In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
language_data = pd.read_csv("language.csv")
```

In [3]:

```
language_data.head()
```

Out[3]:

	id	file_path	file_size	line_count	extenstion	language
0	NaN	NaN	NaN	NaN	NaN	NaN
1	1.0	./dataset\Markdown\000001.md	34784.0	572.0	md	Markdown
2	NaN	NaN	NaN	NaN	NaN	NaN
3	2.0	./dataset\XML\000002.props	3013.0	44.0	props	XML
4	NaN	NaN	NaN	NaN	NaN	NaN

In [4]:

```
language_data.tail()
```

Out[4]:

	id	file_path	file_size	line_count	extenstion	language
170283	85142.0	./dataset\Swift\085142.swift	2223.0	70.0	swift	Swift
170284	NaN	NaN	NaN	NaN	NaN	NaN
170285	85143.0	./dataset\Swift\085143.swift	1106.0	38.0	swift	Swift
170286	NaN	NaN	NaN	NaN	NaN	NaN
170287	85144.0	./dataset\Swift\085144.swift	1522.0	58.0	swift	Swift

In [5]:

```
language_data.shape
```

Out[5]:

```
(170288, 6)
```

In [6]:

```
language_data.columns
```

Out[6]:

```
Index(['id', 'file_path', 'file_size', 'line_count', 'extenstion', 'langua
ge'], dtype='object')
```

In [7]:

```
language_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 170288 entries, 0 to 170287
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   id          85144 non-null  float64
 1   file_path   85144 non-null  object 
 2   file_size   85144 non-null  float64
 3   line_count  85144 non-null  float64
 4   extenstion  85144 non-null  object 
 5   language    85144 non-null  object 
dtypes: float64(3), object(3)
memory usage: 7.8+ MB
```

In [8]:

```
language_data.describe()
```

Out[8]:

	id	file_size	line_count
count	85144.000000	8.514400e+04	85144.000000
mean	42572.500000	1.342207e+04	257.873180
std	24579.099997	2.899809e+05	4940.246601
min	1.000000	0.000000e+00	0.000000
25%	21286.750000	4.390000e+02	18.000000
50%	42572.500000	1.398000e+03	46.000000
75%	63858.250000	4.510000e+03	128.000000
max	85144.000000	2.700998e+07	580544.000000

In [9]:

```
language_data.dropna(inplace = True)
```

In [10]:

```
language_data.shape
```

Out[10]:

```
(85144, 6)
```

In [12]:

```
language_data.groupby(['extenstion'])['language'].nunique().sort_values().sort_values(as
```

Out[12]:

```
extenstion
h               2
jl              1
dfm             1
cc              1
Makefile        1
adoc            1
applescript     1
asd             1
asm             1
awk             1
b               1
bash            1
bat             1
bib             1
boot            1
Name: language, dtype: int64
```

In [13]:

```
language_data.groupby(['language'])['extenstion'].nunique().sort_values(ascending=False)
```

Out[13]:

```
language
XML            13
Ruby            6
GLSL            5
Erlang          5
C++             4
Pascal          3
PowerShell      3
Shell           3
INI             3
Clojure         3
Scala           2
Markdown        2
Kotlin          2
JSON            2
Javascript      2
Name: extenstion, dtype: int64
```

In [14]:

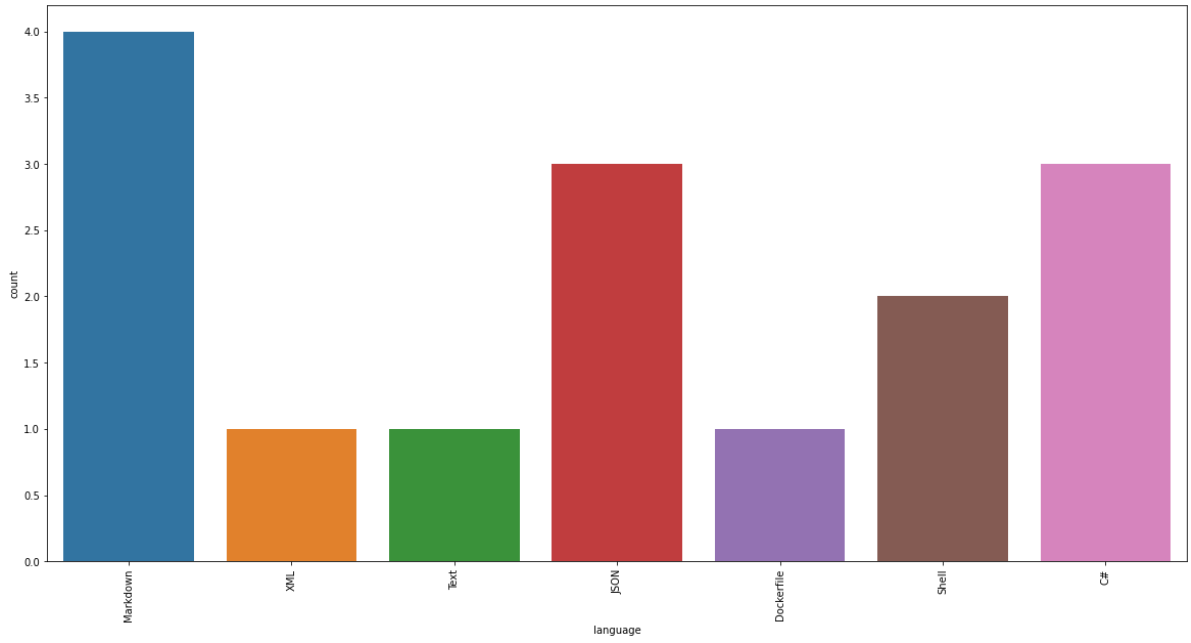
```
language_data[language_data['extenstion'] == 'h'].groupby(['extenstion', 'language'])['
```

Out[14]:

		len
extenstion	**language**	
h	**C**	353
	C++	357

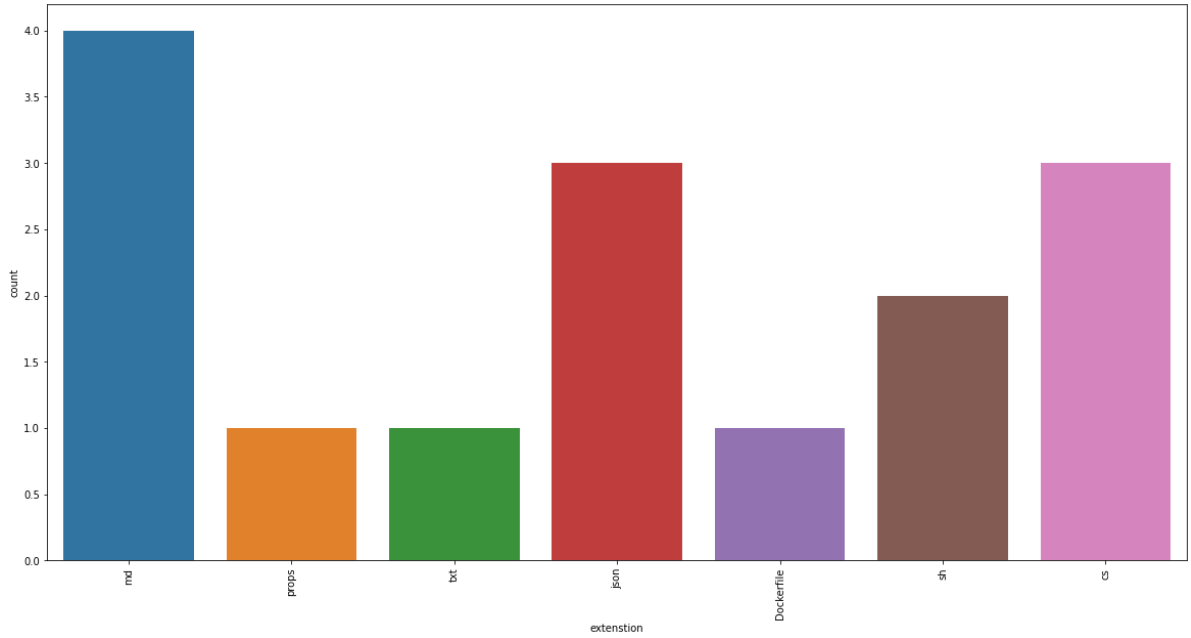
In [28]:

```
plt.figure(figsize=(20,10))
sns.countplot(x='language',data = language_data.head(15))
plt.xticks(rotation = 90)
plt.show()
```



In [30]:

```
plt.figure(figsize=(20,10))
sns.countplot(x='extenstion',data = language_data.head(15))
plt.xticks(rotation = 90)
plt.show()
```



In [35]:

```
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()
language_data['language']= label_encoder.fit_transform(language_data['language'])
language_data['extenstion']= label_encoder.fit_transform(language_data['extenstion'])
```

In [36]:

```
language_data['language'].unique()
```

Out[36]:

```
array([46, 73, 71, 33, 17, 66,  6, 75, 50, 30, 63,  9, 35,  4,  5, 45,  7,
       10, 11, 34, 32, 55, 15,  8, 52, 16, 61, 70, 26, 43, 24, 42, 20, 18,
       22, 51,  3, 48, 14, 23, 41, 31, 67, 36,  0, 69, 65, 53, 39, 38, 27,
       28, 40, 72,  2, 29, 60, 25, 13, 47, 21, 56, 37, 58, 19, 57, 76, 12,
       62, 49, 59, 74, 44, 54, 64,  1, 68])
```

In [37]:

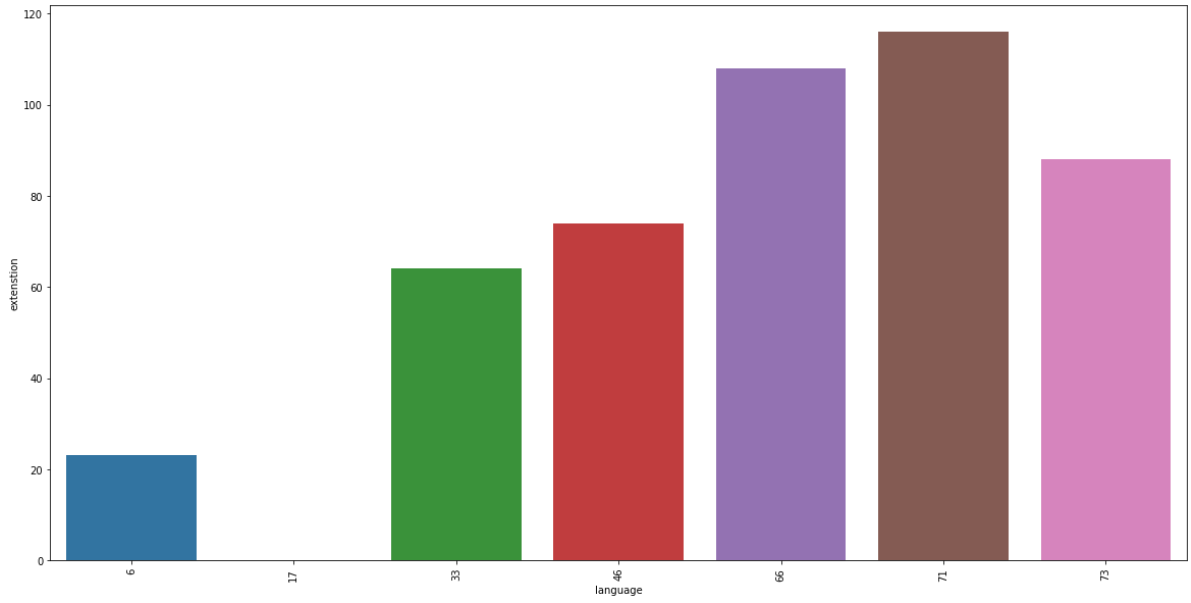
```
language_data['extenstion'].unique()
```

Out[37]:

```
array([ 74,  88, 116,  64,   0, 108,  23,  25, 112, 127,  90,  56, 110,
        26,  63,  20,  92,  24, 122,  91,  32,  13,  52,   1,  22,  53,
        27, 126, 121, 117,  43,  79,  17,  73,  61,  87,  18,  94,  29,
        19,  14,  89,   9,  82,  15,  77, 107, 113,  49,  59,  72, 102,
        68,  41,  40,  35,  39,  38,  54, 128, 123,  84,   6,  83,  28,
         8,  42,  78,  16,   7,  55,  70,  10,  58,  76,  62,  85,   3,
       114, 106,  71,  11,  93,  69,  66,  65,  57,  50,  33,  67, 115,
         5,  51, 103,  46, 119,   4,  80,  36,  95,  60,  45,  47,  97,
        37,  96, 100,  98, 125, 101,  12,  21, 109,  34,  30,  81,  99,
        31, 124, 120,  75,  86, 104, 105,   2, 111,  44,  48, 118])
```

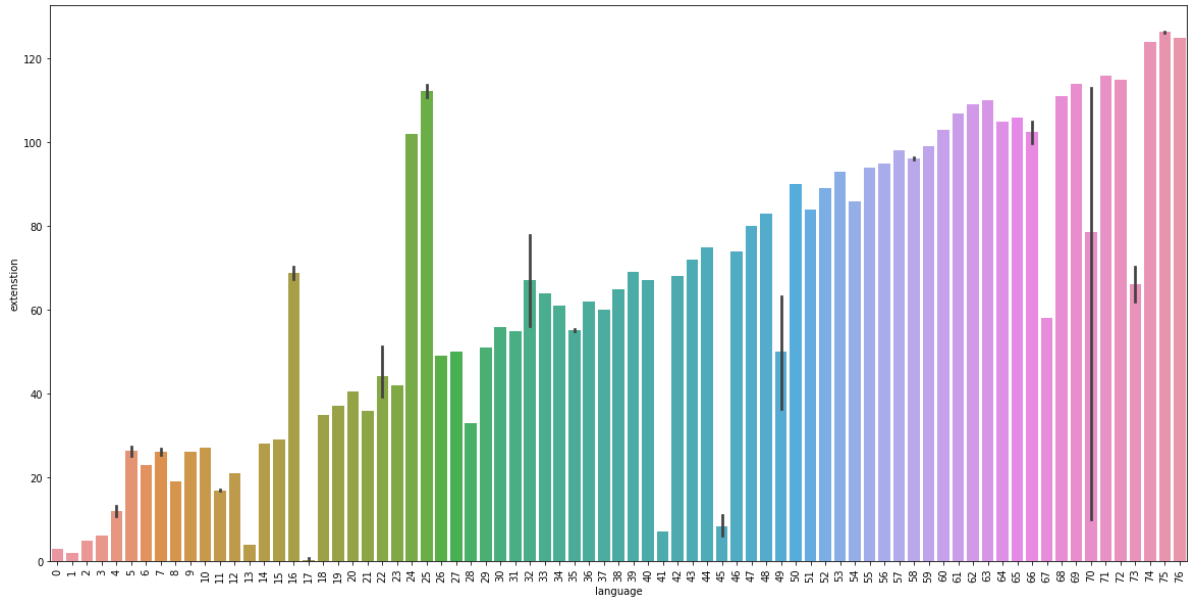
In [40]:

```
plt.figure(figsize=(20,10))
sns.barplot(x='language',y = 'extenstion', data = language_data.head(15))
plt.xticks(rotation = 90)
plt.show()
```



In [42]:

```
plt.figure(figsize=(20,10))
sns.barplot(x='language',y = 'extenstion', data = language_data)
plt.xticks(rotation = 90)
plt.show()
```



In [44]:

```
x = language_data.drop(['id','file_path', 'language'], axis = 1)
y = language_data.language
```

In [45]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

In [46]:

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.3)
```

In [47]:

```
model = LogisticRegression()
model.fit(X_train, y_train)
```

Out[47]:

```
LogisticRegression()
```

In [48]:

```
y_pred = model.predict(X_test)
```

In [49]:

```
print("Training Accuracy :", model.score(X_train, y_train))
print("Testing Accuracy :", model.score(X_test, y_test))
```

```
Training Accuracy : 0.25864093959731543
Testing Accuracy : 0.2535624804259317
```