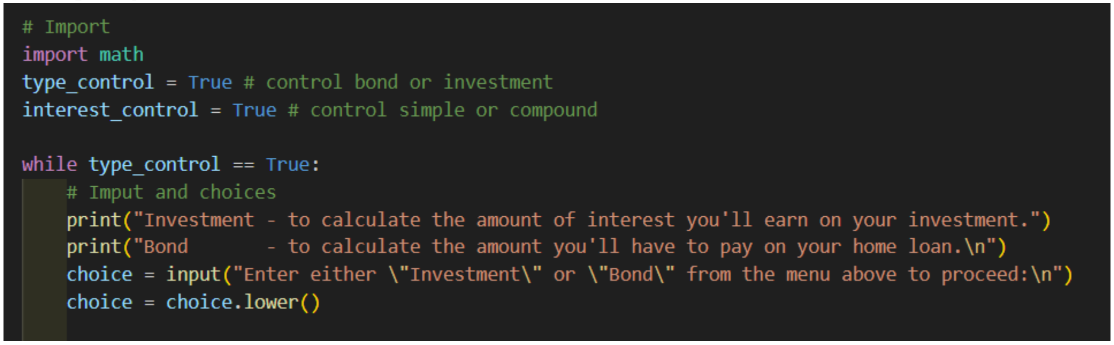
# finalCapstone

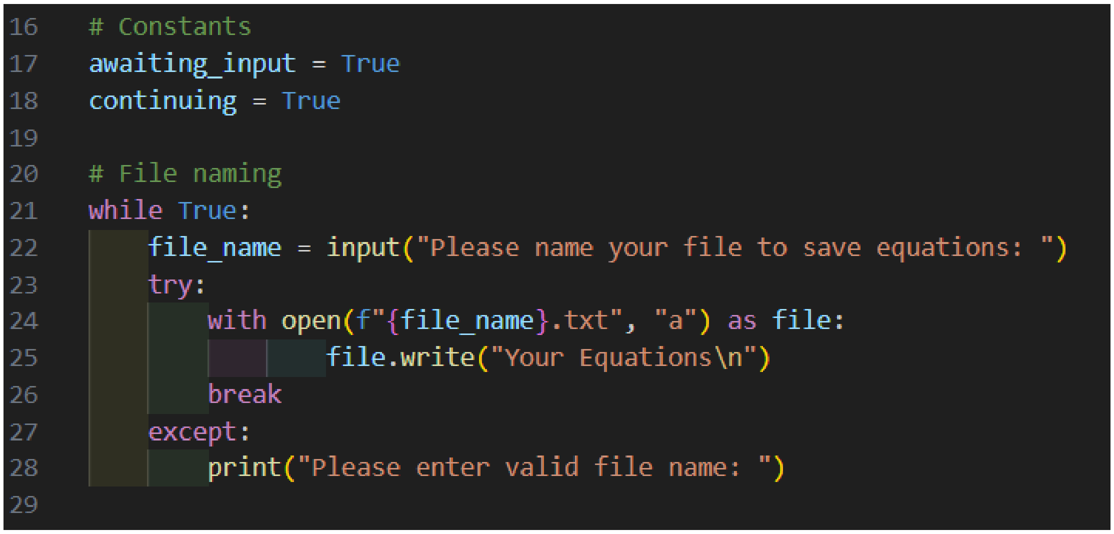
Task 5 First capstone	From line 12.
Task 9 Towards Defensive Programming	From line 17.
Task 21 Semantic Similarity	From line 23.
Task 22 Capstone NLP	From line 35.

# Task 5 First capstone
This was my first chance to express what I had learned so far.
It calculates amount of interest and repayments to be made if you take a loan or mortgage.
All that needs it be imported is math but this is done at the start of the program.



```
# Import
import math
type_control = True # control bond or investment
interest_control = True # control simple or compound

while type_control == True:
    # Imput and choices
    print("Investment - to calculate the amount of interest you'll earn on your investment.")
    print("Bond       - to calculate the amount you'll have to pay on your home loan.\n")
    choice = input("Enter either \"Investment\" or \"Bond\" from the menu above to proceed:\n")
    choice = choice.lower()
```

# Task 9 Towards Defensive Programming
This was a program that taught me to read and write to an external .txt file.
This program is a basic calculator that will add, subtract and divide with the given input from a user.
I went through a lot of trial and error to find all possible bugs and create a stable program.
No additional files are needed.

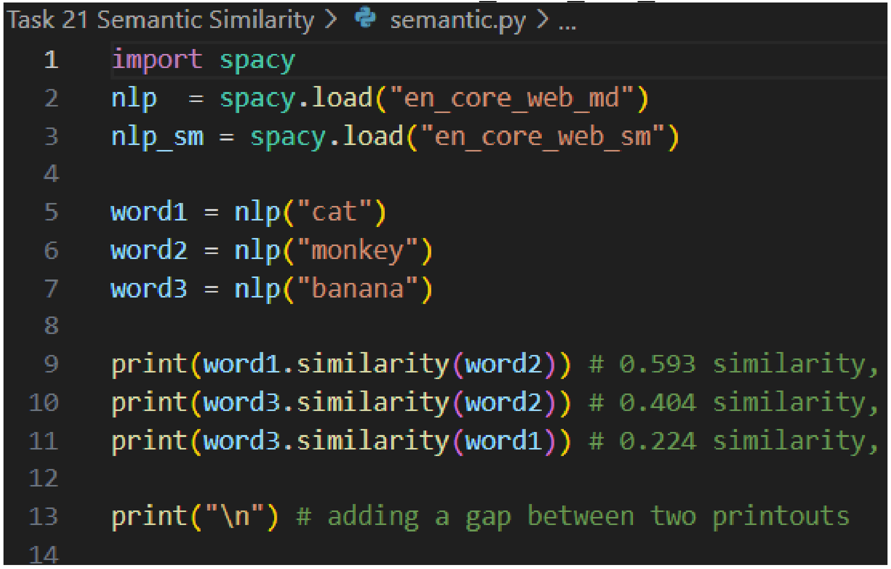


```python
# Constants
awaiting_input = True
continuing = True

# File naming
while True:
    file_name = input("Please name your file to save equations: ")
    try:
        with open(f"{file_name}.txt", "a") as file:
                file.write("Your Equations\n")
        break
    except:
        print("Please enter valid file name: ")

