

Initiation à la programmation, Leçon 6 : numpy et pyplot

1. Sélectionner des entrées

Une autre caractéristique intéressante des arrays est la capacité de sélectionner un sous-ensemble des entrées. Par exemple, si vous voulez savoir quels poids dans notre array `weights_kg` sont au-dessus de 90, nous pourrions rapidement le faire de la façon suivante.

```
import numpy as np
weight_kg = np.array([81.65, 95.25, 92.98, 86.18, 97.52, 88.45])
over_90 = weight_kg > 90
print over_90, weight_kg[over_90]
```

2. Pyramide des âges: Pourcentage d'hommes et de femmes par région

Trouvez la région la plus masculine et la plus féminine.

3. Produit scalaire

Imaginez que vous dirigez un magasin de bagel. Vous avez plusieurs recettes de bagels, et pour chacun vous stockez les ingrédients nécessaires dans une table (en kg). Vous connaissez également le prix de chaque ingrédient: le fromage coûte 15 € / kg, le jambon 40 € / kg et la tomate 5 € / kg. Votre tâche consiste à calculer le prix de chaque bagel.

	Fromage	Jambon	Tomate
Bagel 1	0.1	0.1	0.05
Bagel 2	0.05	0.1	0.1
Bagel 3	0.2	0	0.1

Le prix de Bagel 1 est $15 \cdot 0.1 + 40 \cdot 0.1 + 5 \cdot 0.05 = 5.75$. Le prix de Bagel 2 est $15 \cdot 0.05 + 40 \cdot 0.1 + 5 \cdot 0.1 = 5.25$. Le prix de Bagel 3 est $15 \cdot 0.2 + 40 \cdot 0 + 5 \cdot 0.1 = 3.5$. Nous pouvons écrire les prix comme un array `[5.75, 5.25, 3.5]`. On dit que cet array est le résultat du produit scalaire du array des ingrédients et du array des prix.

```
import numpy as np
ingredients = np.array([[0.1, 0.1, 0.05], [0.05, 0.1, 0.1],
                        [0.2, 0, 0.1]])
prices_ingredients = np.array([15, 40, 5])
prices_bagels = np.dot(ingredients, prices_ingredients)
```

4. Pyramide des âges: Âge moyen par région

En supposant que l'âge de chaque personne dans le groupe 0-19 est de 10, dans le groupe 20-39 est de 30, dans le groupe 40-59 est de 50, dans le groupe 60-74 est de 67 et dans le groupe 70+ est de 80, calculer la moyenne âge dans chaque région. Trouvez les régions ayant les âges moyens les plus élevés et les plus petits.

Pyplot

5. Pyramide des âges: camembert des groupes d'âge féminin et masculin en Alsace

Voici un exemple de camembert dessiné à l'aide de `matplotlib`.

```
import matplotlib.pyplot as plt
import numpy as np

plt.figure(1, figsize=(6,6))
labels = 'Frogs', 'Hogs', 'Dogs', 'Logs'
fracs = np.array([15, 30, 45, 10])
plt.pie(fracs, labels=labels, autopct='%1.1f%%', shadow=True)
plt.title('Raining Hogs and Dogs')
plt.savefig('raining.png')
```

Initiation à la programmation, Leçon 6 : numpy et pyplot

```
plt.show()
```

Votre tâche est de dessiner un camembert des groupes d'âge féminins et masculins en Alsace.

6. Pyramide des âges: Diagramme à barres des populations masculines et féminines selon l'âge

Voici un exemple de diagramme à barres pour l'utilisation du langage de programmation.

```
import matplotlib.pyplot as plt
import numpy as np

languages = np.array(['Python', 'C++', 'Java', 'Perl', 'Scala',
                     'Lisp'])
number_of_users = np.array([10,8,6,4,2,1])

plt.title('Programming language usage')
x_pos = np.arange(len(languages))
bc = plt.bar(x_pos-0.4, number_of_users, color='b')
plt.xticks(x_pos, languages)
plt.ylabel('Usage')

plt.savefig('programming_languages.png')
plt.show()
```

Votre tâche consiste à dessiner un diagramme à barres des populations masculines et féminines selon l'âge. Pour ajouter une légende à votre graphique, utilisez `plt.legend((bc_men[0], bc_women[0]), ('Men', 'Women'))`.

7. Tracés de fonctions "scientifiques"

Dessiner un tracé d'une fonction qui est égal à $-x-5$ sur $[-10,-5]$, à x^2-25 sur $[-5,4]$, et à $\sin(x-4)-9$ sur $[4,10]$.

```
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(-5,5,101) # 101 equally spaces ticks from -5 to 5
y = np.sin(x)
plt.plot(x,y)
plt.show()
```

8. (*) 100 exercices de numpy

Suivez [ce lien](http://www.labri.fr/perso/nrougier/teaching/numpy.100/) : <http://www.labri.fr/perso/nrougier/teaching/numpy.100/>. Vous pouvez faire tous les exercices que vous trouvez intéressant, par exemple, 38, 50, 51. Les zones gris contiennent des solutions, essayez de trouver une solution vous-même avant de les lire.

9. (*) L'ensemble de Mandelbrot

L'ensemble de Mandelbrot est une fractale qui est définie comme l'ensemble des points c du plan complexe pour lesquels la suite récurrente définie par $z_{n+1} = z_n^2 + c$ et la condition $z_0 = 0$ ne tend pas vers l'infini (en module). Si nous reformulons cela sans utiliser les nombres complexes, en remplaçant z_n par le couple (x_n, y_n) et c par le couple (a, b) alors nous obtenons: $x_{n+1} = x_n^2 - y_n^2 + a$ et $y_{n+1} = 2x_n y_n + b$.

Il peut être démontré que dès que le module de z_n est strictement plus grand que 2 (z_n étant sous forme algébrique, quand $x_n^2 + y_n^2 > 2$), la suite diverge vers l'infini, et donc c est en dehors de l'ensemble de Mandelbrot. Cela nous permet d'arrêter le calcul pour les points ayant un module

Initiation à la programmation, Leçon 6 : numpy et pyplot

strictement supérieur à 2 et qui sont donc en dehors de l'ensemble de Mandelbrot. Pour les points de l'ensemble de Mandelbrot, i.e. les nombres complexes c pour lesquels z_n ne tend pas vers l'infini, le calcul n'arrivera jamais à terme, donc il doit être arrêté après un certain nombre d'itérations déterminé par le programme.

Écrire un script qui affiche (une approximation de) l'ensemble de Mandelbrot.