# Practice #5

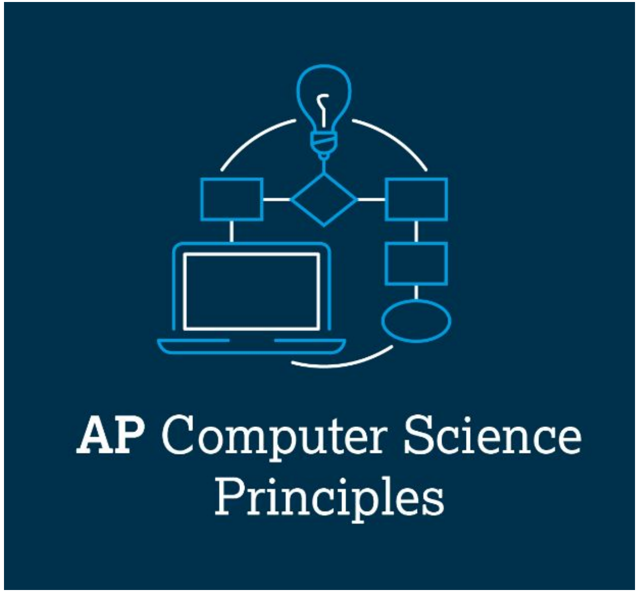
Create Performance Task

FIRIA LABS

# AP CSP Create Performance Task

Part of the AP Exam is to create a program that meets specific requirements:

- Creates a list
- Uses a list in a meaningful way
- Has a function with a parameter
  - Parameter is used in an if statement
- Function has:
  - If statement
  - Loop

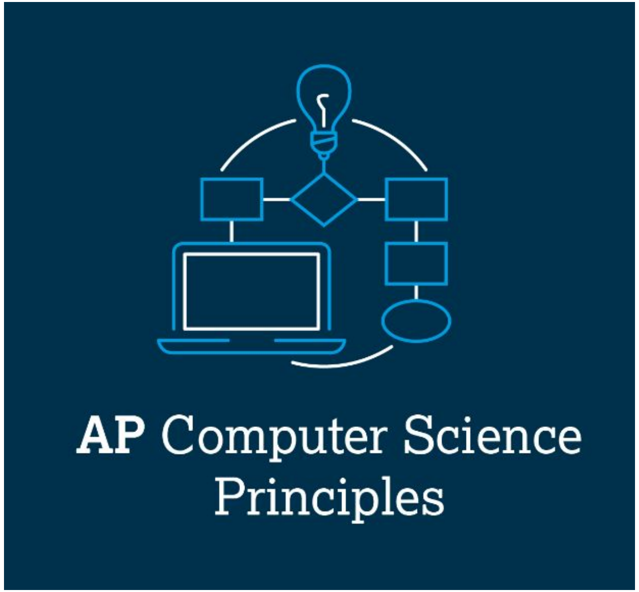


# AP CSP Create Performance Task

For this project, you will:

- Start with two missions you already completed
- Merge them together

So when you are finished, you will have one program, built from two, that meets all the requirements for the Create PT

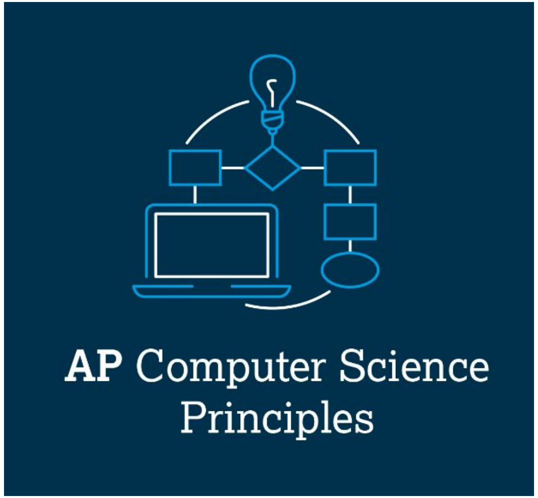


# AP CSP Create Performance Task

During the semester, you completed many missions.

- Look them over
- Which were your favorites?
- Does at least one of your favorites use a list?

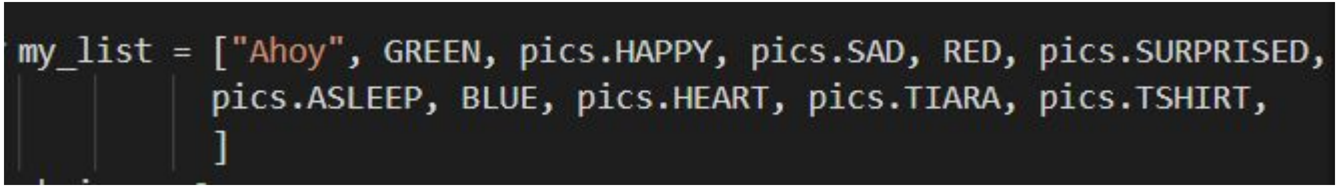
For this practice, you will pick two missions, one with a list, and merge them into one program to meet the requirements.



