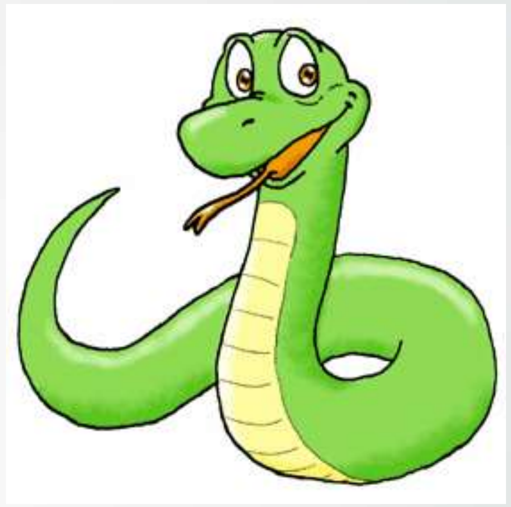
Learn Application Programming

Introduction to Python

Why learn Python3?



- Python is a programming language, that is…
  - ***general-purpose***
  - versatile
  - popular
- It is great as a **first** language concise and easy to read.
- It is also a good language to know as it can be used for
  - web development
  - software development and
  - scientific applications.

ONLINE COURSES

Learn Python | Codecademy

https://www.codecademy.com/learn/python

Course Outcomes. This **course** is a great introduction to both fundamental programming concepts and the **Python** programming language. By the end, you'll be ...

Learn Python - Free Interactive Python Tutorial

https://www.learnpython.org/

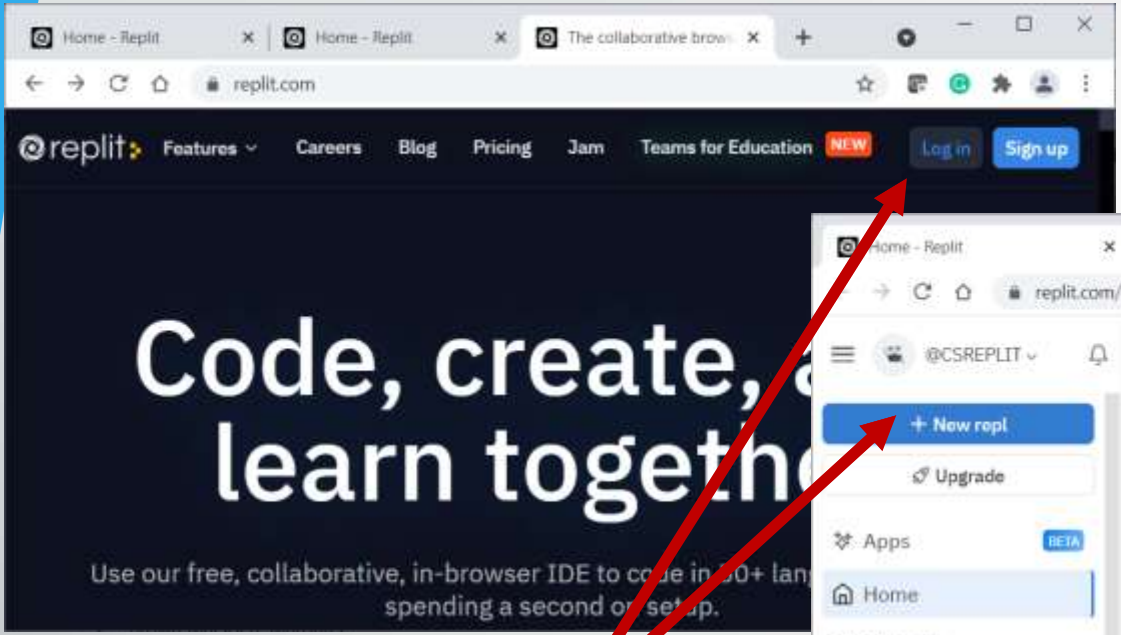
LearnPython.org is a **free** interactive **Python** tutorial for people who want to learn ... machine learning, and more; f.read **Python** Tutorials and References **course** ...

Introduction to Python for Beginners | Udemy

https://www.udemy.com/introduction-to-python-for-beginners/

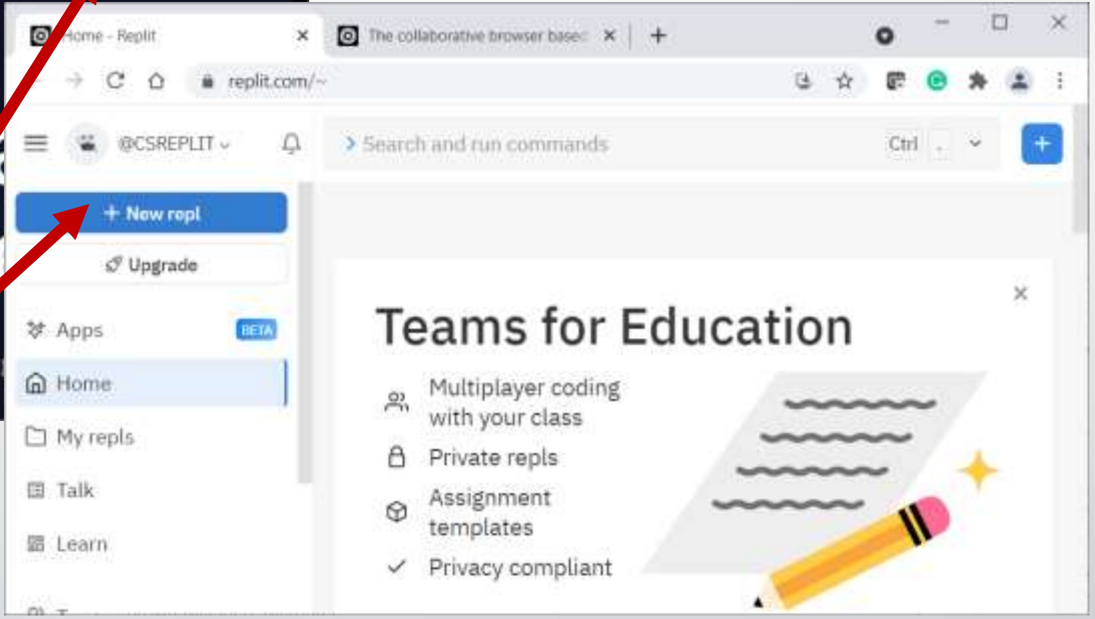
Learn Python Programming on the Mac or PC with "**Python for Beginners**" **Python** training course.

Using repl.it



Code, create, a learn togeth

Use our free, collaborative, in-browser IDE to code in 50+ lan spending a second on setup.

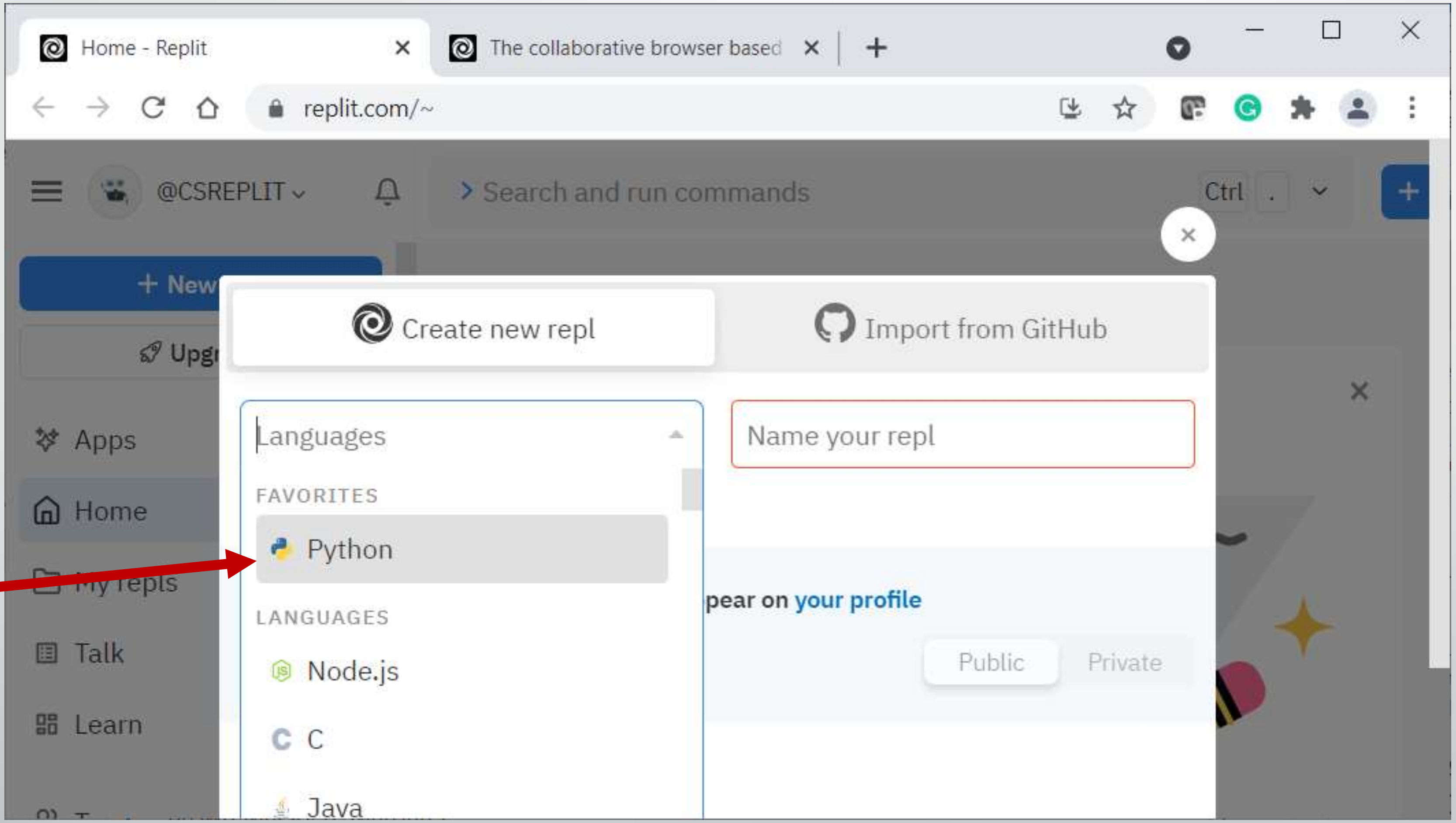


Teams for Education

- Multiplayer coding with your class
- Private repls
- Assignment templates
- Privacy compliant

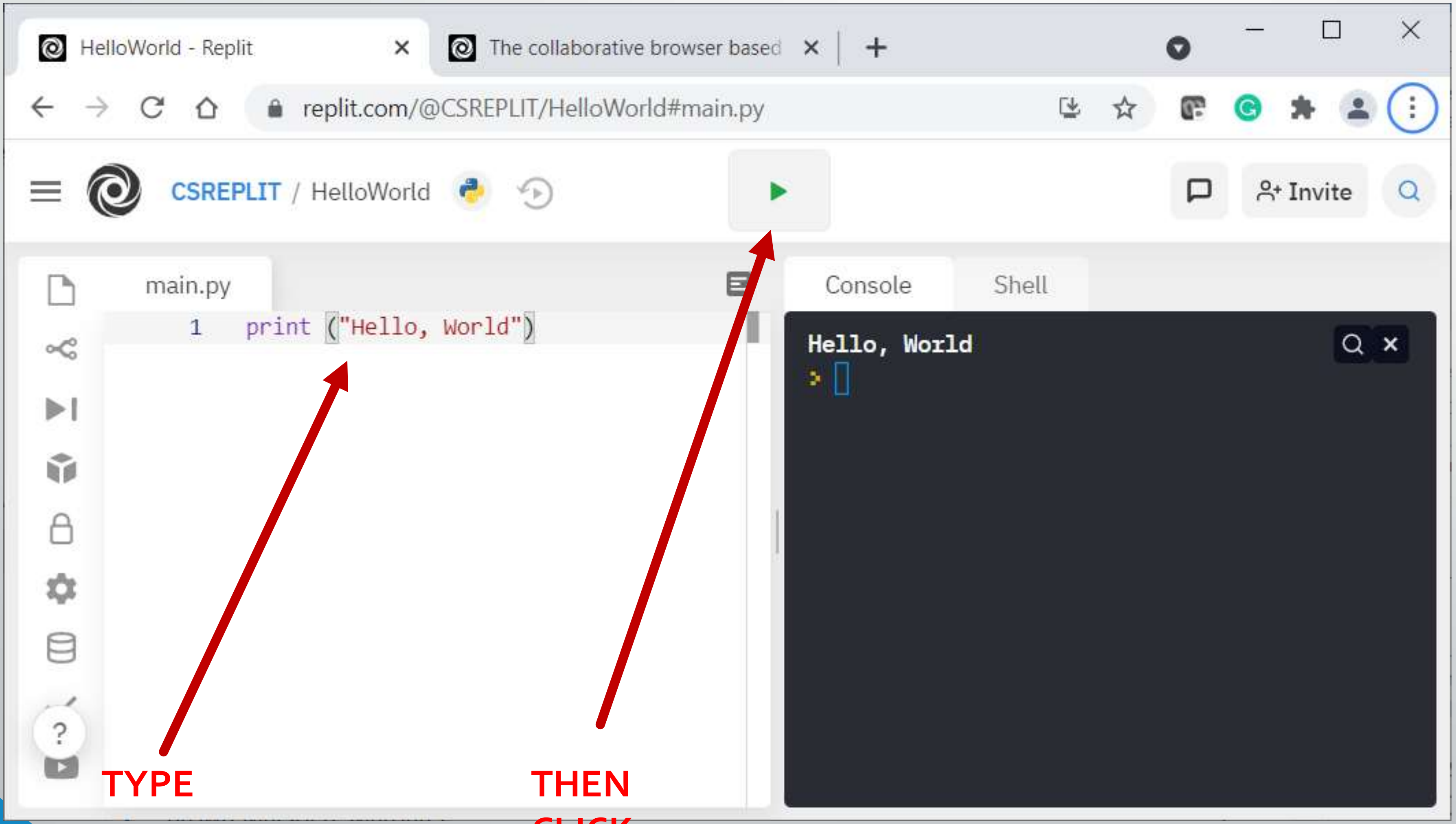
CLICK HERE

Using repl.it



CLICK HERE

Welcome to Python using Repl.it


HelloWorld - Replit
The collaborative browser based
replit.com/@CSREPLIT/HelloWorld#main.py
CSREPLIT / HelloWorld
Invite
main.py
Console
Shell

```
print ("Hello, World")
```

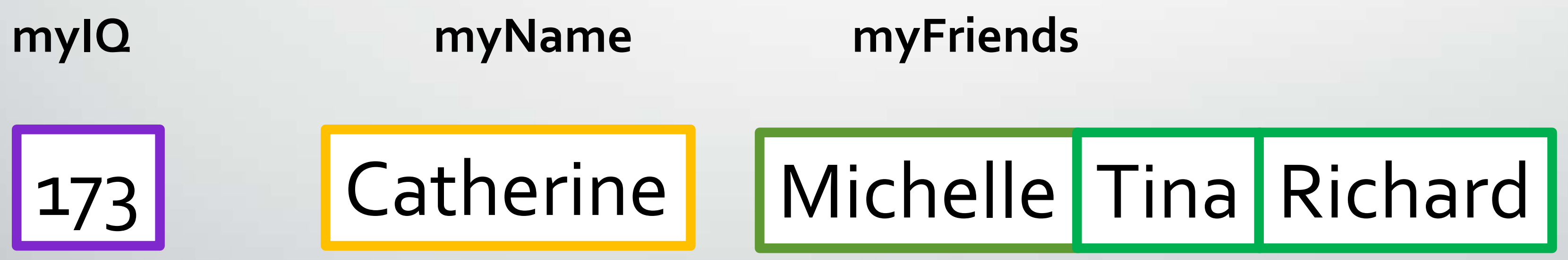
```
Hello, World
```

TYPE

THEN CLICK

Programming Basics

- Must be able to represent data in a computer program
- Do this with *variables* – *memory locations whose values vary (or change)*
  - Our focus on Numbers, Strings and Lists



- Variables have a memory *location*, a *type*, a *name* and a *value*

Variables in Python

- Assignment is one way to put values in variables

```
myIQ = 173
myName = "Catherine"
myFriends = ["Michelle", "Tina", "Richard"]
```

- Output them with print

```
print (myIQ)
print ("Hello")
print (myName)
print (myFriends)
```

Operators for Numbers

- Numbers +, -, *, /   and more

```
a = 5
b = 2
total = a + b
print (total)
```

Operations for Strings

- Strings – concatenation (+) , split, str and more

```
myName = “Catherine"
greeting = "Hello"
phrase = greeting + " " + myName
print (phrase)
```

Operations for Strings

- Strings – concatenation (+) , split, str and more

```
nouns = "boy girl cat pepperoni peppers triangles"
noun_list = nouns.split()
print (noun_list)
```

$\longleftarrow$ by the way noun_list is a *list*

List Operations

- Lists –single element access **`[ ]`**, length or rather **`len( )`**, **`append( )`** and more

```
myFriends = [“Michelle”, “Tina”, “Richard”]
print (myFriends[0])
print (myFriends[1])
print (myFriends[2])
myFriends.append(“Sheldon")
print (myFriends[3])
```

Controlling the Order of Operations

- Python statements are executed in the order they appear *unless* YOU tell them to execute in some *other order*
- *Several* ways to do this
    - By choosing ONE of several alternative groups of statements
    - By repeating a group of statements
    - or by going off and executing another group of statements and then coming back
- We will focus on the first two ways

Alternative Statements

```
myIQ = 173
if (myIQ > 160):
   s = myName + " is a genius."
   print (s)
```

FOR Loops

- There are two types of loops, the **`for`** loop and the **`while`** loop, we will use the **`for`** loop

```
nouns = "boy girl cat pepperoni peppers triangles"
noun_list = nouns.split()
print (noun_list)
for noun in noun_list:
    print (noun)
```

Inputting values

- To make a program interesting, so it will do different things every time it runs, you need to enter values into the program
- To open a file, use `fileHandle = open(string_value)`
- We can readline a string from the file

```
title = readline ("Enter a title for your story: ")
 print ("                  " + title  + "\n\n")
```

