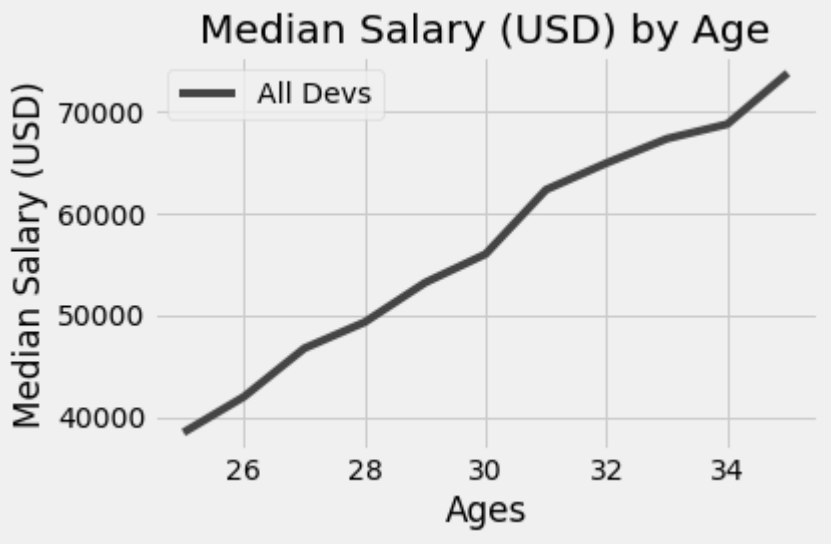
Matplotlib Tutorial (Part 2)

Bar Charts and Analyzing Data from CSVs

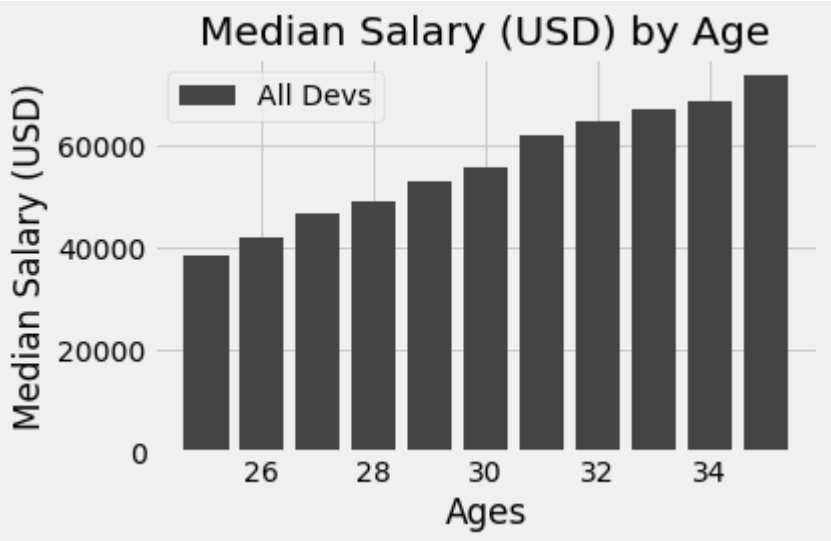
In [1]:
```
from matplotlib import pyplot as plt

plt.style.use("fivethirtyeight")

ages_x = [25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35]

dev_y = [38496, 42000, 46752, 49320, 53200,
         56000, 62316, 64928, 67317, 68748, 73752]
plt.plot(ages_x, dev_y, color="#444444", label="All Devs")

# py_dev_y = [45372, 48876, 53850, 57287, 63016,
#             65998, 70003, 70000, 71496, 75370, 83640]
# plt.plot(ages_x, py_dev_y, color="#008fd5", label="Python")

# js_dev_y = [37810, 43515, 46823, 49293, 53437,
#             56373, 62375, 66674, 68745, 68746, 74583]
# plt.plot(ages_x, js_dev_y, color="#e5ae38", label="JavaScript")

plt.legend()

plt.title("Median Salary (USD) by Age")
plt.xlabel("Ages")
plt.ylabel("Median Salary (USD)")

plt.tight_layout()

plt.show()
```