```

# Task 21 Semantic Similarity's

semantic.py
This program taught me how basic NLP works and the importance to use the correct NLP between en_core_web_md and en_core_web_sm.
for this one you need spaCy and the above cores.
use the following in your terminal to install.

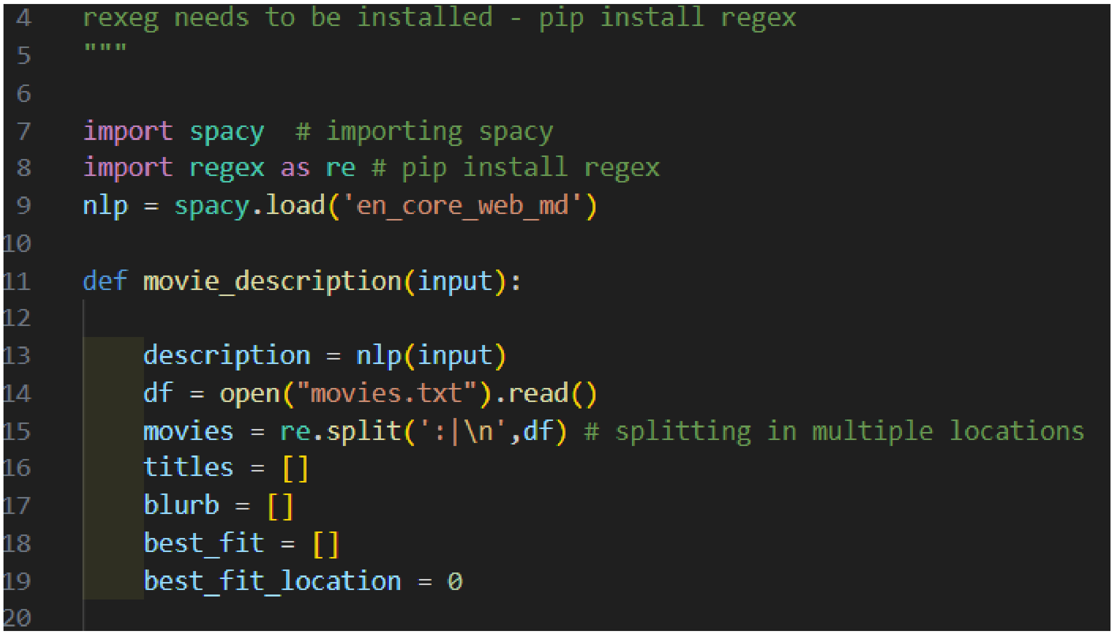
```
pip install spacy
python -m spacy download en_core_web_md
python -m spacy download en_core_web_sm
```



```python
import spacy
nlp  = spacy.load("en_core_web_md")
nlp_sm = spacy.load("en_core_web_sm")

word1 = nlp("cat")
word2 = nlp("monkey")
word3 = nlp("banana")

print(word1.similarity(word2)) # 0.593 similarity,
print(word3.similarity(word2)) # 0.404 similarity,
print(word3.similarity(word1)) # 0.224 similarity,

print("\n") # adding a gap between two printouts

```

watch_next.py
This program taught me to analyse whole sentences for similarity to each other. For this I took an input that was a description of a movie, ran it through NLP and suggested another movie that is similar to the just watched.

As previous this program needs spaCy and the cores but also requires regex to be installed.



```
rexeg needs to be installed - pip install regex
"""

import spacy  # importing spacy
import regex as re # pip install regex
nlp = spacy.load('en_core_web_md')

def movie_description(input):

    description = nlp(input)
    df = open("movies.txt").read()
    movies = re.split(':|\n',df) # splitting in multiple locations
    titles = []
    blurb = []
    best_fit = []
    best_fit_location = 0

```

```
pip install regex
```

# Task 22 Capstone NLP

In this project I demonstrated that I understand the differences and different uses for NLP in the real world.
There are no additions programs required. Just a PDF reader.

Task 1:

a. We could use basic IF statements and Name Entry Recognition NPL to analyse both the email senders address and the content of the email.
NLP could analyse the email address and content, if it is truly odd like 1234abc.whfi@madeup125vw.com then it could just reject the email into spam.
After the NLP we would have an address book that has a list of contact with a category label.
Then the basic IF statements would move them into the corresponding folders.

# Required install

```
pip install spacy
python -m spacy download en_core_web_md
python -m spacy download en_core_web_sm
pip install regex
```

Credits

All of this code is my own work.

Steven Rudley