- We can also enter input from a keyboard

```
s = input (“Enter name”)
```

or

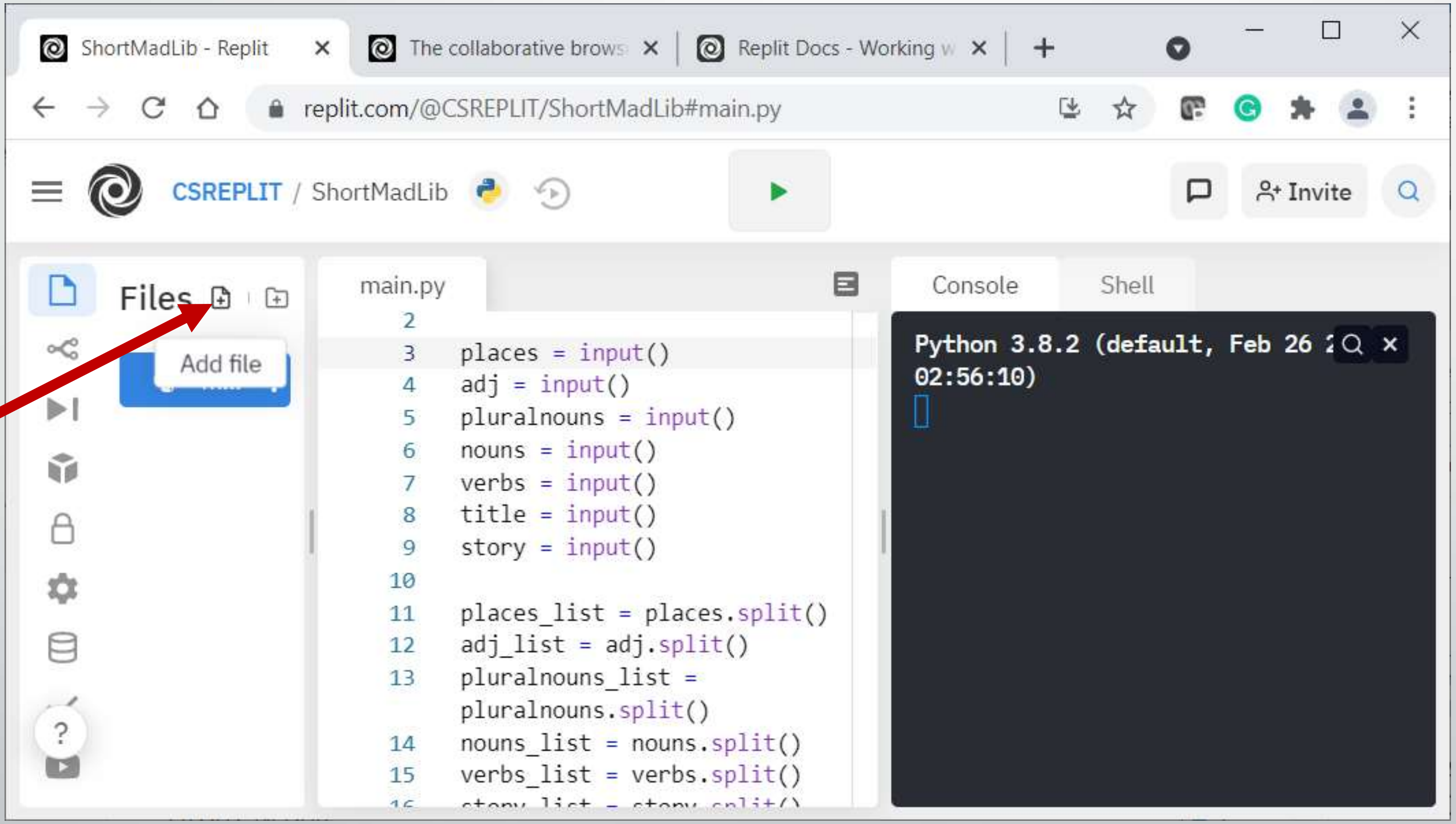
```
s = sys.stdin.readline(“Enter name”)
```

ready to Code MadLibs?

- PROBLEM: This program will print out a funny complete story at the end. The program will get a series of readlines for nouns, verbs, adjectives, etc., and substitute that data into a pre-made story template to produce the final story.
- Concepts to keep in mind that we will use: variables, strings, concatenation, split, lists, ifs, for loops and readline

- Erase everything in repl.it and start fresh. Hit CTRL-A and DEL to clear screen…

Enter data



```
places = input()
adj = input()
pluralnouns = input()
nouns = input()
verbs = input()
title = input()
story = input()

places_list = places.split()
adj_list = adj.split()
pluralnouns_list =
pluralnouns.split()
nouns_list = nouns.split()
verbs_list = verbs.split()
```

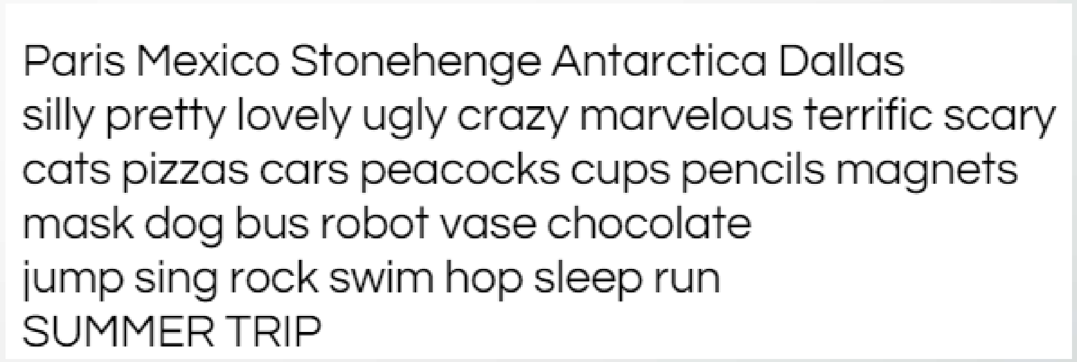
```
Python 3.8.2 (default, Feb 26 2
02:56:10)
```

