1 - DEFINE THE PROBLEM

We are exploring the contents of the `imagenet_class_names.txt` file, which includes the class labels used in ImageNet-based deep learning models. This notebook will help us understand the label structure and prepare it for use in machine learning pipelines.

2 - IMPORT REQUIRED LIBRARIES

2.1 - Base Libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import string
```

2.2 - ML/DL Libraries

In [2]:
```
from sklearn.model_selection import train_test_split
```

3 - LOAD THE DATA

In [3]:
```
with open('../datasets/imagenet_class_names.txt', 'r') as file:
    class_names = [line.strip() for line in file.readlines() if line.strip()]

print(f'Total classes: {len(class_names)}')
class_names[:10]  # Preview first 10
```

```
Total classes: 1000
```

Out[3]:
```
['n01440764 tench, Tinca tinca',
 'n01443537 goldfish, Carassius auratus',
 'n01484850 great white shark, white shark, man-eater, man-eating shark, Carcharodon carcharias',
 'n01491361 tiger shark, Galeocerdo cuvieri',
 'n01494475 hammerhead, hammerhead shark',
 'n01496331 electric ray, crampfish, numbfish, torpedo',
 'n01498041 stingray',
 'n01514668 cock',
 'n01514859 hen',
 'n01518878 ostrich, Struthio camelus']
```

4 - EDA (Exploratory Data Analysis)

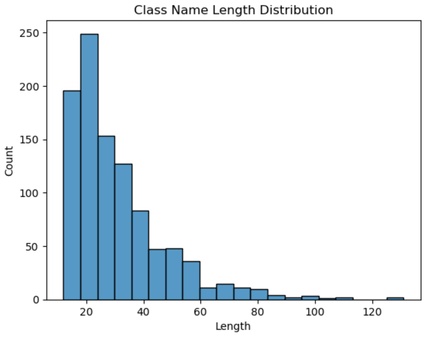
In [4]:
```
# Convert to DataFrame for analysis
df = pd.DataFrame(class_names, columns=['class_name'])
df['length'] = df['class_name'].apply(len)
df['first_letter'] = df['class_name'].str[0]
df.describe(include='all')
```

Out[4]:

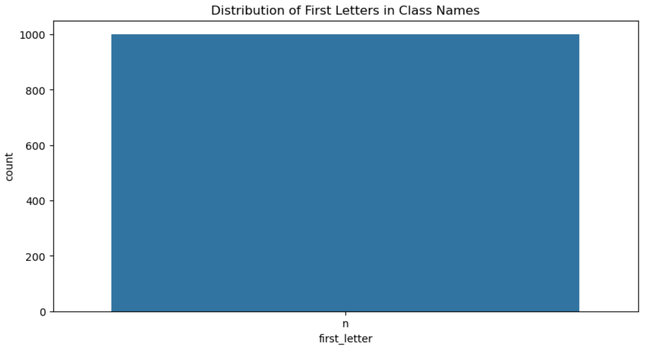
	class_name	length	first_letter
count	1000	1000.000000	1000
unique	1000	NaN	1
top	n01440764 tench, Tinca tinca	NaN	n
freq	1	NaN	1000
mean	NaN	30.675000	NaN
std	NaN	16.886638	NaN
min	NaN	12.000000	NaN
25%	NaN	18.000000	NaN
50%	NaN	26.000000	NaN
75%	NaN	37.000000	NaN
max	NaN	131.000000	NaN

5 - VISUALIZE THE DATA

In [5]:
```
# Histogram of class name lengths
sns.histplot(df['length'], bins=20)
plt.title('Class Name Length Distribution')
plt.xlabel('Length')
plt.ylabel('Count')
plt.show()
```



In [6]:
```
# Frequency of first letters
plt.figure(figsize=(10, 5))
sns.countplot(x='first_letter', data=df,
              order=sorted(df['first_letter'].unique()))
plt.title('Distribution of First Letters in Class Names')
plt.show()
```



6 - PREPROCESS THE DATA

In [7]:
```
# Basic standardization
df['class_name_clean'] = df['class_name'].str.lower().str.replace('-', ' ').str.replace('_', ' ')
df['class_name_clean'] = df['class_name_clean'].str.replace(r'[^a-z ]+', '', regex=True)
df.head()
```

Out[7]:

	class_name	length	first_letter	class_name_clean
0	n01440764 tench, Tinca tinca	28	n	n tench tinca tinca
1	n01443537 goldfish, Carassius auratus	37	n	n goldfish carassius auratus
2	n01484850 great white shark, white shark, man-...	93	n	n great white shark white shark man eater man ...
3	n01491361 tiger shark, Galeocerdo cuvieri	41	n	n tiger shark galeocerdo cuvieri
4	n01494475 hammerhead, hammerhead shark	38	n	n hammerhead hammerhead shark

7 - SPLIT THE DATA (Optional example)

In [8]:
```
# We can split classes into groups for manual use (e.g., 80% train, 20% holdout)
train_classes, test_classes = train_test_split(df['class_name_clean'], test_size=0.2, random_state=42)
```

In [9]:
```
print(f'Train classes: {len(train_classes)}')
```

```
Train classes: 800
```

In [10]:
```
print(f'Test classes: {len(test_classes)}')
```

```
Test classes: 200
```

In [11]:
```
train_classes[:5]
```

Out[11]:
```
29     n axolotl mud puppy ambystoma mexicanum
535                  n disk brake disc brake
695                           n padlock
557                     n flagpole flagstaff
836                          n sunglass
Name: class_name_clean, dtype: object
```