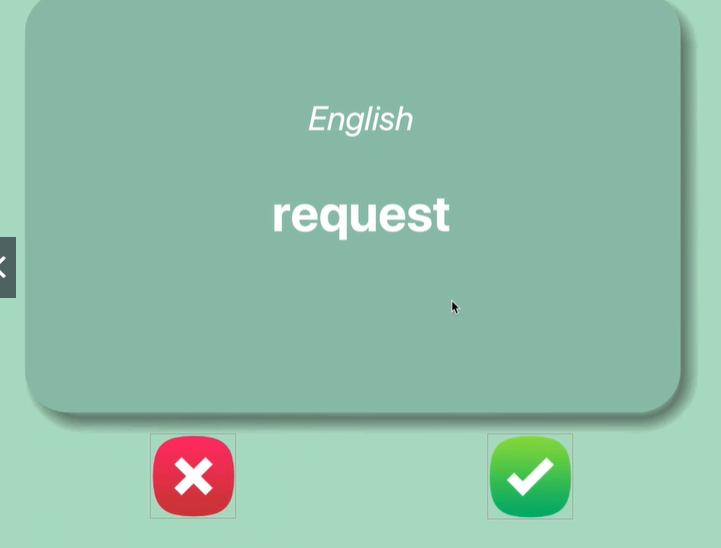
Day 31 Flash Card App project

<https://www.udemy.com/course/100-days-of-code/learn/lecture/20944498#questions>





Starting project has top 100

Step 1 - Create the User Interface (UI) with Tkinter

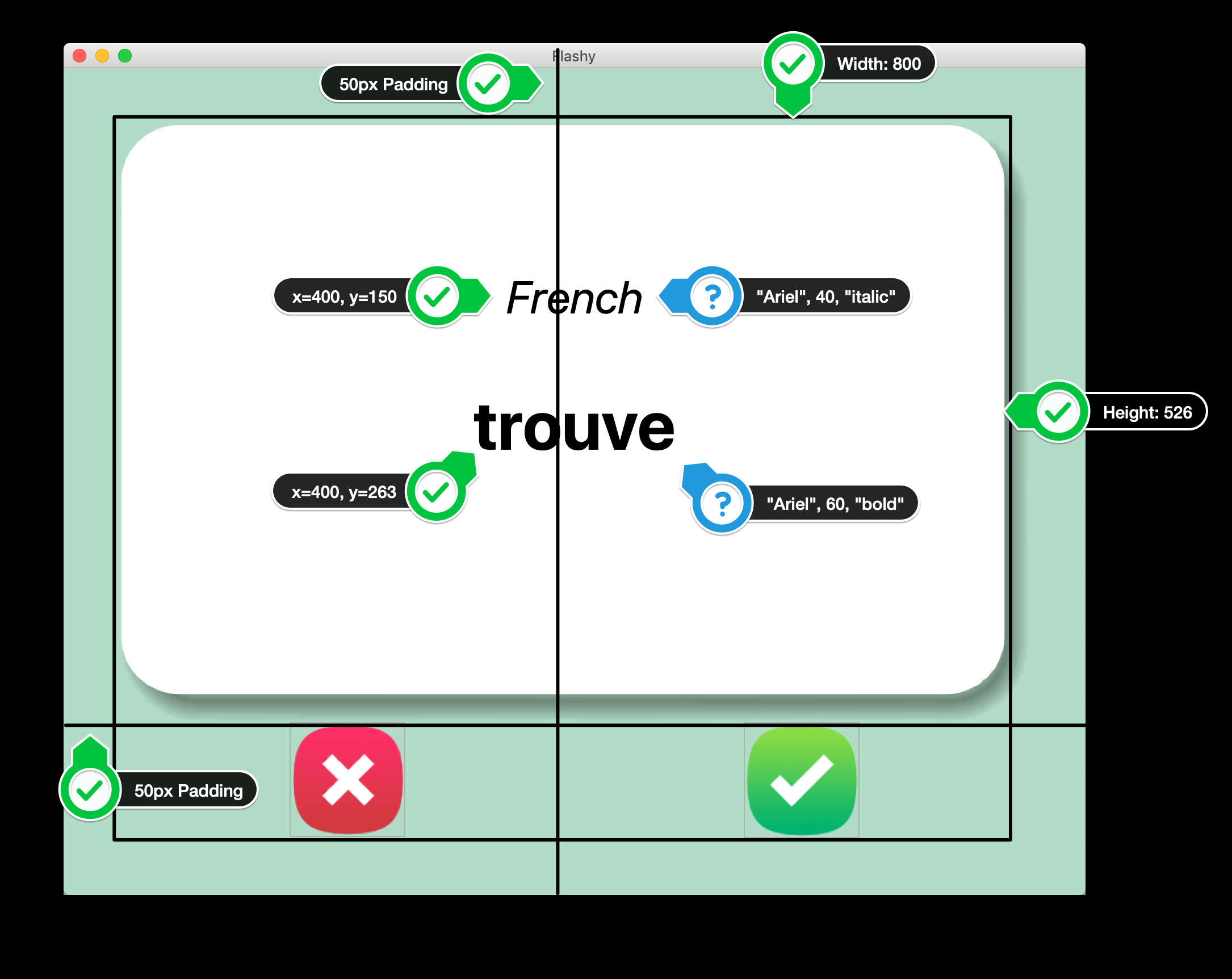
1. Download the starting files from the course resources.

