```
!pip install transformers
!pip install astpretty  # For Python code parsing
!pip install tree_sitter  # For parsing multiple programming languages
!pip install gitpython  # To work with Git repositories
```

```
Requirement already satisfied: transformers in
/usr/local/lib/python3.10/dist-packages (4.42.4)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from transformers) (3.15.4)
Requirement already satisfied: huggingface-hub<1.0,>=0.23.2 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.23.5)
Requirement already satisfied: numpy<2.0,>=1.17 in
/usr/local/lib/python3.10/dist-packages (from transformers) (1.26.4)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from transformers) (24.1)
Requirement already satisfied: pyyaml>=5.1 in
/usr/local/lib/python3.10/dist-packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.10/dist-packages (from transformers)
(2024.5.15)
Requirement already satisfied: requests in
/usr/local/lib/python3.10/dist-packages (from transformers) (2.32.3)
Requirement already satisfied: safetensors>=0.4.1 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.4.4)
Requirement already satisfied: tokenizers<0.20,>=0.19 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.19.1)
Requirement already satisfied: tqdm>=4.27 in
/usr/local/lib/python3.10/dist-packages (from transformers) (4.66.5)
Requirement already satisfied: fsspec>=2023.5.0 in
/usr/local/lib/python3.10/dist-packages (from huggingface-
hub<1.0,>=0.23.2->transformers) (2024.6.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.10/dist-packages (from huggingface-
hub<1.0,>=0.23.2->transformers) (4.12.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers)
(3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers)
(3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers)
(2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers)
(2024.7.4)
Collecting astpretty
  Downloading astpretty-3.0.0-py2.py3-none-any.whl.metadata (5.5 kB)
Downloading astpretty-3.0.0-py2.py3-none-any.whl (4.9 kB)
Installing collected packages: astpretty
```

```
Successfully installed astpretty-3.0.0
Collecting tree_sitter
  Downloading tree_sitter-0.22.3-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (10 kB)
Downloading tree_sitter-0.22.3-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (542 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 542.6/542.6 kB 14.7 MB/s eta
0:00:00
etadata (13 kB)
Collecting gitdb<5,>=4.0.1 (from gitpython)
  Downloading gitdb-4.0.11-py3-none-any.whl.metadata (1.2 kB)
Collecting smmap<6,>=3.0.1 (from gitdb<5,>=4.0.1->gitpython)
  Downloading smmap-5.0.1-py3-none-any.whl.metadata (4.3 kB)
Downloading GitPython-3.1.43-py3-none-any.whl (207 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.3/207.3 kB 8.5 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.7/62.7 kB 4.8 MB/s eta
0:00:00
map-5.0.1-py3-none-any.whl (24 kB)
Installing collected packages: smmap, gitdb, gitpython
Successfully installed gitdb-4.0.11 gitpython-3.1.43 smmap-5.0.1
```

```python
from git import Repo

repo_url = 'https://github.com/Shreyas-Patil-01/Git-Docs'
repo_dir = '/content/your-repo'

Repo.clone_from(repo_url, repo_dir)
```

```
<git.repo.base.Repo '/content/your-repo/.git'>
```

```python
import os

# List the files in the repository
repo_dir = '/content/your-repo'
for root, dirs, files in os.walk(repo_dir):
    print(root, dirs, files)
```

```
/content/your-repo ['.git'] ['app.py', 'README.md']
/content/your-repo/.git ['logs', 'hooks', 'refs', 'branches',
'objects', 'info'] ['index', 'HEAD', 'packed-refs', 'description',
'config']
/content/your-repo/.git/logs ['refs'] ['HEAD']
/content/your-repo/.git/logs/refs ['heads', 'remotes'] []
/content/your-repo/.git/logs/refs/heads [] ['main']
/content/your-repo/.git/logs/refs/remotes ['origin'] []
/content/your-repo/.git/logs/refs/remotes/origin [] ['HEAD']
/content/your-repo/.git/hooks [] ['pre-rebase.sample', 'pre-merge-
commit.sample', 'update.sample', 'pre-push.sample', 'applypatch-
msg.sample', 'push-to-checkout.sample', 'pre-receive.sample', 'commit-
```

```
msg.sample', 'pre-commit.sample', 'pre-applypatch.sample', 'prepare-
commit-msg.sample', 'fsmonitor-watchman.sample', 'post-update.sample']
/content/your-repo/.git/refs ['tags', 'heads', 'remotes'] []
/content/your-repo/.git/refs/tags [] []
/content/your-repo/.git/refs/heads [] ['main']
/content/your-repo/.git/refs/remotes ['origin'] []
/content/your-repo/.git/refs/remotes/origin [] ['HEAD']
/content/your-repo/.git/branches [] []
/content/your-repo/.git/objects ['pack', 'info'] []
/content/your-repo/.git/objects/pack [] ['pack-
3f4c1f4b4d805d106989521f574c05a210a35df3.idx', 'pack-
3f4c1f4b4d805d106989521f574c05a210a35df3.pack']
/content/your-repo/.git/objects/info [] []
/content/your-repo/.git/info [] ['exclude']

python_file = os.path.join(repo_dir, 'src', 'app.py')

import ast
import os

def parse_python_file(file_path):
    with open(file_path, 'r') as file:
        tree = ast.parse(file.read())
        for node in ast.walk(tree):
            if isinstance(node, ast.FunctionDef):
                print(f'Found function: {node.name}')
            elif isinstance(node, ast.ClassDef):
                print(f'Found class: {node.name}')

# Example usage
repo_dir = '/content/your-repo'  # Make sure this is set correctly
python_file = os.path.join(repo_dir, 'app.py')  # Updated path
parse_python_file(python_file)

Found class: Calculator
Found function: __init__
Found function: add
Found function: subtract
Found function: multiply
Found function: divide

for root, dirs, files in os.walk(repo_dir):
    for file in files:
        if file.endswith('.py'):
            file_path = os.path.join(root, file)
            print(f'Parsing {file_path}')
            parse_python_file(file_path)

Parsing /content/your-repo/app.py
Found class: Calculator
Found function: __init__
```

```
Found function: add
Found function: subtract
Found function: multiply
Found function: divide
```

```
from transformers import pipeline

# Load a summarization pipeline
summarizer = pipeline('summarization', model='EleutherAI/gpt-neo-
1.3B')

def generate_documentation(file_path):
    with open(file_path, 'r') as file:
        tree = ast.parse(file.read())
        documentation = []
        for node in ast.walk(tree):
            if isinstance(node, ast.FunctionDef):
                func_code = ast.get_source_segment(file.read(), node)
                summary = summarizer(func_code, max_length=30,
min_length=5, do_sample=False)
                documentation.append(f'Function: {node.name}\nSummary:
{summary[0]["summary_text"]}\n')
            elif isinstance(node, ast.ClassDef):
                documentation.append(f'Class: {node.name}\n')
    return "\n".join(documentation)

# Example usage for generating documentation for the file
doc = generate_documentation(python_file)
print(doc)
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/
_token.py:89: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your
settings tab (https://huggingface.co/settings/tokens), set it as
secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to
access public models or datasets.
  warnings.warn(
```


```
{"model_id":"a70f19f2b5f24f938fcda64fabb50759","version_major":2,"vers
ion_minor":0}

{"model_id":"8db5429ebfa340f4be8e0728804f750a","version_major":2,"vers
ion_minor":0}

{"model_id":"44fb8a37b3ca462999f0425aa5d72478","version_major":2,"vers
ion_minor":0}
```

{"model_id":"84146acfd9ef481da8f11b72184335d7","version_major":2,"version_minor":0}

{"model_id":"37946095883b4514b5568daf2f002e25","version_major":2,"version_minor":0}

{"model_id":"5ec31fed5e344612895bd49a063ba786","version_major":2,"version_minor":0}

```
The model 'GPTNeoForCausalLM' is not supported for summarization.
Supported models are ['BartForConditionalGeneration',
'BigBirdPegasusForConditionalGeneration',
'BlenderbotForConditionalGeneration',
'BlenderbotSmallForConditionalGeneration', 'EncoderDecoderModel',
'FSMTForConditionalGeneration',
'GPTSanJapaneseForConditionalGeneration',
'LEDForConditionalGeneration', 'LongT5ForConditionalGeneration',
'M2M100ForConditionalGeneration', 'MarianMTModel',
'MBartForConditionalGeneration', 'MT5ForConditionalGeneration',
'MvpForConditionalGeneration', 'NllbMoeForConditionalGeneration',
'PegasusForConditionalGeneration', 'PegasusXForConditionalGeneration',
'PLBartForConditionalGeneration',
'ProphetNetForConditionalGeneration', 'SeamlessM4TForTextToText',
'SeamlessM4Tv2ForTextToText',
'SwitchTransformersForConditionalGeneration',
'T5ForConditionalGeneration', 'UMT5ForConditionalGeneration',
'XLMProphetNetForConditionalGeneration'].