# Create PT Requirements

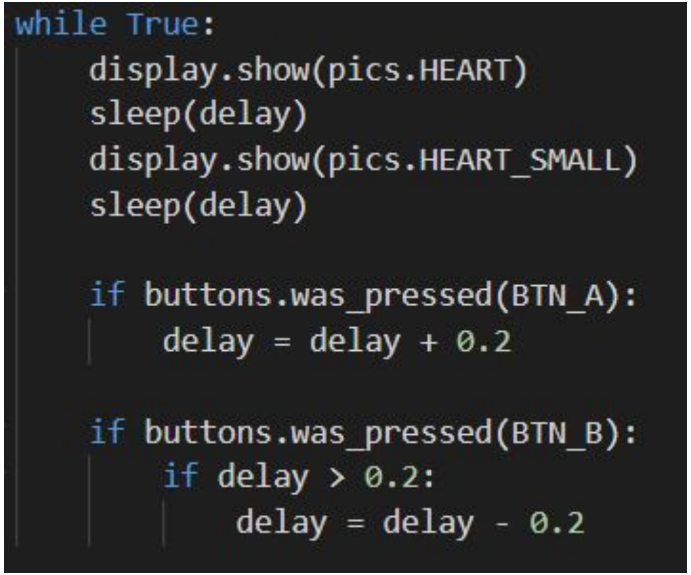
For this practice, you will use Mission 6 - **Heart2**, and Mission 7 - **Billboard**

- Both missions use a loop and if statements (iteration and selection)
- Mission 7 also uses a list (Create PT requirement)



```
my_list = ["Ahoy", GREEN, pics.HAPPY, pics.SAD, RED, pics.SURPRISED,
           pics.ASLEEP, BLUE, pics.HEART, pics.TIARA, pics.TSHIRT,
           ]
```

# Step #1

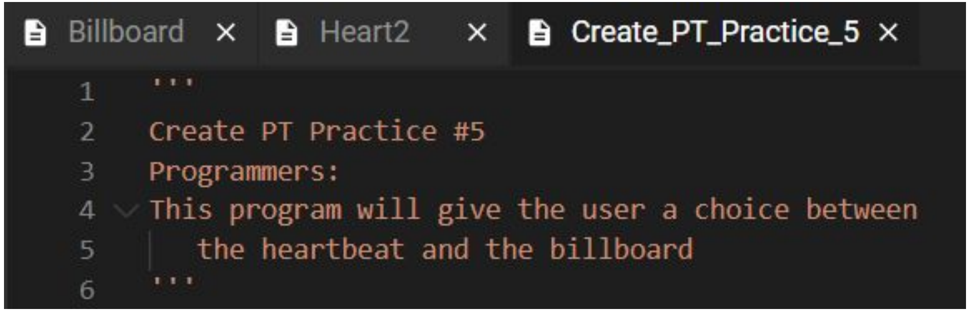


```python
while True:
    display.show(pics.HEART)
    sleep(delay)
    display.show(pics.HEART_SMALL)
    sleep(delay)

    if buttons.was_pressed(BTN_A):
        delay = delay + 0.2

    if buttons.was_pressed(BTN_B):
        if delay > 0.2:
            delay = delay - 0.2
```

## Open your project from Mission 6 - “Heart2”

- Run the code and make sure it works properly
- You may notice the program will throw an error if BTN_B is pressed too many times because delay will go below 0
- Fix the code by adding an if statement so the code will be error-free

# Step #1



```
'''
Create PT Practice #5
Programmers:
This program will give the user a choice between
    the heartbeat and the billboard
'''
```

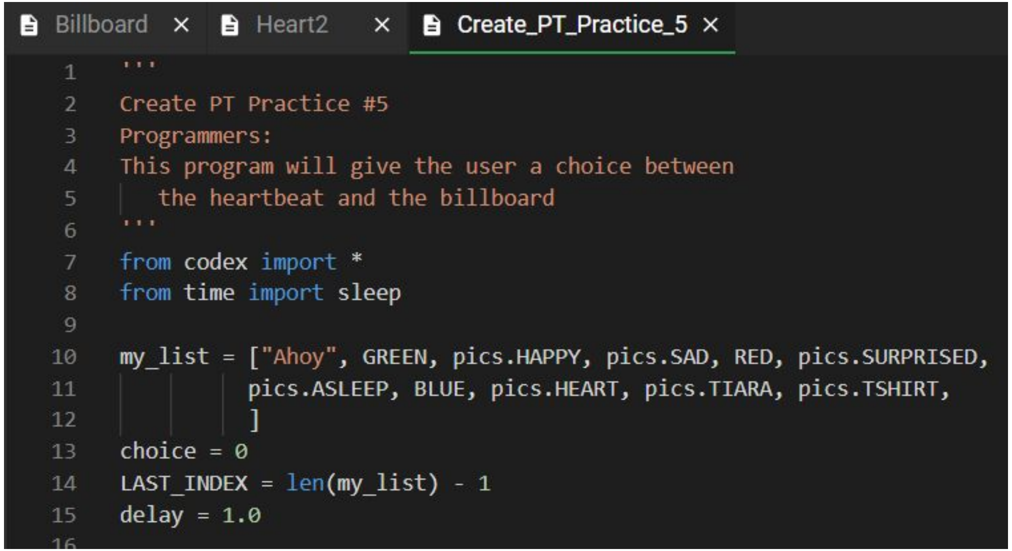
## Open your project from Mission 7 - “Billboard”

