PYTHON SYNTAX FOR THE ASTRONAUTS ESSAY CHALLENGE

Today's goal:
get the computer to read and grade a school essay.

If you are new to Python (or if you simply want to review the exact Python syntax relevant to handling strings, lists, and loops) then this packet is for you!

This packet contains example code which uses:

`print()` **`if`** **`in`** **`for`**

First: make a few variables

On your computer, **open IDLE**.

You should see these symbols:

>>>

These symbols mean you're in the Python shell. Here you can type and experiment.

Type the following things and press ENTER after each one.

```
>>>   magic_word = "really"

>>>   sentence = "Python is really cool."

>>>   magic_word in sentence
```

What word does the computer display after you type all these lines and press ENTER?

*Answer:*_____

The word **`in`** can be used in Python to check whether a particular thing is in another thing.

In our case, we just checked whether or not the data stored in the variable `magic_word` is inside the variable `sentence`.

Fill in the blanks below so that Python checks for the word `"frisbee"` inside `sentence2`.

```
>>>   magic_word = _____________

>>>   sentence2 = "Is there football in space?"

>>>   magic_word in _____________
```

Type the above lines into the shell. What does it print? _____

Try this demo code

From **IDLE,**
go to the File menu and select "*New File*."

→ Name your file **`string_practice.py`**

Save the file to the Desktop.

Start typing the first few lines of the file shown on the next page. Every line or two, stop typing and run your file to make sure there are no mistakes.

How to run a Python file

In IDLE, from the *Run* menu select "RUN MODULE".

string_practice.py

```
# This code demonstrates how Python works with words
# and sentences.


print("This program is about strings and if")
answer = 0
magic_word = "really"
sentence = "Python is really cool."
print()
print(sentence)
print("Python will look for the magic word")

if magic_word in sentence:
    answer = answer + 1
    print("FOUND MAGIC WORD!!!")

print(answer)
```

Line-by-line explanation of **`string_practice.py`**

`print()` is a function.

It lets your Python program display information in the Python shell.

In Python, putting one or more words in quotation marks creates a *string*.

When Python sees a string, it prints it exactly as you wrote it.

```
print("This program is about strings and if")
```

Here, the number zero is stored into the variable `answer`.

```
answer = 0
```

```
magic_word = "really"
```

The value _____ is stored into the variable `magic_word`.

```
sentence = "Python is really cool."
```

The value `"Python is really cool."` is stored into the variable `sentence`

```
print()
```

print() with no parameters will print a blank line.

Look closely: why are there no quotes around the word inside the print() this time?

```
print(sentence)
```

With no quotes, Python assumes the word must be a *variable*.

That is, it must be a piece of information stored somewhere in the computer's memory.

In this line, Python will go look for the value stored inside the variable.

At this point in the program, what value is stored inside `sentence`?

Answer: _____

On this line, Python will print the words exactly as you typed them.

```
print("Python will look for the magic word")
```

The following three lines are unique.

```
if magic_word in sentence:
    answer = answer + 1
    print("FOUND MAGIC WORD!!!")
```

Notice that the first line appears all the way at the left of the screen while the two following lines have a few blank spaces in front of them. These blank spaces, in combination with the colon symbol `:` tell Python that the indented lines belong to the line above them. If the magic word is found inside the sentence, then the indented lines will run. If the magic word is not found inside the sentence, the indented lines will not run.

At the end of the program, Python will print whatever value is contained inside the variable named `answer`.

```
print(answer)
```

Now it's your turn. Write a paragraph which explains what the program does in your own words. You do not need to write about every single line. Write about the overall idea. However, please make sure to explain the line which contains the word **`if`**.

Your explanation:

_____

_____

_____

_____

_____

_____

STOP. *Have a volunteer check your work before you move on.*

The word **`for`** in Python

Now that we've seen how Python handles a few words at a time, let's look at how Python might evaluate a larger number of words. This time we will use one more special word (also called a *reserved word*) from Python: the word **`for`**.

Pretend you need to buy some ingredients at the grocery store in order to bake a cake. Pretend you live in the far future and you and everyone you know has a robot at home to help do chores. A robot is really just a computer with mechanical arms and legs sticking out of it. Telling the robot what to do is very similar to writing a program in Python (in Python, we get to tell the computer what to do).

You ask the robot to go to the grocery store. It agrees!

You have a list of the items you need from the store. You need to explain to the robot (who knows nothing about grocery stores or baking cakes) exactly what to do. Don't worry about Python right now – just think about how you would do this in English.

"Hey robot," you might say. "This is a list of items I need from the grocery store. For each item on the list, I need you to do three specific things."

`for` each item on this list, please:

- Walk down the aisles looking for the item
- If you find it, put it into the shopping cart
- If you don't find it, just move on to the next item

This is EXACTLY how the word **`for`** works in Python. For each item, we ask the computer to do certain specific things. The computer does all of those things **`for`** every item.

Read this page again and make sure you understand it.

If you get stuck, find a volunteer. We're here to help you.

word_or_letters.py

```
# This program demonstrates FOR LOOPS
# Type it and save the file as word_or_letters.py


sentence3 = "Why did the chicken cross the road?"

for x in sentence3:
    print(x)
```

What happens when you run this program?
Explain why.

Your explanation:

_____

_____

_____

_____

_____

_____

STOP. *Have a volunteer check your work before you move on.*

letter_or_words.py

```
# This program demonstrates FOR LOOPS
# Type it and save the file as letter_or_words.py


sentence4 = "I love dinosaurs."

sentence5 = ["I", "love", "dinosaurs."]

print("Python is about to start the first loop....")

for x in sentence4:
    print(x)

print("Now the second loop is starting...")

for x in sentence5:
    print(x)
```

What happens when you run this program?
Explain why.

_____

_____

_____

_____

_____

STOP. *Have a volunteer check your work before you move on.*