---------------------------------------------------------------------
-----
IndexError                                Traceback (most recent call
last)
<ipython-input-17-927e005df08a> in <cell line: 20>()
     18
     19 # Example usage for generating documentation for the file
---> 20 doc = generate_documentation(python_file)
     21 print(doc)

<ipython-input-17-927e005df08a> in generate_documentation(file_path)
     10         for node in ast.walk(tree):
     11             if isinstance(node, ast.FunctionDef):
---> 12                 func_code =
ast.get_source_segment(file.read(), node)
     13                 summary = summarizer(func_code, max_length=30,
min_length=5, do_sample=False)
     14                 documentation.append(f'Function: {node.name}\
nSummary: {summary[0]["summary_text"]}\n')

/usr/lib/python3.10/ast.py in get_source_segment(source, node, padded)
    369         padding = ''
```

```
    370
--> 371     first = padding + lines[lineno].encode()
[col_offset:].decode()
    372     last = lines[end_lineno].encode()
[:end_col_offset].decode()
    373     lines = lines[lineno+1:end_lineno]

IndexError: list index out of range
```

```
from transformers import pipeline

# Load a summarization pipeline with a compatible model
summarizer = pipeline('summarization', model='facebook/bart-large-
cnn')
```


```
{"model_id":"c8411cc660aa4f7ca5fde32101083e51","version_major":2,"vers
ion_minor":0}

{"model_id":"f25a2b1000534ceaa24345efefa22abe","version_major":2,"vers
ion_minor":0}

{"model_id":"e025fe4c27be4dc3b85b3a1bb9d47a68","version_major":2,"vers
ion_minor":0}

{"model_id":"52f804cc1eb140aea7da20cc35b693e4","version_major":2,"vers
ion_minor":0}

{"model_id":"14116c5f247244be87b00c82b4bdfeaf","version_major":2,"vers
ion_minor":0}

{"model_id":"5721cf2ecd8a4a9ca25d21ec2b275f82","version_major":2,"vers
ion_minor":0}
```


```
import ast
from transformers import pipeline

# Load a summarization pipeline
summarizer = pipeline('summarization', model='facebook/bart-large-
cnn')

def generate_documentation(file_path):
    with open(file_path, 'r') as file:
        source_code = file.read()
        tree = ast.parse(source_code)
        documentation = []

        for node in ast.walk(tree):
            if isinstance(node, ast.FunctionDef):
                func_code = ast.get_source_segment(source_code, node)
                summary = summarizer(func_code, max_length=30,
min_length=5, do_sample=False)
                documentation.append(f'Function: {node.name}\nSummary:
```

```
{summary[0]["summary_text"]}\n')
            elif isinstance(node, ast.ClassDef):
                documentation.append(f'Class: {node.name}\n')

    return "\n".join(documentation)

# Example usage for generating documentation for the file
python_file = '/content/your-repo/app.py'
doc = generate_documentation(python_file)
print(doc)
```

```
Your max_length is set to 30, but your input_length is only 21. Since
this is a summarization task, where outputs shorter than the input are
typically wanted, you might consider decreasing max_length manually,
e.g. summarizer('...', max_length=10)
Your max_length is set to 30, but your input_length is only 21. Since
this is a summarization task, where outputs shorter than the input are
typically wanted, you might consider decreasing max_length manually,
e.g. summarizer('...', max_length=10)
Your max_length is set to 30, but your input_length is only 21. Since
this is a summarization task, where outputs shorter than the input are
typically wanted, you might consider decreasing max_length manually,
e.g. summarizer('...', max_length=10)

Class: Calculator

Function: __init__
Summary: def __init__(self, num1, num2): num1 = num1 num2 = num2.
self.num

Function: add
Summary: def add(self): return self.num1 + self. num2. def add( self):
return    return self

Function: subtract
Summary: def subtract(self): return self.num1 - self. num2. def
add(self) : return self.: add self

Function: multiply
Summary: def multiply(self): return self.num1 * self. num2. def
multiply( self): return    return self

Function: divide
Summary:  def divide(self): if self.num2 != 0: return "Division by
zero error" else: return self. num
```

```
# Save the documentation in a .txt file
with open('/content/documentation.txt', 'w') as f:
    f.write(doc)
```

```python
# Download the .txt file
from google.colab import files
files.download('/content/documentation.txt')
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.Javascript object>
```