# 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]:

## 2.2 - ML/DL Libraries

In [2]:

# 3 - LOAD THE DATA

In [3]:
```
# Step 1: Open the file 'imagenet_class_names.txt' from the ../datasets/ directory using a with statement.
# Use read mode ('r') and assign the file handle to a variable.

# Step 2: Read all the lines from the file.
# Use a list comprehension to:
#   - Strip newline characters from each line using .strip()
#   - Skip empty lines

# Step 3: Store the result in a list called class_names.

with open


# Step 4: Print the total number of class names using len(class_names).
print(f'Total classes: {len(class_names)}')

# Step 5: Display the first 10 class entries to preview the data format.
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]:
```
# Step 1: Convert the list of class names into a DataFrame with one column named 'class_name'.
# pd.DataFrame( ...

# Step 2: Add a column 'length' with the character count of each class name.
# df['length'] = ...

# Step 3: Add a column 'first_letter' with the first character of each class name.
# df['first_letter'] = ...

# Step 4: Display summary statistics, including object columns.
# 👉 df.describe(include= ...
```

Out[4]:

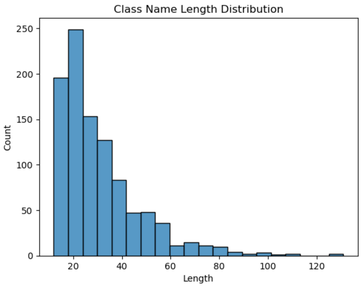
| | class_name | length | first_letter |
|---|---|---|---|
| **count** | 1000 | 1000.000000 | 1000 |
| **unique** | 1000 | NaN | 1 |
| **top** | n01440764 tench, Tinca tinca | NaN | n |
| **freq** | 1 | NaN | 1000 |
| **mean** | NaN | 30.675000 | NaN |
| **std** | NaN | 16.886638 | NaN |
| **min** | 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]:
```
# Step 1: Create a histogram to visualize the distribution of class name lengths.
# sns.histplot( ...

# Step 2: Add a title and axis labels to explain the plot.
# plt.title( ...
# plt.xlabel( ...
# plt.ylabel( ...

# Step 3: Show the plot.
```

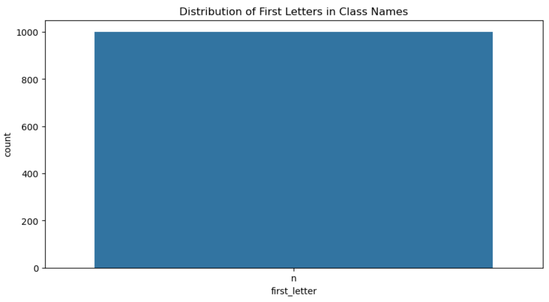



In [6]:
```
# Step 1: Set a custom figure size to improve readability.
# plt.figure( ...

# Step 2: Create a bar chart showing how many class names start with each letter.
# Use sorted() to order the x-axis alphabetically.
# sns.countplot( ...

# Step 3: Add a title to describe the purpose of the plot.
# plt.title( ...

# Step 4: Show the plot.
# plt.show()
```




# 6 - PREPROCESS THE DATA

In [7]:
```
# Step 1: Create a new column called 'class_name_clean'.
# Convert all text to lowercase using .str.lower().
# Replace hyphens and underscores with spaces using .str.replace().
# df['class_name_clean'] = ...

# Step 2: Remove all non-letter characters using a regular expression.
# Use .str.replace() with regex=True.
# df['class_name_clean'] = ...

# Step 3: Show the first 5 rows to verify the cleaning process.
#
```

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

In [8]:
```
# Step 1: Use train_test_split to divide 'class_name_clean' into train and test sets.
# Use test_size=0.2 and random_state=42 for reproducibility.
# train_classes, test_classes =  ...
```

In [9]:
```
# Step 2: Print the number of train classes.
# print( ...
```

```
Train classes: 800
```

In [10]:
```
# Step 3: Print the number of test classes.
# print ...
```

```
Test classes: 200
```

In [11]:
```
# Step 4: Preview the first 5 training class names.
# train_classes[: ...
```

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
```