

# Initiation à la programmation, Leçon 7

---

## 1. Camembert de la météo

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

plt.figure(1, figsize=(6,6))
labels = 'Rain', 'Sunny', 'Cloudy', 'Partly cloudy'
fracs = np.array([15,30,45, 10])
plt.pie(fracs, labels=labels, autopct='%1.1f%%',shadow=True)
plt.title("Weather forecasts")
plt.savefig("forecasts.png")
plt.show()
```

## 2. (\*) Pyramide des âges : camembert des groupes d'âge

Dessiner un camembert des groupes d'âge féminins et un camembert des groupes d'âge masculins en Alsace.

## 3. 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()
```

## 4. (\*) Pyramide des âges: Diagramme à barres

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

## 5. Tracés des fonctions "scientifiques"

```
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()
```

## 6. (\*) Tracés des 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  $[4,10]$ .

## 7. HyperText Markup Language

Allez sur <http://dataquestio.github.io/web-scraping-pages/simple.html>. Utilisez Developer Tools (View → Developer → View source code) dans votre navigateur

## Initiation à la programmation, Leçon 7

---

web pour voir le code du page. HTML comporte des éléments qui s'appellent « tag » (<html>, <head <title>, <body>, etc).

### 8. Télécharger un document HTML

```
import requests
page = requests.get("http://dataquestio.github.io/web-scraping-
pages/simple.html")
page.content
```

### 9. Trouver tous les instances de tag

```
import requests
from bs4 import BeautifulSoup
page = requests.get("http://dataquestio.github.io/web-scraping-
pages/simple.html")
soup = BeautifulSoup(page.content, 'html.parser')
for tag in soup.find_all('p'):
    print tag.get_text()
```

### 10. Classes et ids

Les propriétés `class` et `id` donnent des noms à des éléments de HTML, et les font plus facile à travailler avec. Un élément peut avoir plusieurs classes, et un classe peut être partagé par plusieurs éléments. Chaque élément peut avoir juste un id, et un id peut être utilisé une fois sur une page. Classes et ids ne sont pas obligatoires, et pas tous les éléments en ont.

```
<html>
  <head>
  </head>
  <body>
    <p class="bold-paragraph">
      Here's a paragraph of text!
      <a href="https://www.dataquest.io" id="learn-link">Learn Data Science Online</a>
    </p>
    <p class="bold-paragraph extra-large">
      Here's a second paragraph of text!
      <a href="https://www.python.org" class="extra-large">Python</a>
    </p>
  </body>
</html>
```

### 11. Trouver des tags par classes et ids

```
import requests
from bs4 import BeautifulSoup
page = requests.get("http://dataquestio.github.io/web-scraping-pages/ids_and_classes.html")
soup = BeautifulSoup(page.content, 'html.parser')

#We can find all p tags that have class 'outer-text'
print "Contents of all tags p with class 'outer-text'"
for tag in soup.find_all('p', class_='outer-text'):
    print tag.get_text()

#We can find all tags that have class 'outer-text'
print "Contents of all tags with class 'outer-text'"
for tag in soup.find_all(class_='outer-text'):
    print tag.get_text()

#We can find all tags that have id 'first'
```

## Initiation à la programmation, Leçon 7

---

```
print "Contents of all tags with id 'first'"
for tag in soup.find_all(id='first'):
    print tag.get_text()
```

### 12. Utilisez des sélecteurs CSS

```
import requests
from bs4 import BeautifulSoup
page = requests.get("http://dataquestio.github.io/web-scraping-pages/ids_and_classes.html")
soup = BeautifulSoup(page.content, 'html.parser')

#We can find all p tags that have class 'outer-text'
print "Contents of all tags p that are inside of a div tag"
for tag in soup.select('div p'):
    print tag.get_text()
```

Vous pouvez également utiliser `p` a pour trouver tous les tags 'a' dedans un tag 'p', `body p` a pour trouver tous les tags 'a' dedans un tag 'p' dedans un tag 'body', `html body` pour trouver tous les body tags dedans un html tag, `p.outer-text` pour trouver tous les p tags avec un classe `outer-text`; `p#first` pour trouver tous les p tags avec un id `first`; `body p.outer-text` pour trouver tous les p tags avec un classe `outer-text` dedans un body tag, etc.

### 13. (\*) Cours de ENS

Allez sur [moodle.di.ens.fr](http://moodle.di.ens.fr). Utilisez BeautifulSoup pour trouver les noms de tous les cours qui utilisent Moodle.