

Code Segment Practice Using Snippet

Using Snippet to create the Personalized Project Reference



Create PT Practice #2

In CodeSpace, open
Create_PT_Practice2

- Create a PDF document of the entire code
- Delete all comments from the code

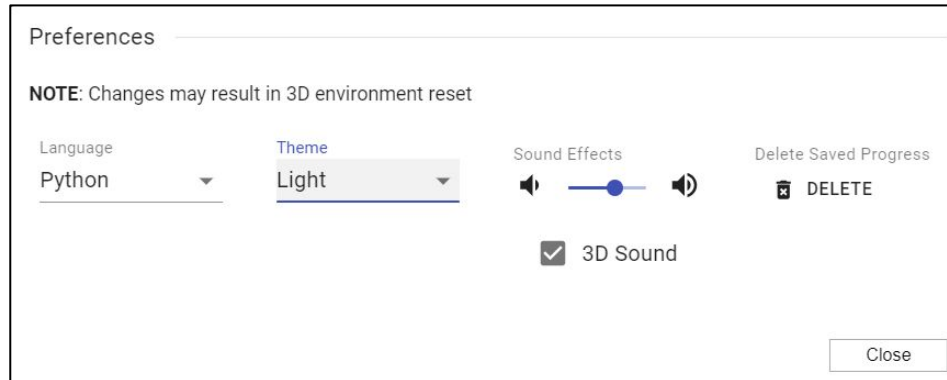
```
Create_PT_Practice2 x
1 from codex import *
2 from time import sleep
3 import random
4
5 dbacks_pos = ["pitcher", "catcher", "1st Base",
6             "2nd Base", "3rd Base", "shortstop"]
7 dbacks_players = ["Merrill Kelley", "Gabriel Moreno",
8                  "Ketel Marte", "Evan Longoria", "Geoff Blum"]
9 rangers_pos = ["cather", "3rd base", "1st base",
10              "shortstop", "2nd base", "outfielder"]
11 rangers_players = ["Mitch Garver", "Josh Jung", "Nolan Arenado",
12                   "Corey Seager", "Marcus Semian", "A.J. Pierrelle"]
13
14
15 def intro():
16     display.print("Welcome to the ")
17     display.print("World Series")
18     display.print("A = Diamondbacks")
19     display.print("B = Rangers")
20     display.print("")
21     display.print("R = Scroll forward")
22     display.print("L = Slideshow")
23     display.print("U = Random player")
24     display.print("D = Quit")
25
26 def ending():
27     display.clear()
28     display.print("Thank you!")
29     display.print("Have a good day!")
30
```



Create PT Practice #2

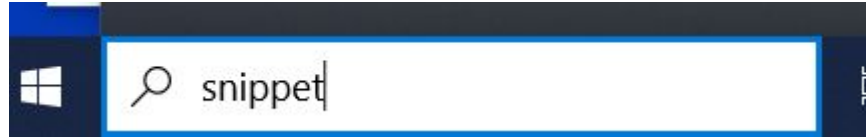
The code will be easier to see and print if the text is black and background is white.

- Click on settings icon
- Change the theme to light



Open the Snippet app

- Go to the search window on the taskbar
- Type snippet

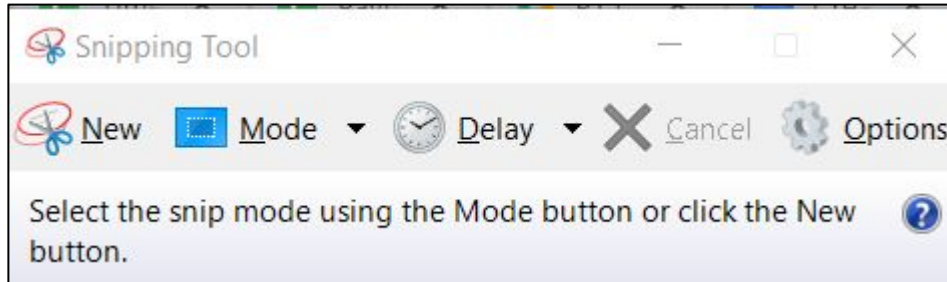


- The app will appear in the pop-up



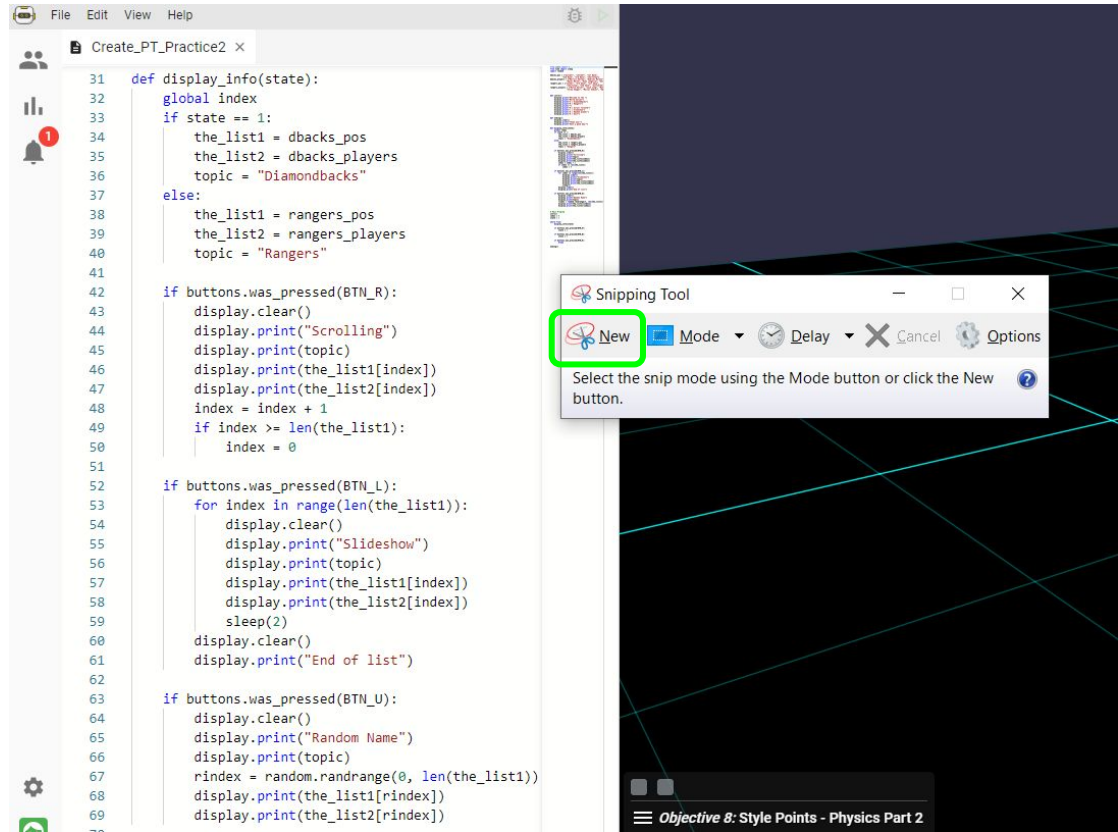
Open the Snippet app

- Click on the app
- The app will appear as a moveable window
- Move the window around your screen so it is out of the way.



First code segment

- Return to your practice PT in CodeSpace
- Identify the first segment – a function with a parameter, loop and if statement.



The image shows a CodeSpace editor window titled "Create_PT_Practice2" with Python code. The code defines a function `display_info(state)` and includes event handlers for buttons `BTN_R`, `BTN_L`, and `BTN_U`. A Snipping Tool window is overlaid on the right side of the editor, with the "New" button highlighted by a green circle. The Snipping Tool window contains the following text:

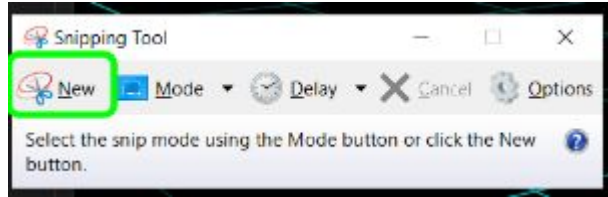
```
Snipping Tool
New Mode Delay Cancel Options
Select the snip mode using the Mode button or click the New button.
```

The background of the CodeSpace editor shows a dark theme with a grid pattern. At the bottom of the editor, there is a status bar that reads "Objective 8: Style Points - Physics Part 2".



