Basic Webscraping, League of Legends Characters

```
1 from bs4 import BeautifulSoup
2 import requests
```

```
1 # Getting the request of the website
2 request = requests.get("https://www.leagueofgraphs.com/champions/builds",
3                         headers= {"User-Agent":"Mozilla/5.0 (X11; Linux x86_64; r
4 request.status_code # itss 200 if its succesfully connected
```

```
200
```

```
1 lol = BeautifulSoup(request.text, 'html.parser') # Using BeautifulSooup for Webs
2 type(lol)
```

```
bs4.BeautifulSoup
def __call__(*args, **kwargs)

/usr/local/lib/python3.10/dist-packages/bs4/__init__.py
A data structure representing a parsed HTML or XML document.

Most of the methods you'll call on a BeautifulSoup object are inherited from
PageElement or Tag.
```

```
1 champion_container = lol.find_all('tr') # Finding the Common grounds in the data
```

```
1 first_champ = champion_container[1] # For prototyping
```

Name

```
1 first_champ.find("span", class_="name").text.strip()
```

```
'Smolder'
```

Role

```
1 first_champ.find("i", class_="").text.strip()
```

```
'AD Carry'
```

Kill

```
first_champ.find('span', class_='kills').text
```

```
'7.7'
```

Death

```
first_champ.find('span', class_='deaths').text
```

```
'6.1'
```

Assist

```
first_champ.find('span', class_='assists').text
```

```
'7.1'
```

Pentakill Per Match

```
first_champ.find_all('td', class_='text-center hide-for-small-down')[1].text.s
```

```
'0.0106'
```

Script

```
champ = []
role = []
kill = []
death = []
assist = []
penta = []

for i in champion_container:
  if i.find("span", class_="name") is not None:  # Filters the None Values
    champ.append(i.find("span", class_="name").text.strip())
```

```
    role.append(i.find("i", class_="").text.strip())
    kill.append(float(i.find('span', class_='kills').text))
    death.append(float(i.find('span', class_='deaths').text))
    assist.append(float(i.find('span', class_='assists').text))
    penta.append(float(i.find_all('td', class_='text-center hide-for-small-down
```

```
import pandas as pd

leaguedf = pd.DataFrame(
    {
        'champion':champ,
        'role':role,
        'kill':kill,
        'death':death,
        'assist':assist,
        "pentapermatch":penta
    }
)

print(leaguedf.info())
leaguedf.head(10)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 167 entries, 0 to 166
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   champion       167 non-null    object
 1   role           167 non-null    object
 2   kill           167 non-null    float64
 3   death          167 non-null    float64
 4   assist         167 non-null    float64
 5   pentapermatch  167 non-null    float64
dtypes: float64(4), object(2)
memory usage: 8.0+ KB
None
```

	champion	role	kill	death	assist	pentapermatch
0	Smolder	AD Carry	7.7	6.1	7.1	0.0106
1	Kai'Sa	AD Carry	8.5	6.3	6.7	0.0026
2	Karma	Mid, Support	4.2	5.9	10.9	0.0002
3	Lee Sin	Jungler	7.5	6.1	8.7	0.0007
4	Ezreal	AD Carry	7.3	6.1	7.2	0.0013
5	Volibear	Jungler, Top	5.7	5.7	8.0	0.0003
6	Senna	Support, AD Carry	5.5	6.3	12.1	0.0009
7	Ahri	Mid	6.6	5.4	7.8	0.0015

8	Lux	Support, Mid	5.2	6.3	11.0	0.0001
9	Yasuo	Mid	6.6	7.3	5.5	0.0012

Next steps: View recommended plots

```
1 leaguedf.to_csv("/content/leaguebasic.csv")
```