2. Use the images in the image folder, to create the following user interface. The ❌ and ✅ are buttons. You can add images to buttons like this:

1. my\_image = PhotoImage(file="path/to/image\_file.png")
2. button = Button(image=my\_image, highlightthickness=0)



3. Here are some hints for the fonts, measurements and positioning.



HINTS:

1. You will need a 2 X 2 grid, with the flash card taking up 2 columns.

2. The flash card is a [Canvas](https://effbot.org/tkinterbook/canvas.htm) with 1 image and 2 pieces of text.

3. The image is card\_front.png, created from the [PhotoImage](https://effbot.org/tkinterbook/photoimage.htm) class. Be careful about the full image path as the image is inside the image folder.

Step 2 - Create New Flash Cards

1. Read the data from the **french\_words.csv** file in the **data** folder.

2. Pick a **random** French word/translation and put the word into the flashcard. Every time you **press** the ❌ or ✅ buttons, it should generate a**new random word**to display. e.g.



HINT:

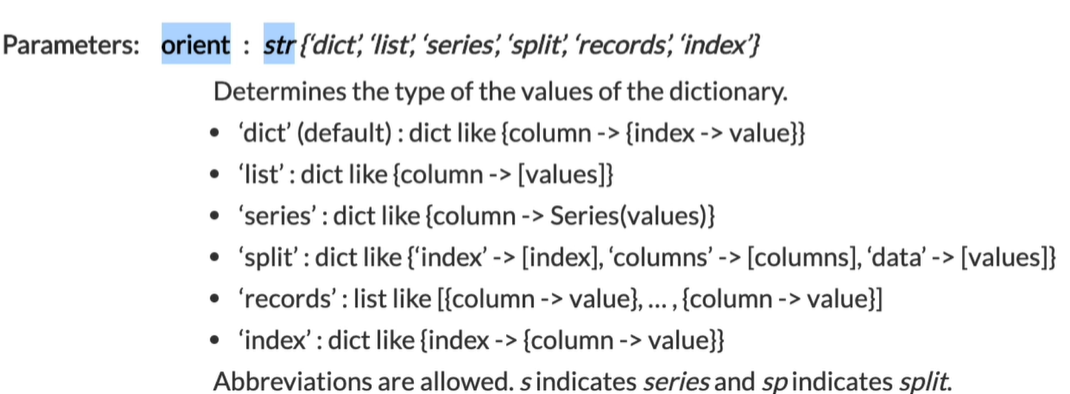
1. You'll need to use pandas to access the CSV file and generate a data frame. To get all the words/translation rows out as a list of dictionaries e.g. [{french\_word: english\_word}, {french\_word2: english\_word2}, {french\_word3: english\_word3}]

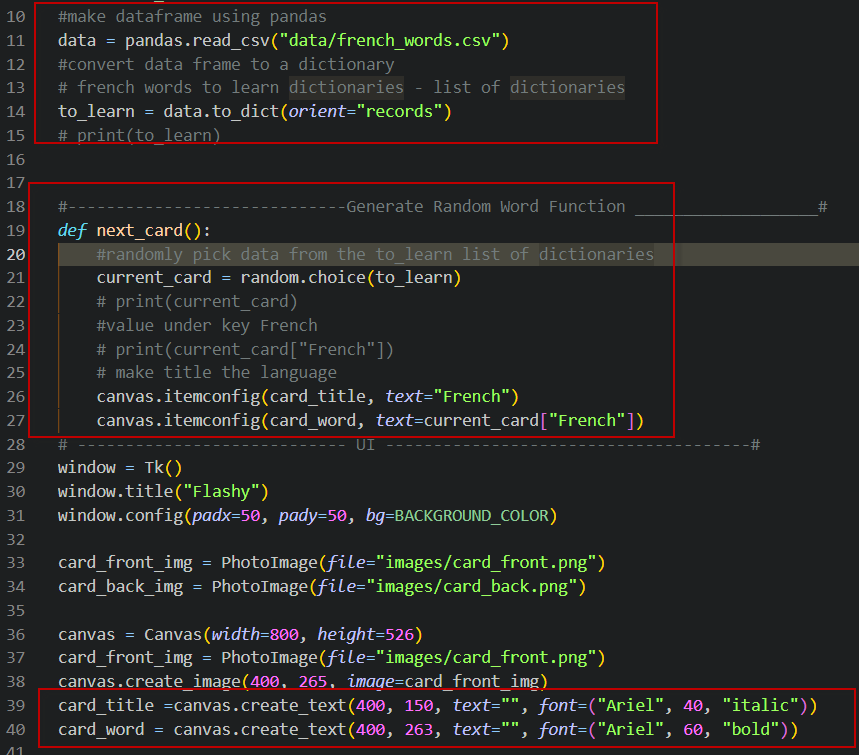
You could use:

DataFrame.to\_dict(orient="records")

Documentation: <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.to_dict.html>

Use orient to set the columns values as a list





Step 3 - Flip the Cards!

1. After a delay of **3s** (3000ms), the card should flip and display the **English translation** for the current word.

2. The card image should change to the **card\_back.png** and the text colour should change to **white**. The **title** of the card should change to "*English*" from "*French*".

e.g.

HINTS:

1. To change the canvas image, you'll need a reference to the image, like what you have with the text created in the canvas. Then you can set the image attribute using itemconfig(). e.g.

1. new\_image = PhotoImage(file="new\_image.png")
2. old\_image = PhotoImage(file="old\_image.png")
3. canvas\_image = canvas.create\_image(300, 300, image=old\_image)
4. #To change the image:
5. canvas.itemconfig(canvas\_image, image=new\_image)

IMPORTANT: PhotoImage objects should not be created inside a function. Otherwise, it will not work.

2. To change the color of the text in a canvas element, use the fill parameter. e.g. <https://stackoverflow.com/questions/41030973/how-can-i-change-the-color-of-text-in-tkinter>

3. Remember in the mainloop() you should **not** create additional delayed loops e.g. with time.sleep() but instead use window.after() e.g. <https://effbot.org/tkinterbook/widget.htm#Tkinter.Widget.after-method>

import random  
from tkinter import \*  
import pandas  
   
  
BACKGROUND\_COLOR = "#B1DDC6"  
  
data = pandas.read\_csv("data/french\_words.csv")  
to\_learn = data.to\_dict(orient="records")  
# set current\_card dictionary to a global , start with empty  
current\_card ={}  
  
#-----------------------------Generate Random Word Function \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_#  
def next\_card():  
 #grab the global var current\_card and flip timer  
 global current\_card**,** flip\_timer  
 #cancel the old flip timer if user clicked next before it ran out  
 window.after\_cancel(flip\_timer)  
 current\_card = random.choice(to\_learn)  
 canvas.itemconfig(card\_title**,** text="French"**,** fill="black")  
 canvas.itemconfig(card\_word**,** text=current\_card["French"])  
 canvas.itemconfig(card\_background**,**image=card\_front\_img)  
 # reset / set new flip\_timer for this card  
 flip\_timer= window.after(**3000,** func=flip\_card)  
# -------------------------Flip Card Function -----------------------------------#  
def flip\_card():  
 canvas.itemconfig(card\_title**,** text="English"**,** fill="white")  
 canvas.itemconfig(card\_word**,** text=current\_card["English"])  
 #change to back\_card image  
 canvas.itemconfig(card\_background**,** image=card\_back\_img)  
  
# ---------------------------- UI --------------------------------------#  
window = Tk()  
window.title("Flashy")  
window.config(padx=**50,** pady=**50,** bg=BACKGROUND\_COLOR)  
# set timer, set to var make it a global  
flip\_timer =window.after(**3000,** func=flip\_card)  
  
card\_front\_img = PhotoImage(file="images/card\_front.png")  
card\_back\_img = PhotoImage(file="images/card\_back.png")  
  
canvas = Canvas(width=**800,** height=**526**)  
card\_front\_img = PhotoImage(file="images/card\_front.png")  
card\_background = canvas.create\_image(**400, 265,** image=card\_front\_img)  
card\_title =canvas.create\_text(**400, 150,** text=""**,** font=("Ariel"**, 40,** "italic"))  
card\_word = canvas.create\_text(**400, 263,** text=""**,** font=("Ariel"**, 60,** "bold"))  
  
canvas.config(bg=BACKGROUND\_COLOR**,** highlightthickness=**0**)  
canvas.grid(row=**0,** column=**0,** columnspan=**3**)  
  
# Buttons  
  
wrong\_img = PhotoImage(file="images/wrong.png")  
wrong\_button = Button(image=wrong\_img**,** highlightthickness=**0,**command=next\_card)  
wrong\_button.grid(row=**2,** column=**0**)  
  
right\_img = PhotoImage(file="images/right.png")  
right\_button = Button(image=right\_img**,** highlightthickness=**0,** command=next\_card)  
right\_button.grid(row=**2,** column=**2**)  
  
#call next card for frist time  
next\_card()  
  
window.mainloop()

Step 4 - Save Your Progress

1. When the user presses on the ✅ button, it means that they know the current word on the flashcard and that word should be removed from the list of words that might come up.

e.g. If french\_words.csv had only 3 records:

1. chaque,each
2. parlé,speak
3. arrivé,come

After the user has seen parlé,speak  it should be **removed** from the list of words that can come up again.