First code segment

- On the snipping tool app, click “NEW”



- Draw a rectangle around the entire function

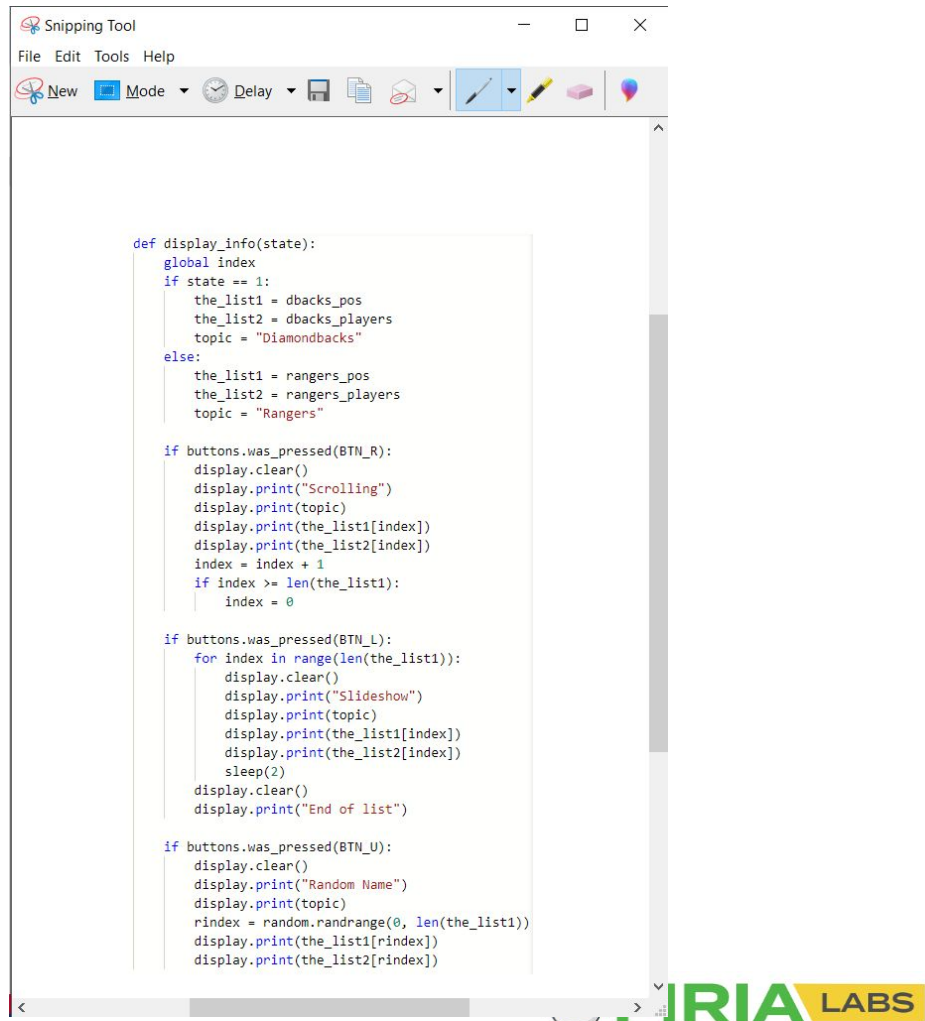
```
File Edit View Help
Create_PT_Practice2 x

31 def display_info(state):
32     global index
33     if state == 1:
34         the_list1 = dbacks_pos
35         the_list2 = dbacks_players
36         topic = "Diamondbacks"
37     else:
38         the_list1 = rangers_pos
39         the_list2 = rangers_players
40         topic = "Rangers"
41
42     if buttons.was_pressed(BTN_R):
43         display.clear()
44         display.print("Scrolling")
45         display.print(topic)
46         display.print(the_list1[index])
47         display.print(the_list2[index])
48         index = index + 1
49         if index >= len(the_list1):
50             index = 0
51
52     if buttons.was_pressed(BTN_L):
53         for index in range(len(the_list1)):
54             display.clear()
55             display.print("Slideshow")
56             display.print(topic)
57             display.print(the_list1[index])
58             display.print(the_list2[index])
59             sleep(2)
60         display.clear()
61         display.print("End of list")
62
63     if buttons.was_pressed(BTN_U):
64         display.clear()
65         display.print("Random Name")
66         display.print(topic)
67         rindex = random.randrange(0, len(the_list1))
68         display.print(the_list1[rindex])
69         display.print(the_list2[rindex])
70
71
```



First code segment

- The function snippet will appear in the snipping tool app



The screenshot shows the Snipping Tool application window. The title bar reads "Snipping Tool" and the menu bar includes "File", "Edit", "Tools", and "Help". The toolbar contains icons for "New", "Mode", "Delay", and various snipping tools. The main area displays a Python code snippet with the following content:

```
def display_info(state):
    global index
    if state == 1:
        the_list1 = dbacks_pos
        the_list2 = dbacks_players
        topic = "Diamondbacks"
    else:
        the_list1 = rangers_pos
        the_list2 = rangers_players
        topic = "Rangers"

    if buttons.was_pressed(BTN_R):
        display.clear()
        display.print("Scrolling")
        display.print(topic)
        display.print(the_list1[index])
        display.print(the_list2[index])
        index = index + 1
        if index >= len(the_list1):
            index = 0

    if buttons.was_pressed(BTN_L):
        for index in range(len(the_list1)):
            display.clear()
            display.print("Slideshow")
            display.print(topic)
            display.print(the_list1[index])
            display.print(the_list2[index])
            sleep(2)
        display.clear()
        display.print("End of list")

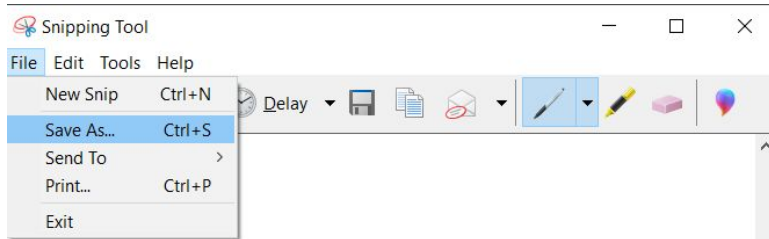
    if buttons.was_pressed(BTN_U):
        display.clear()
        display.print("Random Name")
        display.print(topic)
        rindex = random.randrange(0, len(the_list1))
        display.print(the_list1[rindex])
        display.print(the_list2[rindex])
```

In the bottom right corner of the application window, there is a logo for "IRIA LABS".



First code segment

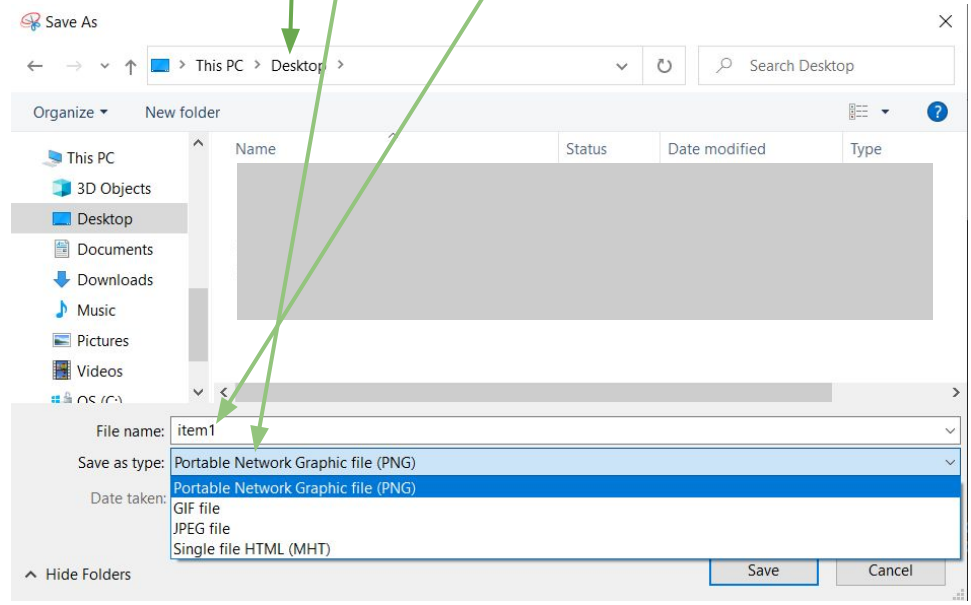
- In the app, go to:
- File
- Save As



```
def display_info(state):
    global index
    if state == 1:
        the_list1 = dbacks_pos
        the_list2 = dbacks_players
        topic = "Diamondbacks"
    else:
        the_list1 = rangers_pos
        the_list2 = rangers_players
        topic = "Rangers"

    if buttons.was_pressed(BTN_R):
        display.clear()
        display.print("Scrolling")
        display.print(topic)
        display.print(the_list1[index])
        display.print(the_list2[index])
        index = index + 1
```

