

```
In [1]: !pip install boto3

Requirement already satisfied: boto3 in c:\users\nani\anaconda3\lib\site-packages
(1.35.44)
Collecting botocore<1.36.0,>=1.35.44 (from boto3)
  Using cached botocore-1.35.44-py3-none-any.whl.metadata (5.7 kB)
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in c:\users\nani\anaconda3\lib
\site-packages (from boto3) (1.0.1)
Requirement already satisfied: s3transfer<0.11.0,>=0.10.0 in c:\users\nani\anaconda3
\lib\site-packages (from boto3) (0.10.3)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in c:\users\nani\anaconda
3\lib\site-packages (from botocore<1.36.0,>=1.35.44->boto3) (2.9.0.post0)
Requirement already satisfied: urllib3!=2.2.0,<3,>=1.25.4 in c:\users\nani\anaconda3
\lib\site-packages (from botocore<1.36.0,>=1.35.44->boto3) (2.2.2)
Requirement already satisfied: six>=1.5 in c:\users\nani\anaconda3\lib\site-packages
(from python-dateutil<3.0.0,>=2.1->botocore<1.36.0,>=1.35.44->boto3) (1.16.0)
Using cached botocore-1.35.44-py3-none-any.whl (12.6 MB)
Installing collected packages: botocore
  Attempting uninstall: botocore
    Found existing installation: botocore 1.34.69
    Uninstalling botocore-1.34.69:
      Successfully uninstalled botocore-1.34.69
Successfully installed botocore-1.35.44

ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
aiobotocore 2.12.3 requires botocore<1.34.70,>=1.34.41, but you have botocore 1.35.44 which is incompatible.
```

```
In [3]: import pandas as pd
import boto3

In [5]: # Initialize a boto3 S3 client to interact with the S3 service.
s3 = boto3.client('s3')

In [7]: # Initialize an S3 resource to access more detailed resource information and object
s3 = boto3.resource(
    service_name='s3',
    region_name='us-east-2',
    aws_access_key_id='AKIAZYFAWAL3PDSRJWOX',
    aws_secret_access_key='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'
)

In [9]: # Print out bucket names
for bucket in s3.buckets.all():
    print(bucket.name)

objectdatasetai

In [11]: # Print out all available S3 bucket names.

from PIL import Image
from io import BytesIO
```

```
# Create an empty list to store the downloaded images.
image_data=[]

# Iterate through all objects in the 'objectdatasetai' bucket.
for obj in s3.Bucket('objectdatasetai').objects.all():

    # Print object details (optional)
    print(obj)

    # Download the object (image)
    image_obj = obj.get()

    # Read the image content into memory
    image_content = image_obj['Body'].read()

    # Convert the image bytes into a PIL Image object
    image = Image.open(BytesIO(image_content))

    # Append the image to the image_data list
    image_data.append(image)
```





































```
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00010.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00011.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00012.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00013.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00014.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00015.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00016.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00017.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00018.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00019.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00020.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00021.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00022.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00023.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00024.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00025.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00026.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00027.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00028.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00029.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00030.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00031.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00032.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00033.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00034.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00035.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00036.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00037.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00038.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00039.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00040.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00041.jpg')
s3.ObjectSummary(bucket_name='objectdatasetai', key='Dataset/ToothBrush/00042.jpg')
```

In [12]: # Check if any images were successfully downloaded and stored in image\_data.

```
if image_data:
    # Display the first image in the list (for verification).
    image_data[0]
else:
    # Print a message if no images were found in the S3 bucket.
    print("No images found in the bucket.")
```

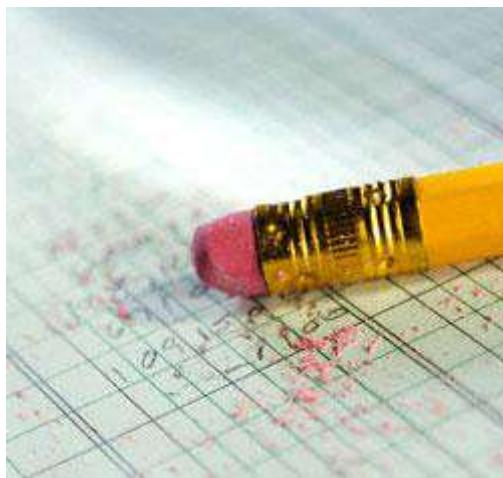
In [13]: # Dataset size

```
len(image_data)
```

Out[13]: 1041

In [14]: image\_data[100]

Out[14]:



In [15]: `image_data[200]`

Out[15]:



In [16]: `image_data[300]`

Out[16]:



In [17]: `image_data[400]`

Out[17]:



In [19]: `image_data[800]`

Out[19]:



In [ ]: