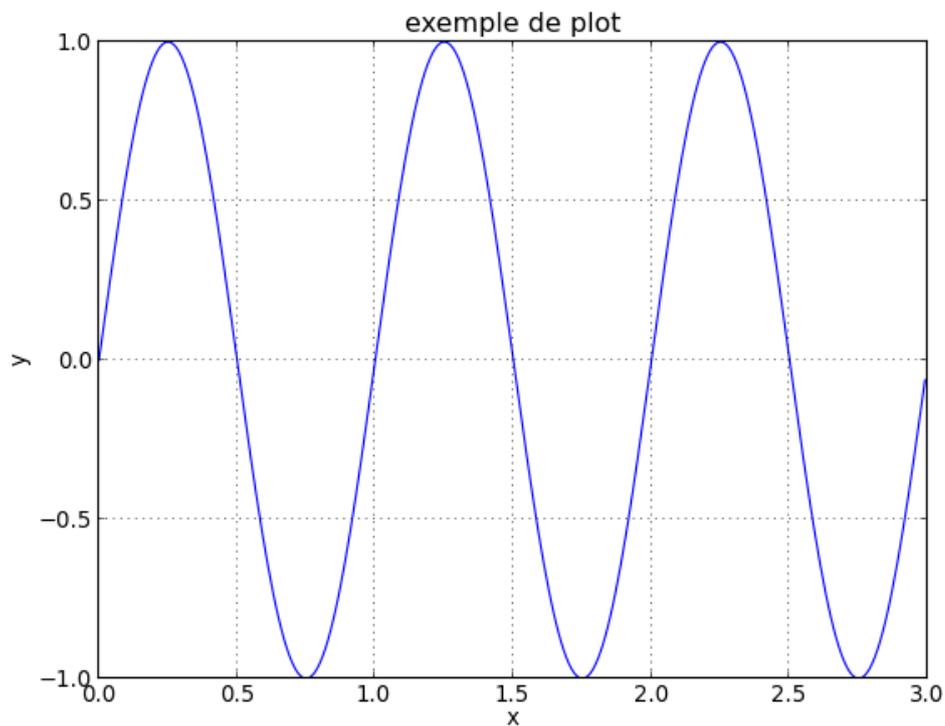
```
from pylab import *
from math import sin
from numpy import *
xlabel('x')
ylabel('y')
title('exemple de plot')
grid(True)
t = arange(0.0,3.0,0.01)
s=sin(2*pi*t)
plot(t,s)
```



Dans ce cas, l'extension ".svg" du nom indique que la figure incluse dans la page XML est en SVG. C'est l'option recommandée si l'on souhaite pouvoir afficher la figure en pleine page. La figure PDF utilisée par LaTeX pour la page PDF est générée également par la fonction `savefig`.

Si l'on préfère afficher dans la page XML une image PNG, il suffit de donner un nom avec l'extension correspondante. C'est l'option recommandée si la figure contient beaucoup d'éléments ou des images. Dans ce cas, un lien vers une version PDF de la figure apparaît à côté de l'image.

```
<python type='plot' name="plot2.png" width="600" height="400"/>
```



#### 1.d. Attribut `type='xml'`

L'élément `python` contient une chaîne de caractères `python`, représentant un fragment XML qui est inséré à l'emplacement correspondant.

Exemple : le fragment suivant (auquel il faut ajouter un `CDATA` pour encadrer la chaîne)

```
<python type="xml">"(<p><sb>%f</sb></p>"%(math.pi))</python>
```

produit le paragraphe suivant :

3.141593

La principale application de cette option est la génération de courbes et autres figures générées par le calcul. Voir la page sur le [tracé de courbes avec le module Draw](#).