- Give the snippet a name
- Choose PNG (preferred) or JPEG
- Select where to save the file



Second code segment

- Return to your practice PT in CodeSpace
- Identify the second segment – the function call
- Click “New” on the Snippet App
- Draw a rectangle around the code segment

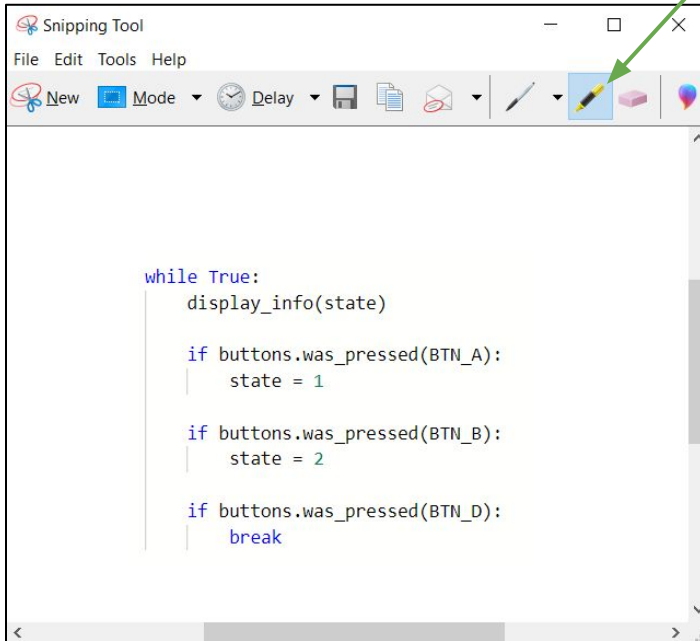
```
67 |         rindex = random.randrange(0, len(
68 |         display.print(the_list1[rindex])
69 |         display.print(the_list2[rindex])
70 |
71 |
72 | # Main Program
73 | intro()
74 | index = 0
75 | state = 1
76 |
77 | while True:
78 |     display_info(state)
79 |
80 |     if buttons.was_pressed(BTN_A):
81 |         state = 1
82 |
83 |     if buttons.was_pressed(BTN_B):
84 |         state = 2
85 |
86 |     if buttons.was_pressed(BTN_D):
87 |         break
88 |
89 | ending()
90 |
```



Second code segment

- The function snippet will appear in the snipping tool app

- Click on the highlighter
- Use a light color and highlight the line of code that is the function call



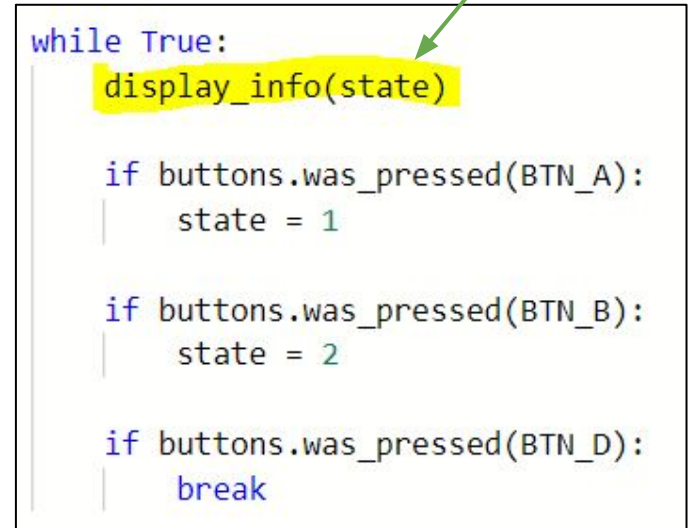
The screenshot shows the Snipping Tool application window. The title bar reads "Snipping Tool" and the menu bar includes "File", "Edit", "Tools", and "Help". The toolbar contains icons for "New", "Mode", "Delay", "Save", "Copy", "Paste", "Highlighter", "Eraser", and "Redaction". The main area displays a Python code snippet:

```
while True:
    display_info(state)

    if buttons.was_pressed(BTN_A):
        state = 1

    if buttons.was_pressed(BTN_B):
        state = 2

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



A close-up view of the code snippet from the previous image. The line `display_info(state)` is highlighted in yellow. A green arrow points from the highlighter icon in the Snipping Tool toolbar to this highlighted line.

```
while True:
    display_info(state)

    if buttons.was_pressed(BTN_A):
        state = 1

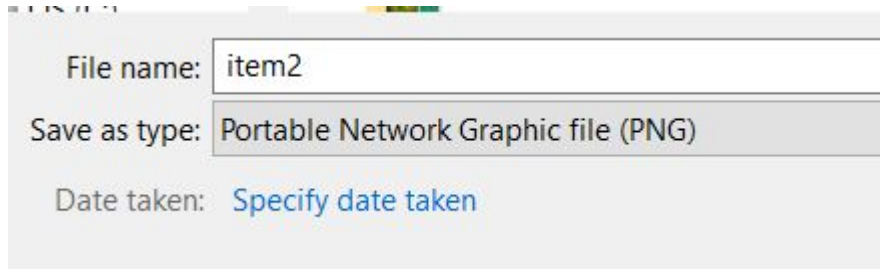
    if buttons.was_pressed(BTN_B):
        state = 2

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



Second code segment

- In the app, go to:
- File
- Save As
- Give the snippet a name and save
 - item2
 - PNG
 - Save in same location as item1



File name: item2

Save as type: Portable Network Graphic file (PNG)

Date taken: [Specify date taken](#)



Third code segment

- Return to your practice PT in CodeSpace
- Identify the third segment – the list creation
- Click “New” on the Snippet App
- Draw a rectangle around the code segment