CLICK HERE

Enter data

- Type in box
  1. Type in place names
  2. Type in adjectives
  3. Type in some plural nouns
  4. Type in some singular nouns
  5. Type in some action verbs
  6. The title of your story

     SUMMER TRIP
  7. Story on the next slide



Paris Mexico Stonehenge Antarctica Dallas
silly pretty lovely ugly crazy marvelous terrific scary
cats pizzas cars peacocks cups pencils magnets
mask dog bus robot vase chocolate
jump sing rock swim hop sleep run
SUMMER TRIP

Enter title and story

- Now type all the below ***WITHOUT HITTING THE ENTER KEY***

Last summer, my mom and dad took me on a trip to PLACE . The weather there is very ADJECTIVE ! There, they have many PLURALNOUN , and they make ADJECTIVE PLURALNOUN there. Many people also go there to VERB or see PLURALNOUN . The people that live there love to eat PLURALNOUN and are very proud of their big NOUN . They also like to VERB in the sun and VERB in the lake! It was a really ADJECTIVE trip!

- Make sure you have a ***space before all punctuation as shown***.
- Now hit the enter key.

Ready to CODE?

f.readline values in your program

- input all these values from a file

1.	Open file	`f = open("words.txt")`
2.	f.readline values and store them in **`places`**	`places = f.readline()`
3.	f.readline values and store them in **`adj`**	*do 2-5 similarly*
4.	f.readline values and store them in **`pluralnouns`**	
5.	f.readline values and store them in **`nouns`**	
6.	f.readline values and store them in **`verbs`**	
7.	f.readline story title and store it in title	`title = f.readline()`
8.	f.readline story and store it	`story = f.read()`

CODE!

- Split all the strings (except for title) into lists

```
places_list = places.split( )
```

do similarly for all lists

```
story_list = story.split( )
```

Code!

- print a couple of new lines before the title, then some spaces before the title, then the title and then a couple of blank lines after

```
print (" \n\n                    " + title + "\n\n")
```

- print the story, as it is, first ... so you can make sure it f.readline correctly

```
print (story)
```

CODE!

- Set a new string variable to the *empty string* (a blank slate so to speak)

```
uniqueStory = ""
```

- We are going to append to it over and over in a loop

```
for word in story_list:
   uniqueStory = uniqueStory + " " + word
```

- Print the title and the changed story

```
print ("\n\n                         " + title + "\n\n")
print (uniqueStory)
```

Run your program!

- Run your program!
- WAIT! What happened?!? We forgot to substitute in the words in the story.
- AND! We want ***the program*** to choose words to substitute into story.
  - At top of program, type in the line

```
from random import choice
```

  - `choice(a_list)` will give us one random item from a list

FIX the CODE!

- if word is a *keyword* like PLURALNOUN or ADJECTIVE, substitute for it instead.

```
for word in story_list:
    subword = word
    if word == "PLACE":
        subword = choice(places_list)
    if word == "ADJECTIVE":
        subword = choice (adj_list)
```

NOTE: INDENTATION VERY IMPORTANT!

do similarly for PLURALNOUN, NOUN , and VERB

```
    uniqueStory = uniqueStory + " " + subword
```

RUN THE CODE!

- Run your program!
- Run it a second time!
- Run it a third time!

Learn Python 3!

```
from random import choice
f = open(“words.txt”)

places = f.readline()
adj = f.readline()
pluralnouns = f.readline()
nouns = f.readline()
verbs = f.readline()
title = f.readline()
story = f.read()

places_list = places.split()
adj_list = adj.split()
pluralnouns_list = pluralnouns.split()
nouns_list = nouns.split()
verbs_list = verbs.split()
story_list = story.split()

print (" \n\n            " + title + "\n\n")
print (story)
```

```
uniqueStory = ""

for word in story_list:
   subword = word
   if word == "PLACE":
     subword = choice(places_list)
   if word == "ADJECTIVE":
     subword = choice (adj_list)
   if word == "PLURALNOUN":
     subword = choice (pluralnouns_list)
   if word == "NOUN":
     subword = choice (nouns_list)
   if word == "VERB":
     subword = choice (verbs_list)

   uniqueStory = uniqueStory + " " + subword

print (" \n\n            " + title + "\n\n")
print (uniqueStory)
```