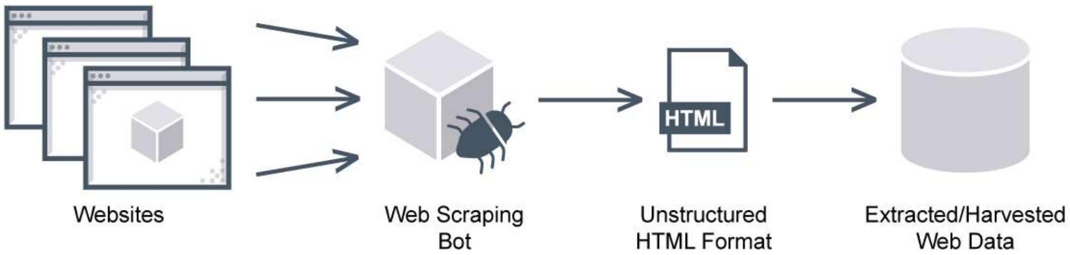
Web Scraping Data



In [1]:
```python
import requests
from bs4 import BeautifulSoup
url = 'https://realpython.github.io/fake-jobs/'
page = requests.get(url)
soup = BeautifulSoup(page.content,"html.parser")
print(soup.prettify())
```

```
     <div class="column is-half">
      <div class="card">
       <div class="card-content">
        <div class="media">
         <div class="media-left">
          <figure class="image is-48x48">
           <img alt="Real Python Logo" src="https://files.realpython.co
m/media/real-python-logo-thumbnail.7f0db70c2ed2.jpg?__no_cf_polish=1"/>
          </figure>
         </div>
         <div class="media-content">
          <h2 class="title is-5">
           Senior Python Developer
          </h2>
          <h3 class="subtitle is-6 company">
           Payne, Roberts and Davis
          </h3>
         </div>
        </div>
        <div class="content">
```

In [2]:

```
result =soup.find(id='ResultsContainer')
print(result.prettify())
```

```
<div class="columns is-multiline" id="ResultsContainer">
 <div class="column is-half">
  <div class="card">
   <div class="card-content">
    <div class="media">
     <div class="media-left">
      <figure class="image is-48x48">
       <img alt="Real Python Logo" src="https://files.realpython.com/med
ia/real-python-logo-thumbnail.7f0db70c2ed2.jpg?__no_cf_polish=1"/>
      </figure>
     </div>
     <div class="media-content">
      <h2 class="title is-5">
       Senior Python Developer
      </h2>
      <h3 class="subtitle is-6 company">
       Payne, Roberts and Davis
      </h3>
     </div>
```

In [3]:

```
job = result.find_all("div",class_="card-content")
print(job)
```

```
[<div class="card-content">
<div class="media">
<div class="media-left">
<figure class="image is-48x48">
<img alt="Real Python Logo" src="https://files.realpython.com/media/real
-python-logo-thumbnail.7f0db70c2ed2.jpg?__no_cf_polish=1"/>
</figure>
</div>
<div class="media-content">
<h2 class="title is-5">Senior Python Developer</h2>
<h3 class="subtitle is-6 company">Payne, Roberts and Davis</h3>
</div>
</div>
<div class="content">
<p class="location">
        Stewartbury, AA
      </p>
<p class="is-small has-text-grey">
<time datetime="2021-04-08">2021-04-08</time>
```

In [4]:

```
for jobs in job:
    title =jobs.find("h2",class_="title")
    print(title.text.strip())
```

```
Senior Python Developer
Energy engineer
Legal executive
Fitness centre manager
Product manager
Medical technical officer
Physiological scientist
Textile designer
Television floor manager
Waste management officer
Software Engineer (Python)
Interpreter
Architect
Meteorologist
Audiological scientist
English as a second language teacher
Surgeon
Equities trader
Newspaper journalist
Materials engineer
Python Programmer (Entry-Level)
Product/process development scientist
Scientist, research (maths)
Ecologist
Materials engineer
Historic buildings inspector/conservation officer
Data scientist
Psychiatrist
Structural engineer
Immigration officer
Python Programmer (Entry-Level)
Neurosurgeon
Broadcast engineer
Make
Nurse, adult
Air broker
Editor, film/video
Production assistant, radio
Engineer, communications
Sales executive
Software Developer (Python)
Futures trader
Tour manager
Cytogeneticist
Designer, multimedia
Trade union research officer
Chemist, analytical
Programmer, multimedia
Engineer, broadcasting (operations)
Teacher, primary school
Python Developer
Manufacturing systems engineer
Producer, television/film/video
Scientist, forensic
Bonds trader
Editorial assistant
Photographer
Retail banker
Jewellery designer
Ophthalmologist
Back-End Web Developer (Python, Django)
```

```
Licensed conveyancer
Futures trader
Counselling psychologist
Insurance underwriter
Engineer, automotive
Producer, radio
Dispensing optician
Designer, fashion/clothing
Chartered loss adjuster
Back-End Web Developer (Python, Django)
Forest/woodland manager
Clinical cytogeneticist
Print production planner
Systems developer
Graphic designer
Writer
Field seismologist
Chief Strategy Officer
Air cabin crew
Python Programmer (Entry-Level)
Warden/ranger
Sports therapist
Arts development officer
Printmaker
Health and safety adviser
Manufacturing systems engineer
Programmer, applications
Medical physicist
Media planner
Software Developer (Python)
Surveyor, land/geomatics
Legal executive
Librarian, academic
Barrister
Museum/gallery exhibitions officer
Radiographer, diagnostic
Database administrator
Furniture designer
Ship broker
```

In [5]:
```
job_title=[]
for jobs in job:
    title = jobs.find("h2",class_="title")
    data =title.text.strip()
    job_title.append(data)
print(job_title)
```

```
['Senior Python Developer', 'Energy engineer', 'Legal executive', 'Fitness
centre manager', 'Product manager', 'Medical technical officer', 'Physiolo
gical scientist', 'Textile designer', 'Television floor manager', 'Waste m
anagement officer', 'Software Engineer (Python)', 'Interpreter', 'Architec
t', 'Meteorologist', 'Audiological scientist', 'English as a second langua
ge teacher', 'Surgeon', 'Equities trader', 'Newspaper journalist', 'Materi
als engineer', 'Python Programmer (Entry-Level)', 'Product/process develop
ment scientist', 'Scientist, research (maths)', 'Ecologist', 'Materials en
gineer', 'Historic buildings inspector/conservation officer', 'Data scient
ist', 'Psychiatrist', 'Structural engineer', 'Immigration officer', 'Pytho
n Programmer (Entry-Level)', 'Neurosurgeon', 'Broadcast engineer', 'Make',
'Nurse, adult', 'Air broker', 'Editor, film/video', 'Production assistant,
radio', 'Engineer, communications', 'Sales executive', 'Software Developer
(Python)', 'Futures trader', 'Tour manager', 'Cytogeneticist', 'Designer,
multimedia', 'Trade union research officer', 'Chemist, analytical', 'Progr
ammer, multimedia', 'Engineer, broadcasting (operations)', 'Teacher, prima
ry school', 'Python Developer', 'Manufacturing systems engineer', 'Produce
r, television/film/video', 'Scientist, forensic', 'Bonds trader', 'Editori
al assistant', 'Photographer', 'Retail banker', 'Jewellery designer', 'Oph
thalmologist', 'Back-End Web Developer (Python, Django)', 'Licensed convey
ancer', 'Futures trader', 'Counselling psychologist', 'Insurance underwrit
er', 'Engineer, automotive', 'Producer, radio', 'Dispensing optician', 'De
signer, fashion/clothing', 'Chartered loss adjuster', 'Back-End Web Develo
per (Python, Django)', 'Forest/woodland manager', 'Clinical cytogeneticis
t', 'Print production planner', 'Systems developer', 'Graphic designer',
'Writer', 'Field seismologist', 'Chief Strategy Officer', 'Air cabin cre
w', 'Python Programmer (Entry-Level)', 'Warden/ranger', 'Sports therapis
t', 'Arts development officer', 'Printmaker', 'Health and safety adviser',
'Manufacturing systems engineer', 'Programmer, applications', 'Medical phy
sicist', 'Media planner', 'Software Developer (Python)', 'Surveyor, land/g
eomatics', 'Legal executive', 'Librarian, academic', 'Barrister', 'Museum/
gallery exhibitions officer', 'Radiographer, diagnostic', 'Database admini
strator', 'Furniture designer', 'Ship broker']
```

In [6]:

```
import pandas as pd
Job_Titles =  pd.DataFrame(job_title,columns=["Job Name"])
Job_Titles
```