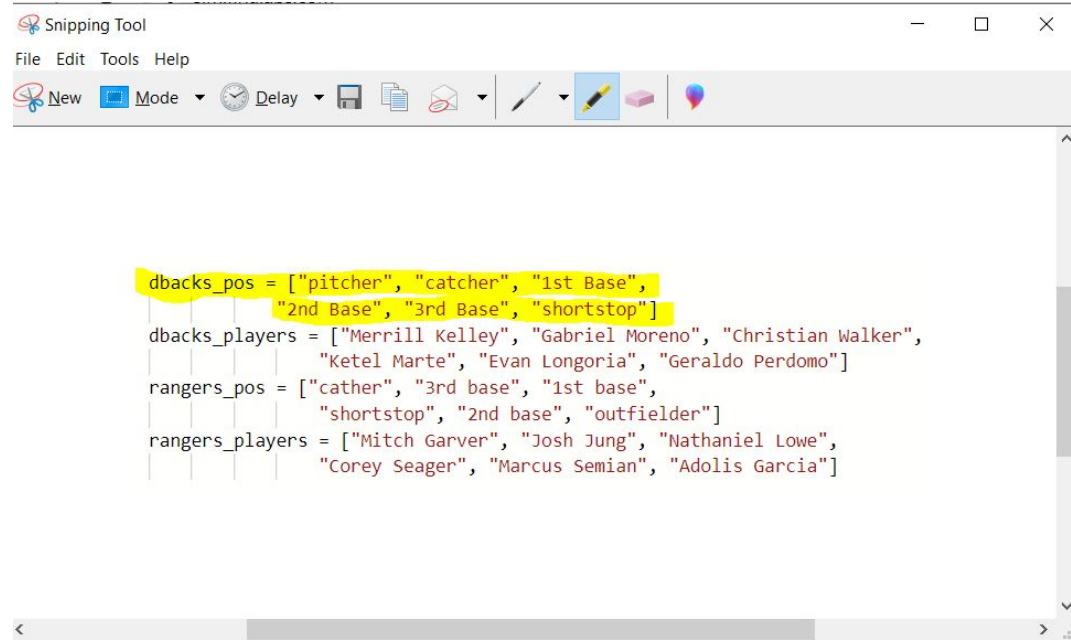
NOTE: *you can snip just one list if you want; you don't have to include more than one.*

```
1 from codex import *
2 from time import sleep
3 import random
4
5 dbacks_pos = ["pitcher", "catcher", "1st Base",
6              "2nd Base", "3rd Base", "shortstop"]
7 dbacks_players = ["Merrill Kelley", "Gabriel Moreno", "Christian Walker",
8                  "Ketel Marte", "Evan Longoria", "Geraldo Perdomo"]
9 rangers_pos = ["catcher", "3rd base", "1st base",
10               "shortstop", "2nd base", "outfielder"]
11 rangers_players = ["Mitch Garver", "Josh Jung", "Nathaniel Lowe",
12                   "Corey Seager", "Marcus Semian", "Adolis Garcia"]
13
14
15 def intro():
16     display.print("Welcome to the ")
17     display.print("World Series")
18     display.print("A = Diamondbacks")
19     display.print("B = Rangers")
20     display.print("")
```



Third code segment

- The function snippet will appear in the snipping tool app
- Click on the highlighter
- Use a light color and highlight the line of code for ONE list



The screenshot shows the Snipping Tool application window. The title bar reads "Snipping Tool" with standard window controls. The menu bar includes "File", "Edit", "Tools", and "Help". The toolbar contains icons for "New", "Mode", "Delay", a save icon, a copy icon, a paste icon, a selection tool, a highlighter (which is active), a eraser, and a location pin. The main area displays a code snippet with a yellow highlighter applied to the first line: `dbacks_pos = ["pitcher", "catcher", "1st Base", "2nd Base", "3rd Base", "shortstop"]`. The rest of the code is as follows:

```
dbacks_players = ["Merrill Kelley", "Gabriel Moreno", "Christian Walker",  
                 "Ketel Marte", "Evan Longoria", "Geraldo Perdomo"]  
rangers_pos = ["cather", "3rd base", "1st base",  
              "shortstop", "2nd base", "outfielder"]  
rangers_players = ["Mitch Garver", "Josh Jung", "Nathaniel Lowe",  
                  "Corey Seager", "Marcus Semian", "Adolis Garcia"]
```



Third code segment

- In the app, go to:
- File
- Save As
- Give the snippet a name and save
 - item3
 - PNG
 - Save in same location as item1 and item2



Fourth code segment

- Return to your practice PT in CodeSpace
- Identify the fourth segment – the list being used
- Click “New” on the Snippet App
- Draw a rectangle around the code segment

NOTE: *you can snip just one list if you want; you don't have to include more than one.*