- Run the code and make sure it works properly

## Start a new project and call it “Create_PT_Practice5”

- Add a comment block at the top with the program name and the programmers names

# Step #2



```
'''
Create PT Practice #5
Programmers:
This program will give the user a choice between
    the heartbeat and the billboard
'''
from codex import *
from time import sleep

my_list = ["Ahoy", GREEN, pics.HAPPY, pics.SAD, RED, pics.SURPRISED,
          pics.ASLEEP, BLUE, pics.HEART, pics.TIARA, pics.TSHIRT,
          ]
choice = 0
LAST_INDEX = len(my_list) - 1
delay = 1.0
```

## Copy and paste the code

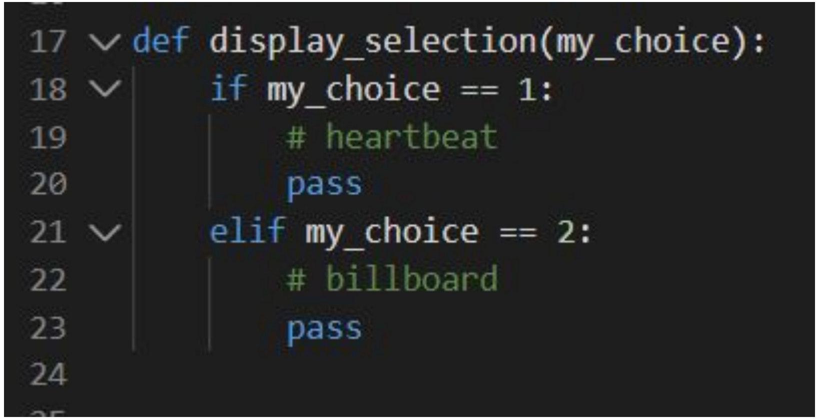
- Go to **Heart2** and copy and paste the code from the top of the program into **Practice5**
- Go to **Billboard** and copy and paste the code from the top of the program into **Practice5**

# Step #2

## Function with parameter

- A Create PT requirement is to have a function with a parameter used in an if statement.
- In **Practice2** your function with parameter was used to select which list to display.
- In **Practice3** your function with parameter was used for easy or hard, and assigned a value to delay.
- In **Practice5**, your function with parameter will be similar.

# Step #2



```
def display_selection(my_choice):
    if my_choice == 1:
        # heartbeat
        pass
    elif my_choice == 2:
        # billboard
        pass

```

## Function with parameter

You will use a parameter to show which choice the user made: heartbeat or billboard.

- Create a function with a parameter and an if statement for the two choices.

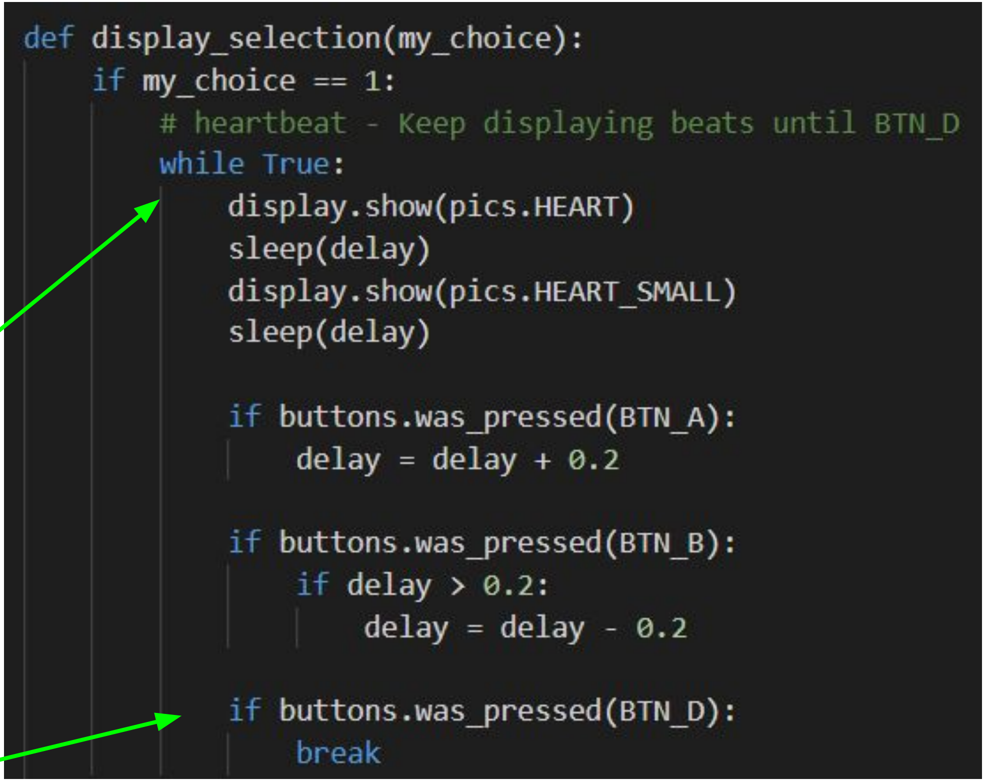
- The function has a parameter
- The parameter is used in a selection
- The “pass” is a placeholder for code – you will delete it when you paste the code

