In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn import datasets
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,confusion_matrix
from sklearn.linear_model import LogisticRegression
import matplotlib.pyplot as plt
```

In [2]:
```
df = sns.load_dataset('iris')
df.head()
```

Out[2]:

| | sepal_length | sepal_width | petal_length | petal_width | species |
|---|---|---|---|---|---|
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | setosa |

In [3]:
```
encoder = LabelEncoder()
df['species'] = encoder.fit_transform(df['species'])
df.head()
```

Out[3]:

| | sepal_length | sepal_width | petal_length | petal_width | species |
|---|---|---|---|---|---|
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | 0 |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | 0 |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | 0 |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 | 0 |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | 0 |

In [4]:
```
df = df[['sepal_length','petal_length','species']]
df.head()
```

Out[4]:

| | sepal_length | petal_length | species |
|---|---|---|---|
| **0** | 5.1 | 1.4 | 0 |
| **1** | 4.9 | 1.4 | 0 |
| **2** | 4.7 | 1.3 | 0 |
| **3** | 4.6 | 1.5 | 0 |
| **4** | 5.0 | 1.4 | 0 |

In [5]:
```
x = df.iloc[:,0:2]
y = df.iloc[:,-1]
```

In [6]:
```
x_train,x_test,y_train,y_test= train_test_split(x,y,test_size=0.2)
```

In [7]:
```
clf = LogisticRegression(multi_class='multinomial')
```

In [8]:
```
clf.fit(x_train,y_train)
```

Out[8]:
```
LogisticRegression(multi_class='multinomial')
```

In [9]:
```
y_pred = clf.predict(x_test)
```

In [10]:
```
print(accuracy_score(y_test,y_pred))
```

```
0.9666666666666667
```

In [11]:
```
pd.DataFrame(confusion_matrix(y_test,y_pred))
```

Out[11]:

| | 0 | 1 | 2 |
|---|---|---|---|
| **0** | 11 | 0 | 0 |
| **1** | 0 | 5 | 0 |
| **2** | 0 | 1 | 13 |

In [12]:
```
#prediction
query = np.array([[3.4,2.7]])
clf.predict_proba(query)
```

```
C:\Users\User15\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning:
X does not have valid feature names, but LogisticRegression was fitted with f
eature names
  warnings.warn(
```

Out[12]:
```
array([[7.96163121e-01, 2.03655804e-01, 1.81075531e-04]])
```

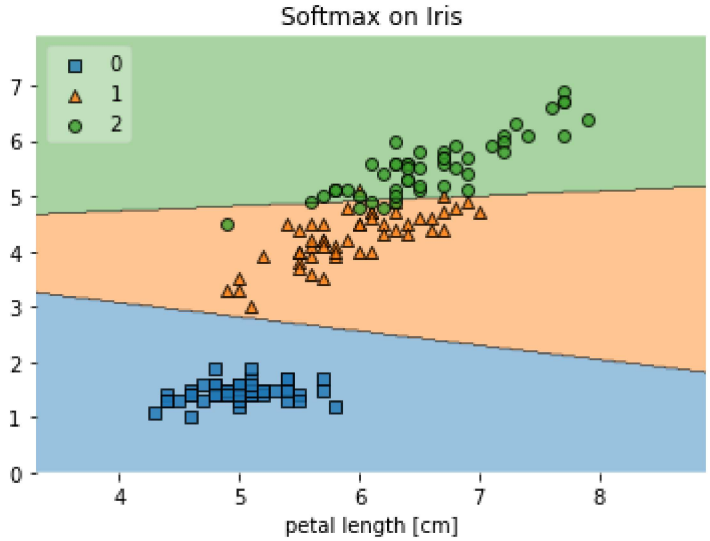
In [13]:
```
clf.predict(query)
```

```
C:\Users\User15\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning:
X does not have valid feature names, but LogisticRegression was fitted with f
eature names
  warnings.warn(
```

Out[13]:
```
array([0])
```

In [14]:
```
from mlxtend.plotting import plot_decision_regions
plot_decision_regions(x.values,y.values,clf,legend=2)
#Adding axes annotations
plt.xlabel('sepal length [cm]')
plt.xlabel('petal length [cm]')
plt.title('Softmax on Iris')
plt.show()
```

```
C:\Users\User15\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning:
X does not have valid feature names, but LogisticRegression was fitted with f
eature names
  warnings.warn(
```




In [ ]: