Introduction to programming, Lesson 7

1. **HyperText Markup Language**

   Go to a webpage http://dataquestio.githb.io/web-scraping-pages/simple.html. Use Developer Tools (View → Developer → View source code) in your browser to see the webpage's code. HTML consists of elements called tags (<html>, <head <title>, <body>, etc).

2. **Download an HTML document**

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

3. **Finding all instances of a tag at once**

```
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()
```

4. **Classes and ids**

   The class and id properties give HTML elements names, and make them easier to interact with when we're scraping. One element can have multiple classes, and a class can be shared between elements. Each element can only have one id, and an id can only be used once on a page. Classes and ids are optional, and not all elements will have them.

```
<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>
```

5. **Searching for tags by class and id**

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

6. Using CSS selectors

```
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')

print "Contents of all tags p that are inside of a div tag"
for tag in soup.select('div p'):
    print tag.get_text()
```

You can also use `p a` — finds all `a` tags inside of a `p` tag; `body p a` — finds all `a` tags inside of a `p` tag inside of a `body` tag; `html body` — finds all `body` tags inside of an `html` tag; `p.outer-text` — finds all `p` tags with a class of `outer-text`; `p#first` — finds all `p` tags with an id of `first`; `body p.outer-text` — finds any `p` tags with a class of `outer-text` inside of a `body` tag, etc.

7. **Weather forecast.** Go to weather.com. Choose your favorite city and open a page that contains the 5-day forecast. Use requests and bs4 to download and parse the forecasts. Write another function that will receive the date and will return the weather forecast for this day.
8. **World population.** Go to http://www.worldometers.info/world-population/population-by-country/. Use requests and bs4 to download and parse the population for every country. What is the country where the yearly change is the largest? The smallest? What is the country with largest / smallest density of population? Draw a barplot for median ages.