Change the plot to bar

In [2]:

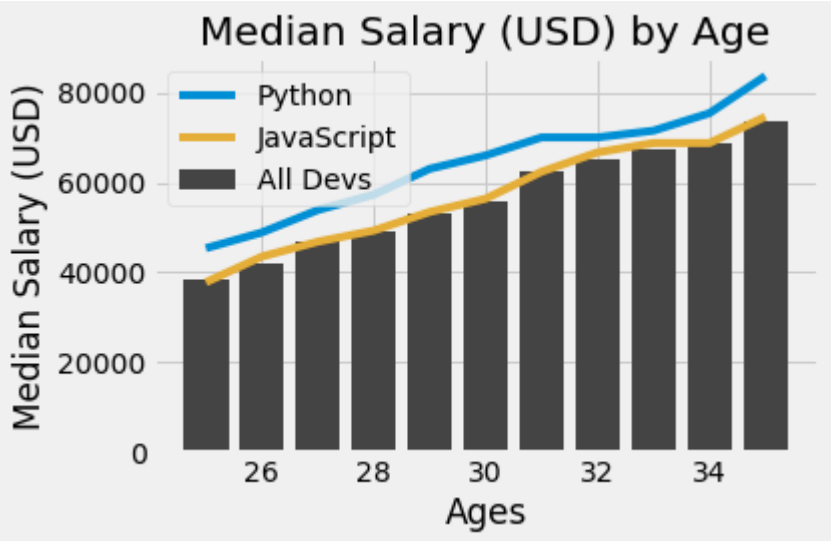
```
from matplotlib import pyplot as plt

plt.style.use("fivethirtyeight")

ages_x = [25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35]

dev_y = [38496, 42000, 46752, 49320, 53200,
         56000, 62316, 64928, 67317, 68748, 73752]
plt.bar(ages_x, dev_y, color="#444444", label="All Devs")

# py_dev_y = [45372, 48876, 53850, 57287, 63016,
#             65998, 70003, 70000, 71496, 75370, 83640]
# plt.plot(ages_x, py_dev_y, color="#008fd5", label="Python")

# js_dev_y = [37810, 43515, 46823, 49293, 53437,
#             56373, 62375, 66674, 68745, 68746, 74583]
# plt.plot(ages_x, js_dev_y, color="#e5ae38", label="JavaScript")

plt.legend()

plt.title("Median Salary (USD) by Age")
plt.xlabel("Ages")
plt.ylabel("Median Salary (USD)")

plt.tight_layout()

plt.show()
```



Overlay the other data

In [3]:

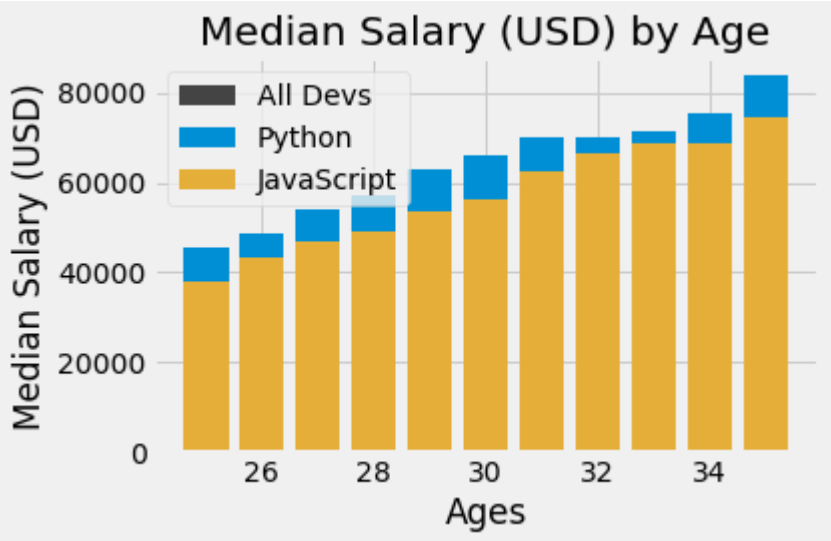
```
from matplotlib import pyplot as plt

plt.style.use("fivethirtyeight")

ages_x = [25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35]

dev_y = [38496, 42000, 46752, 49320, 53200,
         56000, 62316, 64928, 67317, 68748, 73752]
plt.bar(ages_x, dev_y, color="#444444", label="All Devs")

py_dev_y = [45372, 48876, 53850, 57287, 63016,
            65998, 70003, 70000, 71496, 75370, 83640]
plt.plot(ages_x, py_dev_y, color="#008fd5", label="Python")

js_dev_y = [37810, 43515, 46823, 49293, 53437,
            56373, 62375, 66674, 68745, 68746, 74583]
plt.plot(ages_x, js_dev_y, color="#e5ae38", label="JavaScript")

plt.legend()

plt.title("Median Salary (USD) by Age")
plt.xlabel("Ages")
plt.ylabel("Median Salary (USD)")

plt.tight_layout()

plt.show()
```



use bar plot for everything

In [4]:

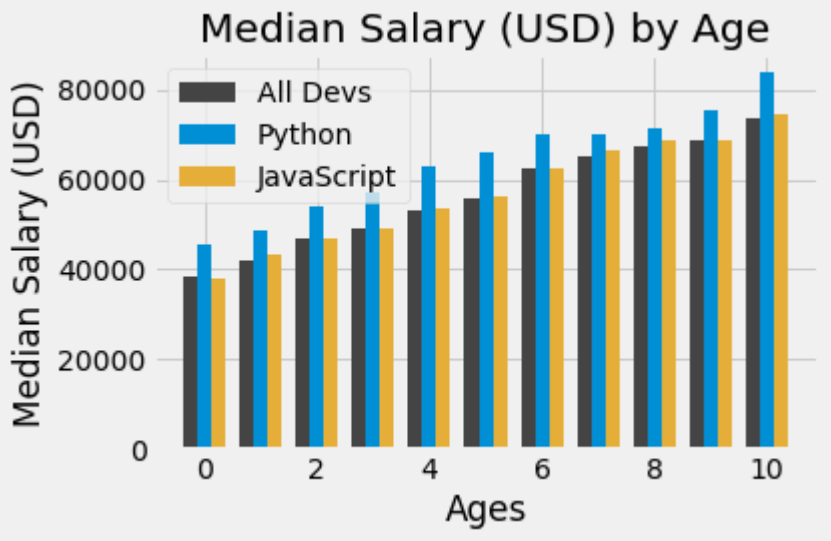
```
from matplotlib import pyplot as plt

plt.style.use("fivethirtyeight")

ages_x = [25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35]

dev_y = [38496, 42000, 46752, 49320, 53200,
         56000, 62316, 64928, 67317, 68748, 73752]
plt.bar(ages_x, dev_y, color="#444444", label="All Devs")

py_dev_y = [45372, 48876, 53850, 57287, 63016,
            65998, 70003, 70000, 71496, 75370, 83640]
plt.bar(ages_x, py_dev_y, color="#008fd5", label="Python")

js_dev_y = [37810, 43515, 46823, 49293, 53437,
            56373, 62375, 66674, 68745, 68746, 74583]
plt.bar(ages_x, js_dev_y, color="#e5ae38", label="JavaScript")

plt.legend()

plt.title("Median Salary (USD) by Age")
plt.xlabel("Ages")
plt.ylabel("Median Salary (USD)")

plt.tight_layout()

plt.show()
```



Adjust the graphs using numpy

In [5]:

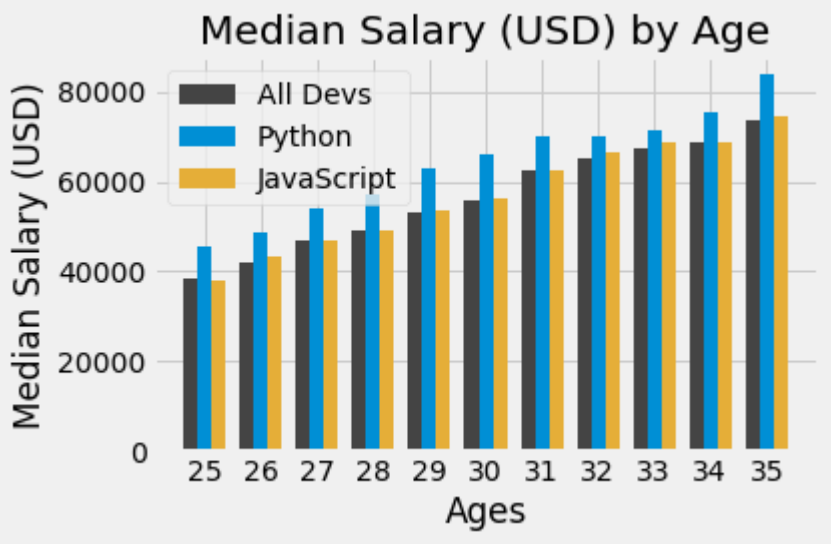
```
import numpy as np
from matplotlib import pyplot as plt

plt.style.use("fivethirtyeight")

ages_x = [25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35]
x_indexes = np.arange(len(ages_x)) # range of the length of ages

width = 0.25 #new width = subtract to the first values and add to the last va

dev_y = [38496, 42000, 46752, 49320, 53200,
         56000, 62316, 64928, 67317, 68748, 73752]
plt.bar(x_indexes - width, dev_y, width=width, color="#444444", label="All Dev

py_dev_y = [45372, 48876, 53850, 57287, 63016,
            65998, 70003, 70000, 71496, 75370, 83640]
plt.bar(x_indexes, py_dev_y, width=width, color="#008fd5", label="Python")

js_dev_y = [37810, 43515, 46823, 49293, 53437,
            56373, 62375, 66674, 68745, 68746, 74583]
plt.bar(x_indexes + width, js_dev_y, width=width, color="#e5ae38", label="Java

plt.legend()

plt.title("Median Salary (USD) by Age")
plt.xlabel("Ages")
plt.ylabel("Median Salary (USD)")

plt.tight_layout()

plt.show()
```



Change ticks label to change the x-values

In [6]:

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

plt.style.use("fivethirtyeight")

ages_x = [25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35]
x_indexes = np.arange(len(ages_x)) # range of the length of ages

width = 0.25 #new width = subtract to the first values and add to the last va

dev_y = [38496, 42000, 46752, 49320, 53200,
         56000, 62316, 64928, 67317, 68748, 73752]
plt.bar(x_indexes - width, dev_y, width=width, color="#444444", label="All Dev

py_dev_y = [45372, 48876, 53850, 57287, 63016,
            65998, 70003, 70000, 71496, 75370, 83640]
plt.bar(x_indexes, py_dev_y, width=width, color="#008fd5", label="Python")

js_dev_y = [37810, 43515, 46823, 49293, 53437,
            56373, 62375, 66674, 68745, 68746, 74583]
plt.bar(x_indexes + width, js_dev_y, width=width, color="#e5ae38", label="Java

plt.legend()

plt.title("Median Salary (USD) by Age")
plt.xlabel("Ages")
plt.ylabel("Median Salary (USD)")

plt.xticks(ticks=x_indexes, labels=ages_x) #revert back to ages

plt.tight_layout()

plt.show()
```



Use real-world data

In [8]:
```
import csv
import numpy as np
import pandas as pd
from collections import Counter
from matplotlib import pyplot as plt