2. The updated data should be saved to a new file called **words\_to\_learn.csv**

e.g. words\_to\_learn.csv

1. chaque,each
2. arrivé,come

3. The next time the program is run, it should check if there is a words\_to\_learn.csv file. If it exists, the program should use those words to put on the flashcards. If the words\_to\_learn.csv does not exist (i.e., the first time the program is run), then it should use the words in the french\_words.csv

We want our flashcard program to only test us on things we don't know. So if the user presses the ✅ button, that means the current card should not come up again.

HINTS:

1. The remove() method can remove elements from a list. e.g. <https://www.w3schools.com/python/ref_list_remove.asp>

2. You can create a new csv file from a dictionary using DataFrame.to\_csv() e.g. <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.to_csv.html>

3. If you don't want to create an index for the new csv, you can set the index parameter to False. e.g.

data.to\_csv("filename.csv", index=False)

import random

from tkinter import \*

import pandas

BACKGROUND\_COLOR = "#B1DDC6"

to\_learn = {}

current\_card ={}

try:

    data = pandas.read\_csv("data/words\_to\_learn.csv")

except FileNotFoundError:

    original\_data = pandas.read\_csv("data/french\_words.csv")

    to\_learn = original\_data.to\_dict(*orient*="records")

else:

    to\_learn = data.to\_dict(*orient*="records")

#-----------------------------Generate Random Word Function \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_#

*def* next\_card():

    global current\_card, flip\_timer

    window.after\_cancel(flip\_timer)

    current\_card = random.choice(to\_learn)

    canvas.itemconfig(card\_title, *text*="French", *fill*="black")

    canvas.itemconfig(card\_word, *text*=current\_card["French"])

    canvas.itemconfig(card\_background,*image*=card\_front\_img)

    flip\_timer= window.after(3000, *func*=flip\_card)

# -------------------------Flip Card Function -----------------------------------#

*def* flip\_card():

    canvas.itemconfig(card\_title, *text*="English", *fill*="white")

    canvas.itemconfig(card\_word, *text*=current\_card["English"])

    canvas.itemconfig(card\_background, *image*=card\_back\_img)

#---------------------------Function to remove known word from list \_\_\_\_\_\_\_\_\_\_\_\_\_#

*def* is\_known():

    #remove this dictionary from the to learn list

    to\_learn.remove(current\_card)

    #save the to learn list to a new csv

    data = pandas.DataFrame(to\_learn)

    #telling pandas not to repeatedly add index column to csv

    data.to\_csv("data/words\_to\_learn.csv", *index*=False)

    next\_card()

# ---------------------------- UI --------------------------------------#

window = Tk()

window.title("Flashy")

window.config(*padx*=50, *pady*=50, *bg*=BACKGROUND\_COLOR)

flip\_timer =window.after(3000, *func*=flip\_card)

card\_front\_img = PhotoImage(*file*="images/card\_front.png")

card\_back\_img = PhotoImage(*file*="images/card\_back.png")

canvas = Canvas(*width*=800, *height*=526)

card\_front\_img = PhotoImage(*file*="images/card\_front.png")

card\_background = canvas.create\_image(400, 265, *image*=card\_front\_img)

card\_title =canvas.create\_text(400, 150, *text*="", *font*=("Ariel", 40, "italic"))

card\_word = canvas.create\_text(400, 263, *text*="", *font*=("Ariel", 60, "bold"))

canvas.config(*bg*=BACKGROUND\_COLOR, *highlightthickness*=0)

canvas.grid(*row*=0, *column*=0, *columnspan*=3)

# Buttons

wrong\_img = PhotoImage(*file*="images/wrong.png")

wrong\_button = Button(*image*=wrong\_img, *highlightthickness*=0,*command*=next\_card)

wrong\_button.grid(*row*=2, *column*=0)

right\_img = PhotoImage(*file*="images/right.png")

# user knows this word so remove it from the list to learn

right\_button = Button(*image*=right\_img, *highlightthickness*=0, *command*=is\_known)

right\_button.grid(*row*=2, *column*=2)

#call next card for frist time

next\_card()

window.mainloop()