# Step #2

## Add the code from “Heart2”

Delete “**pass**” for **my_choice == 1**

- Copy the code from **Heart2** and paste it in the if statement.
- You will need to increase the indenting – do so carefully!
- Add code to stop the heart if BTN_D is pressed



```
def display_selection(my_choice):
    if my_choice == 1:
        # heartbeat - Keep displaying beats until BTN_D
        while True:
            display.show(pics.HEART)
            sleep(delay)
            display.show(pics.HEART_SMALL)
            sleep(delay)

            if buttons.was_pressed(BTN_A):
                delay = delay + 0.2

            if buttons.was_pressed(BTN_B):
                if delay > 0.2:
                    delay = delay - 0.2

            if buttons.was_pressed(BTN_D):
                break
```

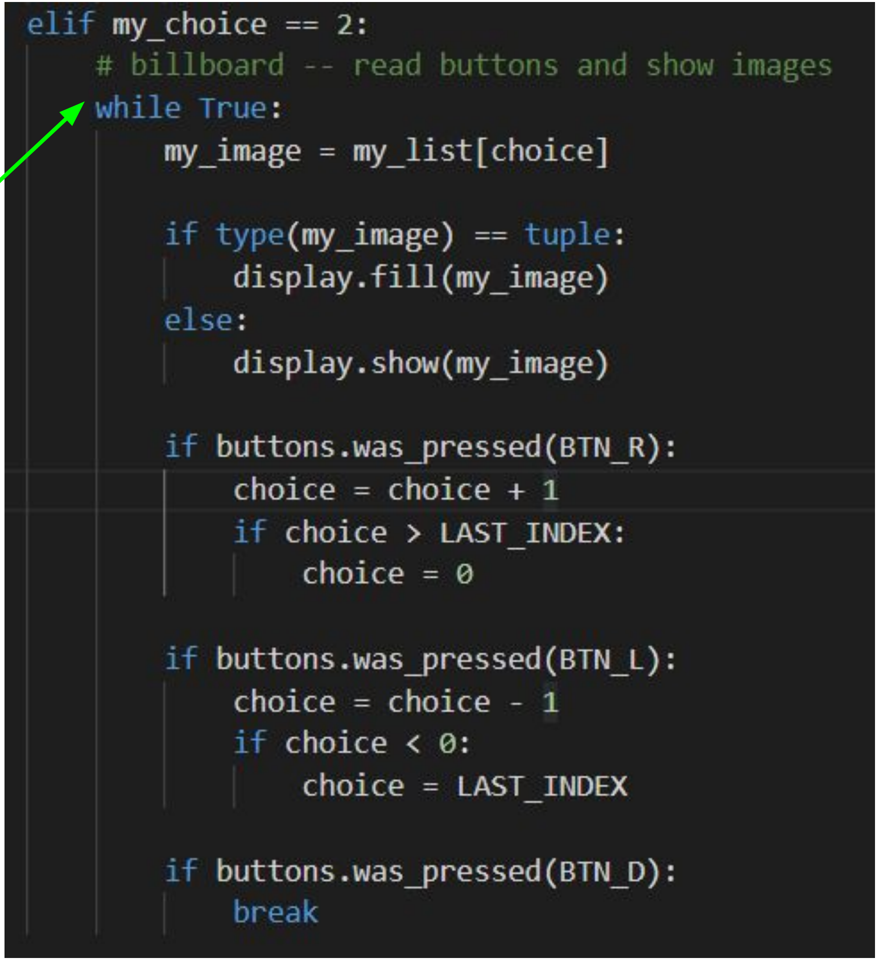
FIRIA LABS

# Step #2

## Add the code from “Billboard”

Delete “**pass**” for **my_choice == 2**

- Copy the code from **Billboard** and paste it in the if statement.
- You will need to increase the indenting – do so carefully!
- Add code to stop the heart if BTN_D is pressed



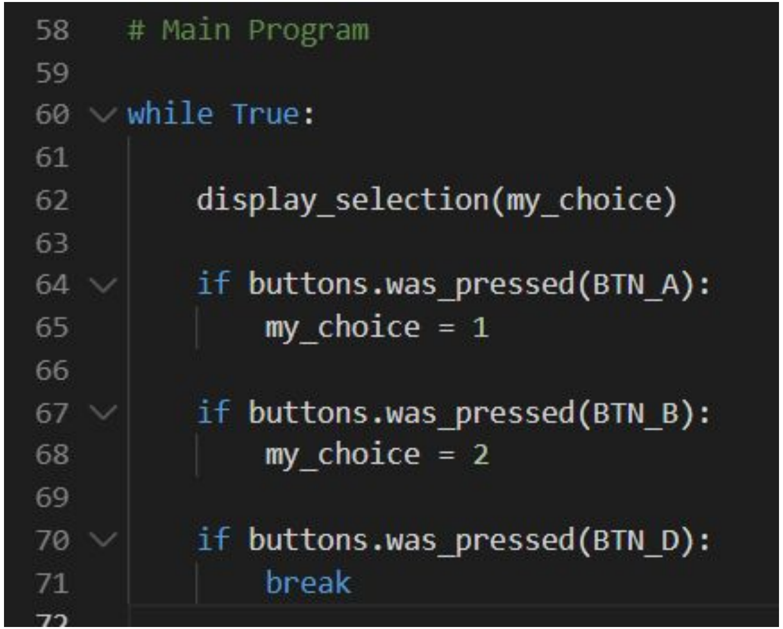
```
elif my_choice == 2:
    # billboard -- read buttons and show images
    while True:
        my_image = my_list[choice]

        if type(my_image) == tuple:
            display.fill(my_image)
        else:
            display.show(my_image)

        if buttons.was_pressed(BTN_R):
            choice = choice + 1
            if choice > LAST_INDEX:
                choice = 0

        if buttons.was_pressed(BTN_L):
            choice = choice - 1
            if choice < 0:
                choice = LAST_INDEX

        if buttons.was_pressed(BTN_D):
            break
```

# Step #3

## Time for the main program

You have most of the program done already, just by copying and pasting code

- Create a main program that:
    - Calls the function
    - Checks if BTN_A is pressed
    - Checks if BTN_B is pressed
    - Breaks if BTN_D is pressed



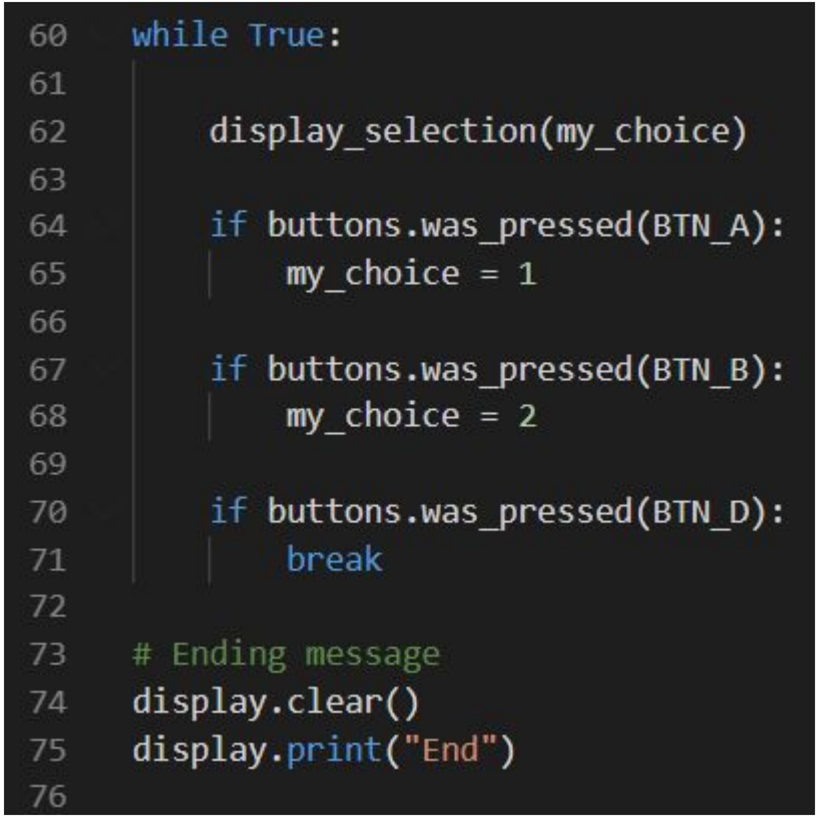
```
# Main Program

while True:

    display_selection(my_choice)

    if buttons.was_pressed(BTN_A):
        my_choice = 1

    if buttons.was_pressed(BTN_B):
        my_choice = 2

    if buttons.was_pressed(BTN_D):
        break
```

*Be careful with indenting – make sure you are against the edge and not indented inside the function.*

# Step #3

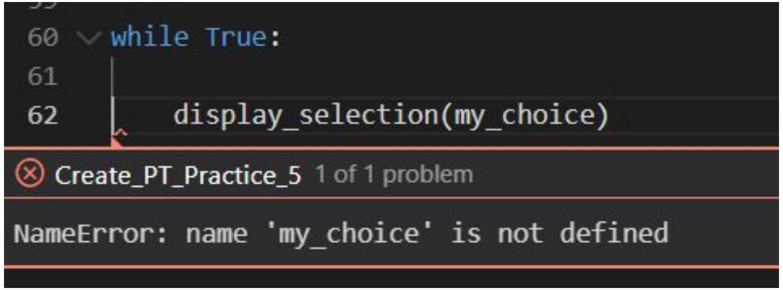
## Main program

- Add a short ending AFTER the while loop ends
- Be careful with indenting – the print statements should NOT be indented inside the while loop



```
while True:

    display_selection(my_choice)

    if buttons.was_pressed(BTN_A):
        my_choice = 1

    if buttons.was_pressed(BTN_B):
        my_choice = 2

    if buttons.was_pressed(BTN_D):
        break

# Ending message
display.clear()
display.print("End")