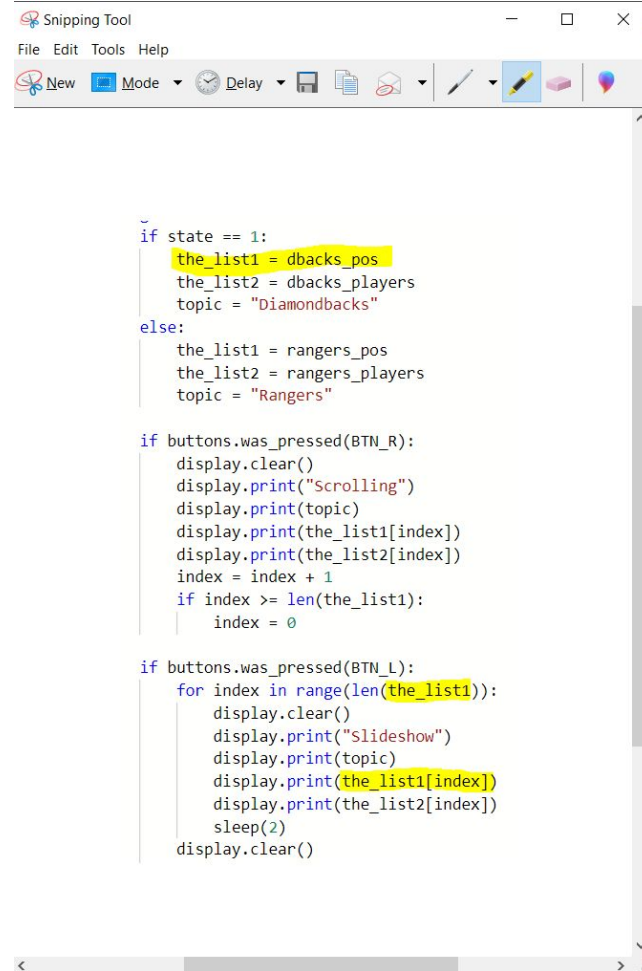
```
def display_info(state):  
    global index  
    if state == 1:  
        the_list1 = dbacks_pos  
        the_list2 = dbacks_players  
        topic = "Diamondbacks"  
    else:  
        the_list1 = rangers_pos  
        the_list2 = rangers_players  
        topic = "Rangers"  
  
    if buttons.was_pressed(BTN_R):  
        display.clear()  
        display.print("Scrolling")  
        display.print(topic)  
        display.print(the_list1[index])  
        display.print(the_list2[index])  
        index = index + 1  
        if index >= len(the_list1):  
            index = 0  
  
    if buttons.was_pressed(BTN_L):  
        for index in range(len(the_list1)):  
            display.clear()  
            display.print("slideshow")  
            display.print(topic)  
            display.print(the_list1[index])  
            display.print(the_list2[index])  
            sleep(2)  
        display.clear()  
        display.print("End of list")
```



Fourth code segment

- The function snippet will appear in the snipping tool app
- Click on the highlighter
- Use a light color and highlight the lines of code where the one already identified list is used

NOTE: make sure you highlight the SAME list being used as being created



The screenshot shows the Snipping Tool application window. The title bar reads "Snipping Tool" and the menu bar includes "File", "Edit", "Tools", and "Help". The toolbar contains icons for "New", "Mode", "Delay", "Copy", "Paste", "Erase", "Highlight", "Redact", and "Screenshot". The main area displays a Python code snippet with the following content:

```
if state == 1:
    the_list1 = dbacks_pos
    the_list2 = dbacks_players
    topic = "Diamondbacks"
else:
    the_list1 = rangers_pos
    the_list2 = rangers_players
    topic = "Rangers"

if buttons.was_pressed(BTN_R):
    display.clear()
    display.print("scrolling")
    display.print(topic)
    display.print(the_list1[index])
    display.print(the_list2[index])
    index = index + 1
    if index >= len(the_list1):
        index = 0

if buttons.was_pressed(BTN_L):
    for index in range(len(the_list1)):
        display.clear()
        display.print("slideshow")
        display.print(topic)
        display.print(the_list1[index])
        display.print(the_list2[index])
        sleep(2)
    display.clear()
```

In the code, the lines `the_list1 = dbacks_pos`, `the_list1 = rangers_pos`, `display.print(the_list1[index])`, and `display.print(the_list1[index])` are highlighted in yellow.



Fourth code segment

- In the app, go to:
- File
- Save As
- Give the snippet a name and save
 - item4
 - PNG
 - Save in same location as the other snippets



Ready for uploading

- Your four images are now ready for uploading

and main methods, are not considered student-developed.

- The first program code segment must be a student-developed procedure that:
 - Defines the procedure's name and return type (if necessary)
 - Contains and uses one or more parameters that have an effect on the functionality of the procedure
 - Implements an algorithm that includes sequencing, selection, and iteration

- The second program code segment must show where your student-developed procedure is being called in your program.

