```python
# Appendix C13 - test_content_importer.py

from klassify.src.tables import Document
from klassify.src.content_importer import ContentImporter
import os
import pytest

database_name = "test_klassify"2
if os.path.exists("%s.db" % database_name):
    os.remove("%s.db" % database_name)

DOCUMENT = Document(
    base_path = "/intelligent-machines",
    title = "The Intelligent Machines",
    html = open("test/fixtures/document_page.html", 'r').read())
STRING_PRESENT_IN_BOTH_HEADER_AND_FOOTER = "How government works"
STRING_PRESENT_IN_SCRIPT_TAG = "<![CDATA["
STRING_PRESENT_IN_TITLE = "HM Revenue & Customs"

def setup_module(module):
    global IMPORTER
    IMPORTER = ContentImporter(db_name="test_klassify")
    IMPORTER.DBH.session.add(DOCUMENT)
    IMPORTER.DBH.session.commit()
def teardown_module(module):
    IMPORTER.DBH.session.close()
    IMPORTER.DBH.destroy_db_if_present()

def test_cleaning_methods():
    doc = IMPORTER.DBH.session.query(Document).first()
    page = IMPORTER.parse_page(doc.html)

    assert STRING_PRESENT_IN_BOTH_HEADER_AND_FOOTER in page.text
    assert STRING_PRESENT_IN_SCRIPT_TAG in page.text
    page = IMPORTER.remove_unwanted_tags(page)
    assert STRING_PRESENT_IN_BOTH_HEADER_AND_FOOTER not in page.text
    assert STRING_PRESENT_IN_SCRIPT_TAG not in page.text

    assert STRING_PRESENT_IN_TITLE in page.text
    page = IMPORTER.get_body(page)
    assert STRING_PRESENT_IN_TITLE not in page.text

    page_content = IMPORTER.extract_page_content(page)
    page_content = IMPORTER.remove_non_relevant_content(page_content)
    for phrase in IMPORTER.NON_RELEVANT_PHRASES:
        assert phrase not in page_content

    assert "2016" in page_content
    page_content = IMPORTER.remove_punctuaction_and_numbers(page_content)
    assert "2016" not in page_content

def test_extract_content_single_method():
    doc = IMPORTER.DBH.session.query(Document).first()

    assert STRING_PRESENT_IN_BOTH_HEADER_AND_FOOTER in doc.html
    assert STRING_PRESENT_IN_SCRIPT_TAG in doc.html

    clean_content = IMPORTER.extract_content(doc)

    assert STRING_PRESENT_IN_BOTH_HEADER_AND_FOOTER not in clean_content
    assert STRING_PRESENT_IN_SCRIPT_TAG not in clean_content
    for phrase in IMPORTER.NON_RELEVANT_PHRASES:
        assert phrase not in clean_content

```