```

# Step #4



```
while True:

    display_selection(my_choice)
```

Create_PT_Practice_5 1 of 1 problem

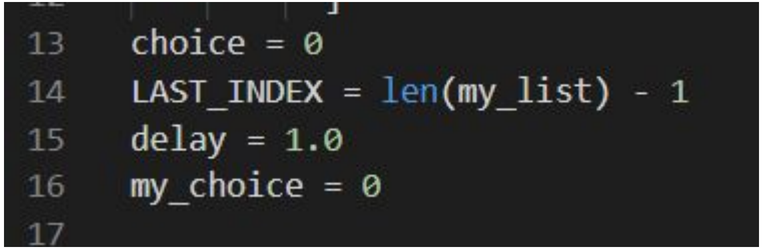
```
NameError: name 'my_choice' is not defined
```

## Fix all the little things that need fixing

You have almost all the code done now. But, it won’t work yet. Run the code, and fix all the little things that come up, one by one.

- Run the code
- What is the error message?
- What does it mean?

# Step #4



```
choice = 0
LAST_INDEX = len(my_list) - 1
delay = 1.0
my_choice = 0

```

*Why my_choice = 0? Why not my_choice = 1 or my_choice = 2? We want the user to make a selection, so start the value at something other than 1 or 2.*

## Fix all the little things that need fixing

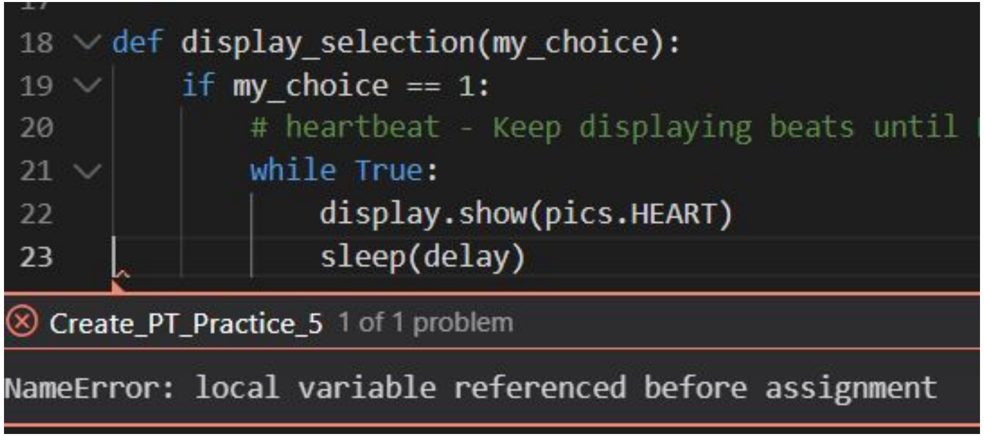
You should have seen this error many times throughout the semester

- You have a global variable that has not been defined
- Go to the top of your code where the list and other variables are
- Add a definition for my_choice

# Step #4

## Fix all the little things that need fixing

- Run the code again
- Press A for the heartbeat
- What is the error message?
- What does it mean?



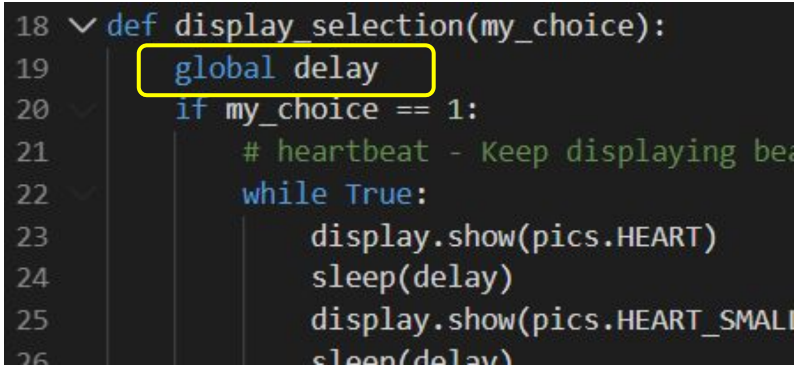
```
def display_selection(my_choice):
    if my_choice == 1:
        # heartbeat - Keep displaying beats until
        while True:
            display.show(pics.HEART)
            sleep(delay)
```

Create_PT_Practice_5 1 of 1 problem

```
NameError: local variable referenced before assignment
```