Out[6]:

```
                              Job Name
 0              Senior Python Developer
 1                      Energy engineer
 2                      Legal executive
 3              Fitness centre manager
 4                      Product manager
...                                 ...
95  Museum/gallery exhibitions officer
96           Radiographer, diagnostic
97             Database administrator
98                  Furniture designer
99                         Ship broker

100 rows × 1 columns
```

In [7]:

```
for jobs in job:
    company =jobs.find("h3",class_="company")
    print(company.text.strip())
```

```
Payne, Roberts and Davis
Vasquez-Davidson
Jackson, Chambers and Levy
Savage-Bradley
Ramirez Inc
Rogers-Yates
Kramer-Klein
Meyers-Johnson
Hughes-Williams
Jones, Williams and Villa
Garcia PLC
Gregory and Sons
Clark, Garcia and Sosa
Bush PLC
Salazar-Meyers
Parker, Murphy and Brooks
Cruz-Brown
Macdonald-Ferguson
Williams, Peterson and Rojas
Smith and Sons
Moss, Duncan and Allen
Gomez-Carroll
Manning, Welch and Herring
Lee, Gutierrez and Brown
Davis, Serrano and Cook
Smith LLC
Thomas Group
Silva-King
Pierce-Long
Walker-Simpson
Cooper and Sons
Donovan, Gonzalez and Figueroa
Morgan, Butler and Bennett
Snyder-Lee
Harris PLC
Washington PLC
Brown, Price and Campbell
Mcgee PLC
Dixon Inc
Thompson, Sheppard and Ward
Adams-Brewer
Schneider-Brady
Gonzales-Frank
Smith-Wong
Pierce-Herrera
Aguilar, Rivera and Quinn
Lowe, Barnes and Thomas
Lewis, Gonzalez and Vasquez
Taylor PLC
Oliver, Jones and Ramirez
Rivera and Sons
Garcia PLC
Johnson, Wells and Kramer
Gonzalez LLC
Morgan, White and Macdonald
Robinson-Fitzpatrick
Waters, Wilson and Hoover
Hill LLC
Li-Gregory
Fisher, Ryan and Coleman
Stewart-Alexander
```

```
Abbott and Sons
Bryant, Santana and Davenport
Smith PLC
Patterson-Singh
Martinez-Berry
May, Taylor and Fisher
Bailey, Owen and Thompson
Vasquez Ltd
Leblanc LLC
Jackson, Ali and Mckee
Blankenship, Knight and Powell
Patton, Haynes and Jones
Wood Inc
Collins Group
Flores-Nelson
Mitchell, Jones and Olson
Howard Group
Kramer-Edwards
Berry-Houston
Mathews Inc
Riley-Johnson
Spencer and Sons
Camacho-Sanchez
Oliver and Sons
Eaton PLC
Stanley-Frederick
Bradley LLC
Parker, Goodwin and Zavala
Kim-Miles
Moreno-Rodriguez
Brown-Ortiz
Hartman PLC
Brooks Inc
Washington-Castillo
Nguyen, Yoder and Petty
Holder LLC
Yates-Ferguson
Ortega-Lawrence
Fuentes, Walls and Castro
```

In [8]:

```
Company_Title=[]
for jobs in job:
    Company=jobs.find("h3",class_="company")
    dataa = Company.text.strip()
    Company_Title.append(dataa)
print(Company_Title)
```

```
['Payne, Roberts and Davis', 'Vasquez-Davidson', 'Jackson, Chambers and Le
vy', 'Savage-Bradley', 'Ramirez Inc', 'Rogers-Yates', 'Kramer-Klein', 'Mey
ers-Johnson', 'Hughes-Williams', 'Jones, Williams and Villa', 'Garcia PL
C', 'Gregory and Sons', 'Clark, Garcia and Sosa', 'Bush PLC', 'Salazar-Mey
ers', 'Parker, Murphy and Brooks', 'Cruz-Brown', 'Macdonald-Ferguson', 'Wi
lliams, Peterson and Rojas', 'Smith and Sons', 'Moss, Duncan and Allen',
'Gomez-Carroll', 'Manning, Welch and Herring', 'Lee, Gutierrez and Brown',
'Davis, Serrano and Cook', 'Smith LLC', 'Thomas Group', 'Silva-King', 'Pie
rce-Long', 'Walker-Simpson', 'Cooper and Sons', 'Donovan, Gonzalez and Fig
ueroa', 'Morgan, Butler and Bennett', 'Snyder-Lee', 'Harris PLC', 'Washing
ton PLC', 'Brown, Price and Campbell', 'Mcgee PLC', 'Dixon Inc', 'Thompso
n, Sheppard and Ward', 'Adams-Brewer', 'Schneider-Brady', 'Gonzales-Fran
k', 'Smith-Wong', 'Pierce-Herrera', 'Aguilar, Rivera and Quinn', 'Lowe, Ba
rnes and Thomas', 'Lewis, Gonzalez and Vasquez', 'Taylor PLC', 'Oliver, Jo
nes and Ramirez', 'Rivera and Sons', 'Garcia PLC', 'Johnson, Wells and Kra
mer', 'Gonzalez LLC', 'Morgan, White and Macdonald', 'Robinson-Fitzpatric
k', 'Waters, Wilson and Hoover', 'Hill LLC', 'Li-Gregory', 'Fisher, Ryan a
nd Coleman', 'Stewart-Alexander', 'Abbott and Sons', 'Bryant, Santana and
Davenport', 'Smith PLC', 'Patterson-Singh', 'Martinez-Berry', 'May, Taylor
and Fisher', 'Bailey, Owen and Thompson', 'Vasquez Ltd', 'Leblanc LLC', 'J
ackson, Ali and Mckee', 'Blankenship, Knight and Powell', 'Patton, Haynes
and Jones', 'Wood Inc', 'Collins Group', 'Flores-Nelson', 'Mitchell, Jones
and Olson', 'Howard Group', 'Kramer-Edwards', 'Berry-Houston', 'Mathews In
c', 'Riley-Johnson', 'Spencer and Sons', 'Camacho-Sanchez', 'Oliver and So
ns', 'Eaton PLC', 'Stanley-Frederick', 'Bradley LLC', 'Parker, Goodwin and
Zavala', 'Kim-Miles', 'Moreno-Rodriguez', 'Brown-Ortiz', 'Hartman PLC', 'B
rooks Inc', 'Washington-Castillo', 'Nguyen, Yoder and Petty', 'Holder LL
C', 'Yates-Ferguson', 'Ortega-Lawrence', 'Fuentes, Walls and Castro']
```

In [9]:

```
Company_Titles = pd.DataFrame(Company_Title,columns=["Company"])
Company_Titles
```

Out[9]:

	Company
0	Payne, Roberts and Davis
1	Vasquez-Davidson
2	Jackson, Chambers and Levy
3	Savage-Bradley
4	Ramirez Inc
...	...
95	Nguyen, Yoder and Petty
96	Holder LLC
97	Yates-Ferguson
98	Ortega-Lawrence
99	Fuentes, Walls and Castro

100 rows × 1 columns

In [10]:

```
for jobs in job:
    location =jobs.find("p",class_="location")
    print(location.text.strip())
```

```
Stewartbury, AA
Christopherville, AA
Port Ericaburgh, AA
East Seanview, AP
North Jamieview, AP
Davidville, AP
South Christopher, AE
Port Jonathan, AE
Osbornetown, AE
Scotttown, AP
Ericberg, AE
Ramireztown, AE
Figueroaview, AA
Kelseystad, AA
Williamsburgh, AE
Mitchellburgh, AE
West Jessicabury, AA
Maloneshire, AE
Johnsonton, AA
South Davidtown, AP
Port Sara, AE
Marktown, AA
Laurenland, AE
Lauraton, AP
South Tammyberg, AP
North Brandonville, AP
Port Robertfurt, AA
Burnettbury, AE
Herbertside, AA
Christopherport, AP
West Victor, AE
Port Aaron, AP
Loribury, AA
Angelastad, AP
Larrytown, AE
West Colin, AP
West Stephanie, AP
Laurentown, AP
Wrightberg, AP
Alberttown, AE
Brockburgh, AE
North Jason, AE
Arnoldhaven, AE
Lake Destiny, AP
South Timothyburgh, AP
New Jimmyton, AE
New Lucasbury, AP
Port Cory, AE
Gileston, AA
Cindyshire, AA
East Michaelfort, AA
Joybury, AE
Emmatown, AE
Colehaven, AP
Port Coryton, AE
Amyborough, AA
Reynoldsville, AA
Port Billy, AP
Adamburgh, AA
Wilsonmouth, AA
South Kimberly, AA
```

```
Benjaminland, AP
Zacharyport, AA
Port Devonville, AE
East Thomas, AE
New Jeffrey, AP
Davidside, AA
Jamesville, AA
New Kelly, AP
Lake Antonio, AA
New Elizabethside, AA
Millsbury, AE
Lloydton, AP
Port Jeremy, AA
New Elizabethtown, AA
Charlesstad, AE
Josephbury, AE
Seanfurt, AA
Williambury, AA
South Jorgeside, AP
Robertborough, AP
South Saratown, AP
Hullview, AA
Philipland, AP
North Patty, AE
North Stephen, AE
Stevensland, AP
Reyesstad, AE
Bellberg, AP
North Johnland, AE
Martinezburgh, AE
Joshuatown, AE
West Ericstad, AA
Tuckertown, AE
Perezton, AE
Lake Abigail, AE
Jacobshire, AP
Port Susan, AE
North Tiffany, AA
Michelleville, AP
```

In [11]:
```
Location_Title=[]
for jobs in job:
    location = jobs.find("p",class_="location")
    dataaa = location.text.strip()
    Location_Title.append(dataaa)
print(Location_Title)
```

```
['Stewartbury, AA', 'Christopherville, AA', 'Port Ericaburgh, AA', 'East S
eanview, AP', 'North Jamieview, AP', 'Davidville, AP', 'South Christopher,
AE', 'Port Jonathan, AE', 'Osbornetown, AE', 'Scotttown, AP', 'Ericberg, A
E', 'Ramireztown, AE', 'Figueroaview, AA', 'Kelseystad, AA', 'Williamsburg
h, AE', 'Mitchellburgh, AE', 'West Jessicabury, AA', 'Maloneshire, AE', 'J
ohnsonton, AA', 'South Davidtown, AP', 'Port Sara, AE', 'Marktown, AA', 'L
aurenland, AE', 'Lauraton, AP', 'South Tammyberg, AP', 'North Brandonvill
e, AP', 'Port Robertfurt, AA', 'Burnettbury, AE', 'Herbertside, AA', 'Chri
stopherport, AP', 'West Victor, AE', 'Port Aaron, AP', 'Loribury, AA', 'An
gelastad, AP', 'Larrytown, AE', 'West Colin, AP', 'West Stephanie, AP', 'L
aurentown, AP', 'Wrightberg, AP', 'Alberttown, AE', 'Brockburgh, AE', 'Nor
th Jason, AE', 'Arnoldhaven, AE', 'Lake Destiny, AP', 'South Timothyburgh,
AP', 'New Jimmyton, AE', 'New Lucasbury, AP', 'Port Cory, AE', 'Gileston,
AA', 'Cindyshire, AA', 'East Michaelfort, AA', 'Joybury, AE', 'Emmatown, A
E', 'Colehaven, AP', 'Port Coryton, AE', 'Amyborough, AA', 'Reynoldsville,
AA', 'Port Billy, AP', 'Adamburgh, AA', 'Wilsonmouth, AA', 'South Kimberl
y, AA', 'Benjaminland, AP', 'Zacharyport, AA', 'Port Devonville, AE', 'Eas
t Thomas, AE', 'New Jeffrey, AP', 'Davidside, AA', 'Jamesville, AA', 'New
Kelly, AP', 'Lake Antonio, AA', 'New Elizabethside, AA', 'Millsbury, AE',
'Lloydton, AP', 'Port Jeremy, AA', 'New Elizabethtown, AA', 'Charlesstad,
AE', 'Josephbury, AE', 'Seanfurt, AA', 'Williambury, AA', 'South Jorgesid
e, AP', 'Robertborough, AP', 'South Saratown, AP', 'Hullview, AA', 'Philip
land, AP', 'North Patty, AE', 'North Stephen, AE', 'Stevensland, AP', 'Rey
esstad, AE', 'Bellberg, AP', 'North Johnland, AE', 'Martinezburgh, AE', 'J
oshuatown, AE', 'West Ericstad, AA', 'Tuckertown, AE', 'Perezton, AE', 'La
ke Abigail, AE', 'Jacobshire, AP', 'Port Susan, AE', 'North Tiffany, AA',
'Michelleville, AP']
```

