Silly Sentences

Predict

Take a look at the code below. Read it carefully and try to make a prediction about what might happen when this code is executed.

```python
noun = "Car"
adverb = "gently"
adjective = "loud"
print(f"The {noun} was {adjective} when it {adverb} went to school")
noun = "Zebra"
adverb = "aggressively"
adjective = "giant"
print(f"The {noun} was {adjective} when it {adverb} went to school")
```

Run

Open and **run** the file with this code. A copy can be found here (ncce.io/py-silly-sentences) if needed.

Was your prediction correct? Did anything unexpected happen? Write down your thoughts below:

Investigate

Questions/activities	**Your answers** ▽
In which line is the variable `adjective` **initialised**?	
In which line is the variable `adverb` **first reassigned**?	
When is the variable noun **first used**?	
Is there a **difference** between the code in line 4 and line 8?	
On **line 4**, remove the `f` after print and before the speech mark. Run the code and write down what happens. **! Remember to place the `f` back in when you have finished.**	
What do you think the `f` is used for? This [article](realpython.com/python-f-strings/) (realpython.com/python-f-strings/) might help with your understanding.	
On **line 4**, remove the curly {} brackets that surround noun and run the code again. What happens?	

! Remember to place the {} back in when you have finished.

Why does it not display `The Car Zebra was loud giant when it gently aggressively went to school` when the code is executed?

Modify

Activities	**Hints**▽
Change all values in both occurrences of `noun`, `adjective`, and `adverb` to something different.	You can search for lists of nouns, adjectives, and adverbs on the internet if you need some ideas. Remember to keep the speech marks around the values. These are important.
Create a new variable called `proper_noun` and **initialise** it as London.	Think about the placing of the variable when it is initialised. Where are all of the other variables initialised? Place it with those.
Replace the word `school` with `{proper_noun}` in **both** print statements.	Be careful not to delete any important syntax. Look at the code on line 8 to make sure that you still have the right brackets and speech marks in place.

Make a change to your code that will ensure that the second print statement displays a different proper noun.	Remember that variables can be assigned something different later on in the code. Think carefully about placement.
Add a completely new silly sentence to the bottom of the code. You can use the same variables, but think of a different sentence to write.	Look carefully at the syntax used for the other print statements, and use this to help you find errors in your code.

Make

Create your own silly story using the code from this activity as a guide. There are lots of ‘fill in the blank’ style stories online that you could use for inspiration.

1. Write a silly story
2. Decide which words should be blank
3. Assign variables to the blanks
4. Write the program

You don’t have to use an `adverb, adjective,` or noun. You could use words like `popular_restaurant` or `public_transport`. Be as creative as possible, the sillier the better!

Predict

Take a look at the code below. Read it carefully and try to make a prediction about what might happen when this code is executed.

```python
noun = "Car"
adverb = "gently"
adjective = "loud"
print(f"The {noun} was {adjective} when it {adverb} went to school")
noun = "Zebra"
adverb = "aggressively"
adjective = "giant"
print(f"The {noun} was {adjective} when it {adverb} went to school")
```

Run

Open and **run** the file with this code. A copy can be found here (ncce.io/py-silly-sentences) if needed.

Was your prediction correct? Did anything unexpected happen? Write down your thoughts below:

Investigate

Questions/activities	**Your answers** ▿
In which line is the variable `adjective` **initialised**?	3
In which line is the variable `adverb` **first reassigned**?	6
When is the variable noun **first used**?	4
Is there a **difference** between the code in line 4 and line 8?	No
On **line 4**, remove the `f` after print and before the speech mark. Run the code and write down what happens. **! Remember to place the `f` back in when you have finished.**	It prints the line like this: The {noun} was {adjective} when it {adverb} went to school
What do you think the `f` is used for? This article (realpython.com/python-f-strings/) might help with your understanding.	The f tells Python that you will be using f-strings in this statement.
On **line 4**, remove the curly {} brackets that surround noun and run the code again. What happens? **! Remember to place the { } back in when you have finished.**	It displays the word noun rather than Car

Why does it not display `The Car Zebra was loud giant when it gently aggressively went to school` when the code is executed?	Because a variable only holds one value at a time and they have been reassigned before the last line is executed.

Modify

Activities	**Hints▿**
Change all values in both occurrences of `noun`, `adjective`, and `adverb` to something different.	You can search for lists of nouns, adjectives, and adverbs on the internet if you need some ideas. Remember to keep the speech marks around the values. These are important.
Create a new variable called `proper_noun` and **initialise** it as London.	Think about the placing of the variable when it is initialised. Where are all of the other variables initialised? Place it with those.
Replace the word `school` with `{proper_noun}` in **both** print statements.	Be careful not to delete any important syntax. Look at the code on line 8 to make sure that you still have the right brackets and speech marks in place.

Make a change to your code that will ensure that the second print statement displays a different proper noun.	Remember that variables can be assigned something different later on in the code. Think carefully about placement.
Add a completely new silly sentence to the bottom of the code. You can use the same variables but think of a different sentence to write.	Look carefully at the syntax used for the other print statements and use this to help you find errors in your code.

Make

Create your own silly story using the code from this activity as a guide. There are lots of ‘fill in the blank’ style stories online that you could use for inspiration.

1. Write a silly story
2. Decide which words should be blank
3. Assign variables to the blanks
4. Write the program

You don’t have to use an `adverb, adjective,` or noun. You could use words like `popular_restaurant` or `public_transport`. Be as creative as possible, the sillier the better!