# Step #4

## Fix all the little things that need fixing



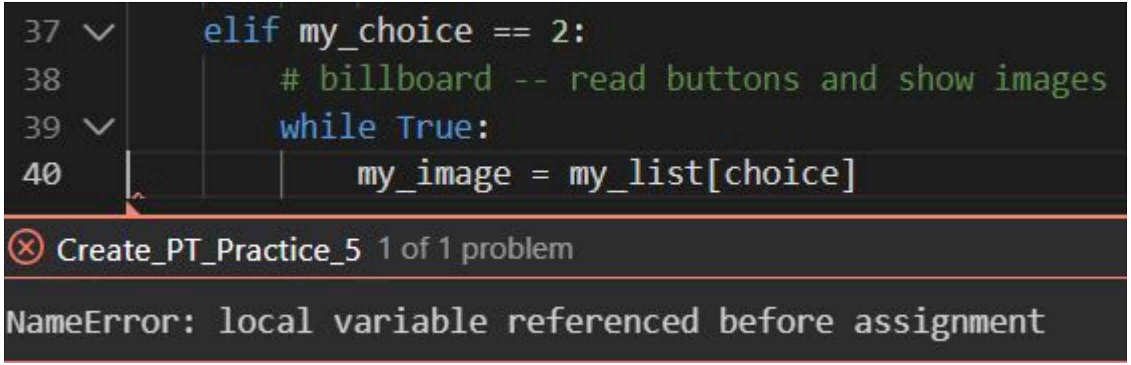
```
def display_selection(my_choice):
    global delay
    if my_choice == 1:
        # heartbeat - Keep displaying bea
        while True:
            display.show(pics.HEART)
            sleep(delay)
            display.show(pics.HEART_SMALL
            sleep(delay)
```

- The code stops at **delay** in the function
- **delay** is a global variable, but the computer thinks it is local – notice the error message
- You could make it a parameter, but the easiest way to fix the error is to add a global declaration in the function

# Step #4

## Fix all the little things that need fixing

- Run the code again
- Press B for the billboard
- What is the error message?
- What does it mean?

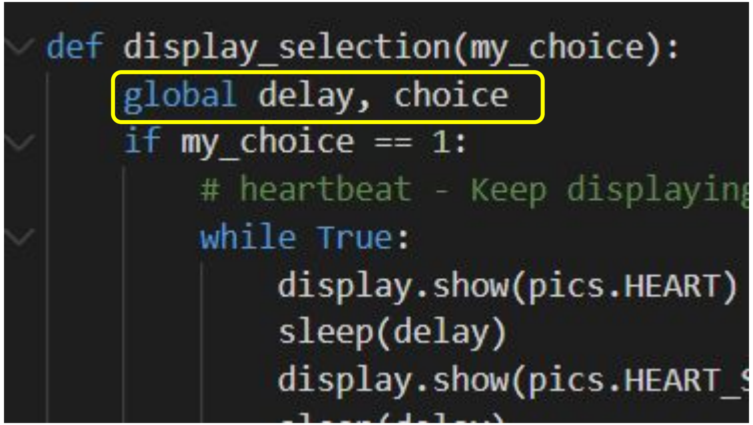


```
    elif my_choice == 2:
        # billboard -- read buttons and show images
        while True:
            my_image = my_list[choice]
```

Create_PT_Practice_5 1 of 1 problem

```
NameError: local variable referenced before assignment
```

# Step #4



```
def display_selection(my_choice):
    global delay, choice
    if my_choice == 1:
        # heartbeat - Keep displaying
        while True:
            display.show(pics.HEART)
            sleep(delay)
            display.show(pics.HEART_S
            sleep(delay)
```

## Fix all the little things that need fixing

- This is the SAME ERROR as the last one, but it stops at **choice**
- **choice** is also a global variable, but the computer thinks it is local – notice the error message
- You know what to do to fix this error

# Step #5

## Still one more thing to fix

Run the code again

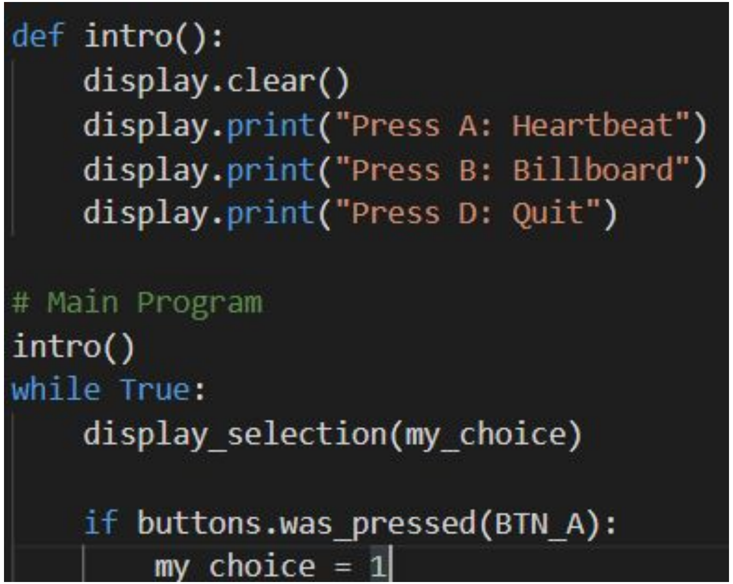
- Press either A or B (doesn't matter)
- As the code is running, press D to break the loop
- Does the loop "break" and the heartbeat or billboard stop?
- Why not????