plt.style.use("fivethirtyeight")

data = pd.read_csv('data2.txt')
ids = data['Responder_id']
lang_responses = data['LanguagesWorkedWith']
```

In [9]:
```
data
```

Out[9]:

	Responder_id	LanguagesWorkedWith
0	1	HTML/CSS;Java;JavaScript;Python
1	2	C++;HTML/CSS;Python
2	3	HTML/CSS
3	4	C;C++;C#;Python;SQL
4	5	C++;HTML/CSS;Java;JavaScript;Python;SQL;VBA
...	...	...
87564	88182	HTML/CSS;Java;JavaScript
87565	88212	HTML/CSS;JavaScript;Python
87566	88282	Bash/Shell/PowerShell;Go;HTML/CSS;JavaScript;W...
87567	88377	HTML/CSS;JavaScript;Other(s):
87568	88863	Bash/Shell/PowerShell;HTML/CSS;Java;JavaScript...

87569 rows × 2 columns

Changing the separator using 'counter' method

- It is a sub-class that is used to `count hashable objects`. It implicitly creates a hash table of an iterable when invoked. elements() is one of the functions of Counter class, when invoked on the Counter object will return an itertool of all the known elements in the Counter object
- Do this at the columns/variables you wish to update
- Using loop

In [22]:
```
import numpy as np
import pandas as pd
from collections import Counter
from matplotlib import pyplot as plt

plt.style.use("fivethirtyeight")

data = pd.read_csv('data2.txt')
ids = data['Responder_id']
lang_responses = data['LanguagesWorkedWith']

language_counter = Counter()

for response in lang_responses:
    language_counter.update(response.split(';'))
```

In [24]:
```
language_counter # after looping, it will create a list of tuples separated by
```

Out[24]:
```
Counter({'HTML/CSS': 55466,
         'Java': 35917,
         'JavaScript': 59219,
         'Python': 36443,
         'C++': 20524,
         'C': 18017,
         'C#': 27097,
         'SQL': 47544,
         'VBA': 4781,
         'R': 5048,
         'Bash/Shell/PowerShell': 31991,
         'Ruby': 7331,
         'Rust': 2794,
         'TypeScript': 18523,
         'WebAssembly': 1015,
         'Other(s):': 7920,
         'Go': 7201,
         'PHP': 23030,
         'Assembly': 5833,
         'Kotlin': 5620,
         'Swift': 5744,
         'Objective-C': 4191,
         'Elixir': 1260,
         'Erlang': 777,
         'Clojure': 1254,
         'F#': 973,
         'Scala': 3309,
         'Dart': 1683})
```

Create a new list

- One for languages and popularity
- You append the top 15 most common programming languages to those empty lists

In [34]:
```
import numpy as np
import pandas as pd
from collections import Counter
from matplotlib import pyplot as plt

plt.style.use("fivethirtyeight")

data = pd.read_csv('data2.txt')
ids = data['Responder_id']
lang_responses = data['LanguagesWorkedWith']

language_counter = Counter() # you invoked the counter method and stored it i

for response in lang_responses:
    language_counter.update(response.split(';')) #loop through the 'lang_resp

#create empty lists to separate two variables of interest

languages = []
popularity = []

