

Create PT Prep

Hints on doing extra things in your code



#1: adding audio pitches

You already know how to play music. What if you just want to play some sounds?

- Include this audio command near the top of your code
 - can be just under the imports
 - or at the top of the main program

```
# Main Program
audio.enable_audio_amp(True)
intro()
continues = True
```



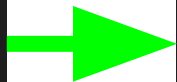
#1: adding audio pitches

- Use this code to play a pitch:

```
audio.pitch(tone, delay)
```

- The tone is a number
 - The smaller the number the lower the pitch
 - The larger the number the higher the pitch
- The delay is the amount of time to play the note
 - It can replace sleep(delay)

```
pixels.set(0, GREEN)  
sleep(delay)
```



```
pixels.set(0, GREEN)  
audio.pitch(1200, delay)
```



#2: Map a number to something

If your code gets a random number, like for a pixel, and you want to use a specific color, image, or pitch for that number, you can use an if statement to “map” the number to the color, image and/or pitch

- Just use a separate if statement for each thing you want to “map” to a number, and return the value



#2: Map a number to something

Here is an example for color:

- Create a function
- Use the number to map as a parameter
 - Use if statements to assign a value
- Return the value

Example of call statement for the function:

```
col = get_color(pix)
```

```
def get_color(pixel):  
    if pixel == 0:  
        col = RED  
    elif pixel == 1:  
        col = BLUE  
    elif pixel == 2:  
        col = GREEN  
    else:  
        col = YELLOW  
    return col
```



#2: Map a number to something

Here is an example for pitch:

- Create a function
- Use the number to map as a parameter
 - Use if statements to assign a value
- Return the value

Example of call statement for the function:

```
tone = get_tone(pix)
```

```
def get_tone(pixel):  
    if pixel == 0:  
        pitch = 800  
    elif pixel == 1:  
        pitch = 1200  
    elif pixel == 2:  
        pitch = 1700  
    else:  
        pitch = 2300  
    return pitch
```



#2: Map a number to something

Here is an example for an image:

- Create a function
- Use the number to map as a parameter
- Return the value

What would the call statement look like?

```
def get_image(number):  
    if number == 0:  
        pic = pics.ARROW_N  
    elif number == 1:  
        pic = pics.ARROW_NE  
    elif number == 2:  
        pic = pics.ARROW_E  
    else:  
        pic = pics.ARROW_SE  
    return pic
```



#2: Map a number to something

Here is an example of it all put together, using tone and color:

What would the code look like with an image?

```
for i in range(len(sequence)):
    pix = sequence[i]
    tone = get_tone(pix)
    col = get_color(pix)
    pixels.set(pix, col)
    audio.pitch(tone, delay)
    pixels.set(pix, BLACK)
    sleep(delay/2)
```



#3: Traversing a list

You have traversed a list using a for loop. But usually you use the counter in the loop as the index.

- What if you need to use a counter for the loop and a different variable for the index of the list?
- This can be especially useful for traversing a list backwards



#3: Traversing a list

- The list can be arrows or other images
- Code that goes forward
- **index** is a variable
- Initialize **index** to 0
 - (first item in the list)
- The variable **i** (or anything else) is used as the loop counter
- Increment the **index** inside the loop

```
def spin_animation(count):  
    index = 0  
    delay = 0.1  
    for i in range(count):  
        display.show(arrows[index])  
        sleep(delay)  
        delay = delay + 0.005  
        index = index + 1  
        if index == 8:  
            index = 0  
    num = random.randrange(8)  
    display.show(arrows[num])
```



#3: Traversing a list

- Code that goes backward
- **index** is a variable
- **index** is initialized to the last index
 - Last item in the list
- The variable **i** (or anything else) is used as the loop counter
- Decrement the **index** inside the loop

```
def spin_animation_reverse(count):  
    index = len(arrows)-1  
    delay = 0.1  
    for i in range(count):  
        display.show(arrows[index])  
        sleep(delay)  
        delay = delay + 0.005  
        index = index - 1  
        if index == -1:  
            index = len(arrows) - 1  
    num = random.randrange(8)  
    display.show(arrows[num])
```



Selecting a project

You should fully understand the requirements of the Create PT and are ready to start brainstorming an idea for your project. You will be given class time to complete the project. Keep these things in mind:

- Should be able to complete the code in 7 hours (not too complicated)
- Personally relevant – pick a project you are interested in
- Has a clear purpose (educational, entertainment, etc.)
- No new programming skills – This isn't the time to learn something new. Stick with what you already know and make it work.



Final comments:

- Your project should have a detailed **intro()** with clear instructions
 - Tell the user what buttons to use for what!!
- Your project should have some kind of ending
- Start small, and build from there.
 - Get the basics done, with input and output, a list and a function.
 - Go step-by-step (have a plan)
 - Test as you go, a few lines at a time, debugging as necessary
 - Then if you have time, add more to the project.
- You can get feedback from your peers
- You can get help from your peers
- DO NOT ask the teacher!