# Step #5

## One more thing

- The loop does break, and the heartbeat or billboard does stop.
- However, in the main program, **my_choice** hasn't changed value, so it calls the function again with the same value, as if it hasn't stopped
- You need to reset the value of **my_choice** before breaking

# Step #5



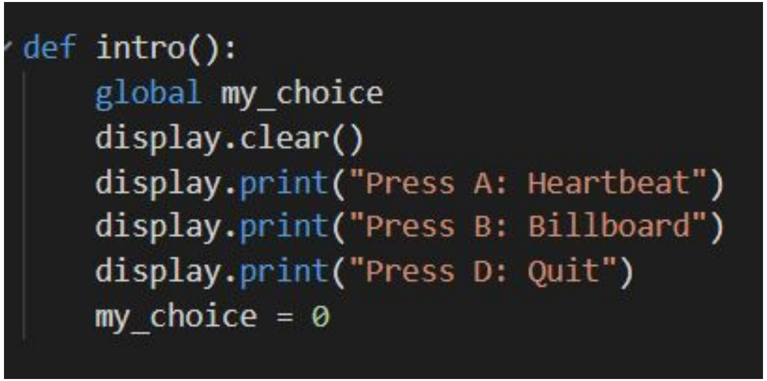
```
def intro():
    display.clear()
    display.print("Press A: Heartbeat")
    display.print("Press B: Billboard")
    display.print("Press D: Quit")

# Main Program
intro()
while True:
    display_selection(my_choice)

    if buttons.was_pressed(BTN_A):
        my_choice = 1
```

## Reset my_choice

There is more than one way to do this. We are going to solve another problem at the same time by creating a function.

- You may also notice that a user doesn’t know to press A for heartbeat or B for billboard
- Create a function for intro()
- For now, keep it simple
- Call the intro() function in the main program, before the while loop

# Step #5



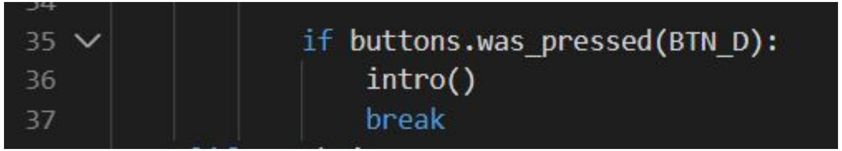
```
def intro():
    global my_choice
    display.clear()
    display.print("Press A: Heartbeat")
    display.print("Press B: Billboard")
    display.print("Press D: Quit")
    my_choice = 0
```

## Reset my_choice

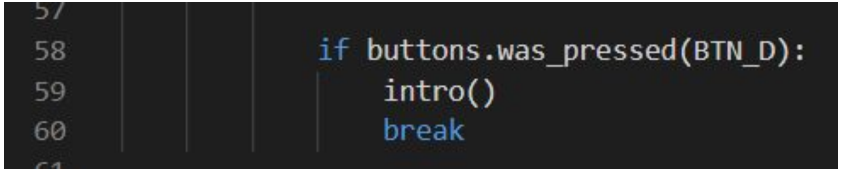
Run the code. This solves the problem of knowing what to press, but it still doesn’t reset **my_choice**

- Add a global declaration in the function
- Set the value of **my_choice** to 0
- Can you explain why the global declaration is needed?

# Step #5



```
            if buttons.was_pressed(BTN_D):
                intro()
                break
```



```
            if buttons.was_pressed(BTN_D):
                intro()
                break
```

## Reset my_choice

Call the **intro()** function in the function with a parameter (**display_selection**)

- You will need to call it **2 times**
  - Before the break in my_choice == 1
  - Before the break in my_choice == 2

# Step #6

## Test and Debug

- Run the code again
- Does it work properly?
- Press A for heartbeat, and then D for quitting
- Do you get the intro again?
- Press B for billboard, and then D for quitting
- Do you get the intro again?
- Press D to quit
- Do you get your ending?

FIRIA LABS

# Step #6

*You don’t need to add anything else for Practice5. Just think about it if you want to do a similar program for your actual Create PT.*

## Anything else?

What else could you do to make the program better?

- You could add an introduction (print statements) to the heartbeat code so the user knows how to work it
- You could add an intro to the billboard so the user knows how to work it
- You could add a button press in billboard for a random item in the list
- You could make the ending more interesting
- You are only limited by your creativity!

# Step #7

- Creates a list
- Uses a list in a meaningful way
- Has a function with a parameter
  - Parameter is used in an if statement (selection)
- Function has:
  - If statement (selection)
  - Loop (iteration)

## Meet the requirements?

Review the requirements for the Create Performance Task. In your code:

- Identify a list and where it is being created
- Identify where the list is being used
- Identify the function with a parameter
- Is the parameter used in an if statement?
- Does the function have a loop?

# And now you have another Create PT practice

## Congratulations!

By completing this practice project you have prepared for the PT by:

- Creating a list (Mission 7)
- Using the list in a meaningful way
- Creating a function with a parameter
- Calling the function
- Using the parameter in an if statement (my_choice)
- Using sequence and selection in the function