for item in language_counter.most_common(15): #the most common function is bu
    languages.append(item[0]) #append to the languages variable the first ite
    popularity.append(item[1])#append to the popularity variable the second i
# and it will iterate through each loop
```

In [13]:
```
languages # we see that all the values on programming langauges are stored in
```

Out[13]:
```
['JavaScript',
 'HTML/CSS',
 'SQL',
 'Python',
 'Java',
 'Bash/Shell/PowerShell',
 'C#',
 'PHP',
 'C++',
 'TypeScript',
 'C',
 'Other(s):',
 'Ruby',
 'Go',
 'Assembly']
```

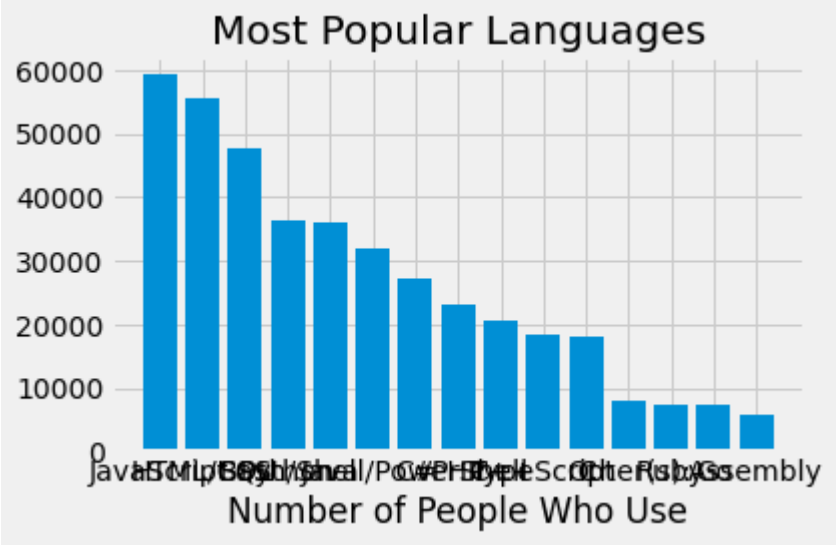
In [35]:
```
popularity # this one too
```

Out[35]:
```
[59219,
 55466,
 47544,
 36443,
 35917,
 31991,
 27097,
 23030,
 20524,
```

```
18523,
18017,
7920,
7331,
7201,
```

In [36]:

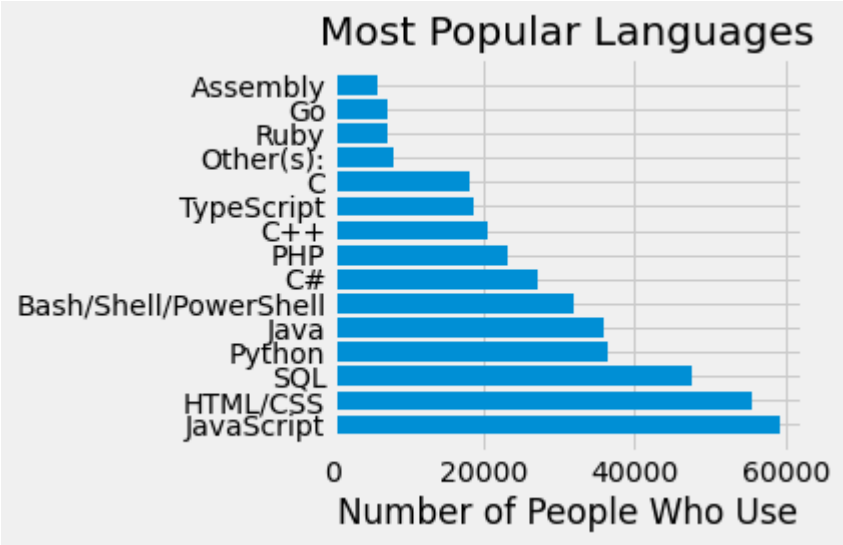
```
import csv
import numpy as np
import pandas as pd
from collections import Counter
from matplotlib import pyplot as plt

plt.style.use("fivethirtyeight")

data = pd.read_csv('data2.txt')
ids = data['Responder_id']
lang_responses = data['LanguagesWorkedWith']

language_counter = Counter()

for response in lang_responses:
    language_counter.update(response.split(';'))

languages = []
popularity = []

for item in language_counter.most_common(15):
    languages.append(item[0])
    popularity.append(item[1])

plt.bar(languages, popularity)

plt.title("Most Popular Languages")
# plt.ylabel("Programming Languages")
plt.xlabel("Number of People Who Use")

plt.tight_layout()

plt.show()
```



Use a horizontal graph

In [37]:

```
import csv
import numpy as np
import pandas as pd
from collections import Counter
from matplotlib import pyplot as plt

plt.style.use("fivethirtyeight")

data = pd.read_csv('data2.txt')
ids = data['Responder_id']
lang_responses = data['LanguagesWorkedWith']

language_counter = Counter()

for response in lang_responses:
    language_counter.update(response.split(';'))

languages = []
popularity = []

for item in language_counter.most_common(15):
    languages.append(item[0])
    popularity.append(item[1])

plt.barh(languages, popularity) # by adding 'h' at the end of bar

plt.title("Most Popular Languages")
# plt.ylabel("Programming Languages")
plt.xlabel("Number of People Who Use")

plt.tight_layout()

plt.show()
```

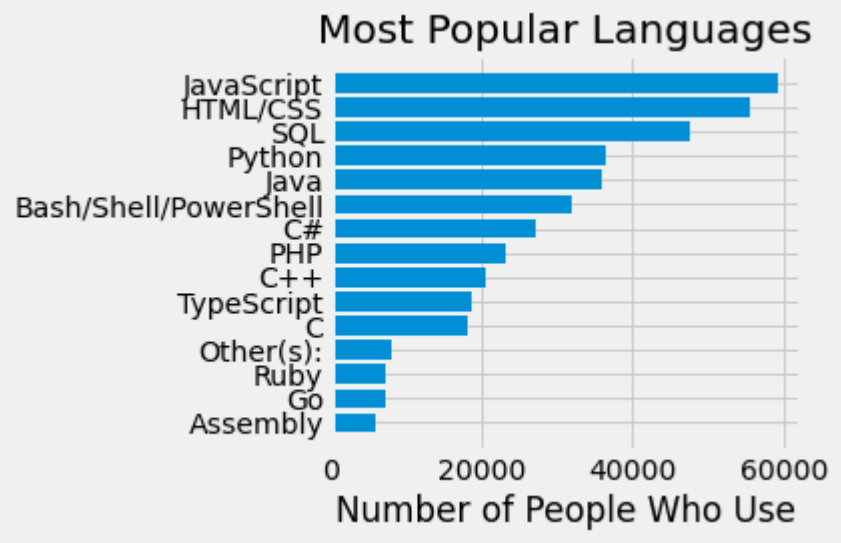


Reverse the graph to make the highest values

to appear from the top

In [39]:

```
import csv
import numpy as np
import pandas as pd
from collections import Counter
from matplotlib import pyplot as plt

plt.style.use("fivethirtyeight")

data = pd.read_csv('data2.txt')
ids = data['Responder_id']
lang_responses = data['LanguagesWorkedWith']

language_counter = Counter()

for response in lang_responses:
    language_counter.update(response.split(';'))

languages = []
popularity = []

for item in language_counter.most_common(15):
    languages.append(item[0])
    popularity.append(item[1])

languages.reverse()
popularity.reverse()

plt.barh(languages, popularity)

plt.title("Most Popular Languages")
# plt.ylabel("Programming Languages")
plt.xlabel("Number of People Who Use")

plt.tight_layout()

plt.show()
```



Final code

In [7]:

```
import csv
import numpy as np
import pandas as pd
from collections import Counter
from matplotlib import pyplot as plt

plt.style.use("fivethirtyeight")

data = pd.read_csv('data2.txt')
ids = data['Responder_id']
lang_responses = data['LanguagesWorkedWith']

language_counter = Counter()

for response in lang_responses:
    language_counter.update(response.split(';'))

languages = []
popularity = []

for item in language_counter.most_common(15):
    languages.append(item[0])
    popularity.append(item[1])

languages.reverse()
popularity.reverse()

plt.barh(languages, popularity)

plt.title("Most Popular Languages")
# plt.ylabel("Programming Languages")
plt.xlabel("Number of People Who Use")

plt.tight_layout()

plt.show()
```