In [12]:
```
Location_Titles = pd.DataFrame(Location_Title,columns=["Location"])
Location_Titles
```

Out[12]:

	Location
0	Stewartbury, AA
1	Christopherville, AA
2	Port Ericaburgh, AA
3	East Seanview, AP
4	North Jamieview, AP
...	...
95	Lake Abigail, AE
96	Jacobshire, AP
97	Port Susan, AE
98	North Tiffany, AA
99	Michelleville, AP

100 rows × 1 columns

Add All Columns In One Data Frame

In [13]:
```
Job_Titles
```

Out[13]:

	Job Name
0	Senior Python Developer
1	Energy engineer
2	Legal executive
3	Fitness centre manager
4	Product manager
...	...
95	Museum/gallery exhibitions officer
96	Radiographer, diagnostic
97	Database administrator
98	Furniture designer
99	Ship broker

100 rows × 1 columns

In [13]:
```
Company_Titles
```

Out[13]:

	Company
0	Payne, Roberts and Davis
1	Vasquez-Davidson
2	Jackson, Chambers and Levy
3	Savage-Bradley
4	Ramirez Inc
...	...
95	Nguyen, Yoder and Petty
96	Holder LLC
97	Yates-Ferguson
98	Ortega-Lawrence
99	Fuentes, Walls and Castro

100 rows × 1 columns

In [21]:
```
Company_Data = pd.concat([Company_Titles, Job_Titles], axis=1)
Company_Data
```

Out[21]:

	Company	Job Name
0	Payne, Roberts and Davis	Senior Python Developer
1	Vasquez-Davidson	Energy engineer
2	Jackson, Chambers and Levy	Legal executive
3	Savage-Bradley	Fitness centre manager
4	Ramirez Inc	Product manager
...	...	...
95	Nguyen, Yoder and Petty	Museum/gallery exhibitions officer
96	Holder LLC	Radiographer, diagnostic
97	Yates-Ferguson	Database administrator
98	Ortega-Lawrence	Furniture designer
99	Fuentes, Walls and Castro	Ship broker

100 rows × 2 columns

In [22]:
```
Company_Data= pd.concat([Company_Data,Location_Titles],axis=1)
Company_Data
```

Out[22]:

	Company	Job Name	Location
0	Payne, Roberts and Davis	Senior Python Developer	Stewartbury, AA
1	Vasquez-Davidson	Energy engineer	Christopherville, AA
2	Jackson, Chambers and Levy	Legal executive	Port Ericaburgh, AA
3	Savage-Bradley	Fitness centre manager	East Seanview, AP
4	Ramirez Inc	Product manager	North Jamieview, AP
...	...	...	...
95	Nguyen, Yoder and Petty	Museum/gallery exhibitions officer	Lake Abigail, AE
96	Holder LLC	Radiographer, diagnostic	Jacobshire, AP
97	Yates-Ferguson	Database administrator	Port Susan, AE
98	Ortega-Lawrence	Furniture designer	North Tiffany, AA
99	Fuentes, Walls and Castro	Ship broker	Michelleville, AP

100 rows × 3 columns

In [23]:

```
Company_Data.info
```

Out[23]:

```
<bound method DataFrame.info of                        Company
Job Name  \
0     Payne, Roberts and Davis             Senior Python Developer
1             Vasquez-Davidson                     Energy engineer
2   Jackson, Chambers and Levy                     Legal executive
3               Savage-Bradley              Fitness centre manager
4                  Ramirez Inc                     Product manager
..                         ...                                 ...
95     Nguyen, Yoder and Petty  Museum/gallery exhibitions officer
96                  Holder LLC            Radiographer, diagnostic
97              Yates-Ferguson              Database administrator
98             Ortega-Lawrence                  Furniture designer
99   Fuentes, Walls and Castro                         Ship broker

                Location
0        Stewartbury, AA
1   Christopherville, AA
2    Port Ericaburgh, AA
3      East Seanview, AP
4    North Jamieview, AP
..                   ...
95      Lake Abigail, AE
96        Jacobshire, AP
97        Port Susan, AE
98     North Tiffany, AA
99     Michelleville, AP

[100 rows x 3 columns]>
```

In [24]:

```
Company_Data.dtypes
```

Out[24]:

```
Company     object
Job Name    object
Location    object
dtype: object
```