

## AP CSP Unit 3 CodeX Python Code By Mission

Types of Division	
Absolute value	<pre>abs(num)</pre> Returns the absolute value of num
Round	<pre>round(num)</pre> Rounds up or down the float num, returns the integer
Float (real) division	<pre>num / 5</pre> Returns the decimal quotient
Integer division	<pre>num // 5</pre> Returns the integer only quotient (no rounding)
Modulo division	<pre>num % 5</pre> Returns the integer remainder only
Mission 9 - Game Spinner	
Using a logical operator:	<pre>if buttons.is_pressed(BTN_A) or buttons.is_pressed(BTN_B):</pre>
Define a function	<pre>def show_random_arrow():     num = random.randrange(8)     display.show(pics.ALL_ARROWS[num])</pre>
Call a function	<pre>while True:     if buttons.is_pressed(BTN_A) or buttons.is_pressed(BTN_B):         show_random_arrow()</pre>
Finite loop with condition  (increment the control variable)	<pre>while index &lt; 8:     my_arrow = pics.ALL_ARROWS[index]     display.show(my_arrow)     sleep(0.1)     index = index + 1</pre>
Finite loop with condition and list wrapping	<pre>while loops &lt; count:     my_arrow = pics.ALL_ARROWS[index]     display.show(my_arrow)     sleep(delay)     delay = delay + 0.005     loops = loops + 1     index = index + 1     if index == 8:         index = 0</pre>

## Functions with Parameters

Function with parameter

```
def turn_pixel(color):  
def game(message, button):
```

Function call with argument

```
turn_pixel(RED)  
game("Press B", BTN_B)
```

## Traversing a List

For loop that traversing a list

```
for index in range(len(answers)):  
    display.print(answers[index])  
    sleep(1)
```

Specialized for loop (traverses a list)

```
for item in answers:  
    display.print(item)  
    sleep(1)
```

For loop for lighting pixels

```
for pix in range(4):  
    pixels.set(pix, random.choice(COLOR_LIST))
```

For loop that traverses two lists

```
for index in range(len(pix_colors)):  
    turn_pixels(pix_colors[index], pix_brights[index])
```

Create a matrix (list of lists)

```
pix_info = [  
    [(77, 158, 100), 75],  
    [YELLOW, 50],  
    [(203, 182, 6), random.randrange(100)],  
    [RED, 50],  
    [random.choice(COLOR_LIST), 100],  
    [YELLOW, 75],  
    [BLACK, 0]  
]
```

For loop that traverses a matrix

```
for index in range(len(pix_info)):  
    turn_pixels(pix_colors[index][0], pix_brights[index][1])
```

For loop to look for an item in a list

```
def check_schedule(course):  
    for item in schedule:  
        if item == course:  
            display.print("On the schedule")
```

Specialized loop to look for an item in a list	<pre>def check_schedule(course):     if course in schedule:         display.print(course)         display.print("On the schedule")  — or — def check_schedule(course):     if course in schedule:         display.print(course)         display.print("On the schedule")     else:         display.print(course)         display.print("Not on schedule")</pre>
Input a string value on the Console	<pre>course = input("Enter a class: ")</pre>
Input an integer value on the Console	<pre>number = int(input("Enter a number: "))</pre>
Fill an empty list with random numbers	<pre>def fill_list():     for i in range(20):         number = random.randrange(50)         numlist.append(number)</pre>
Filter a list (create a sub-list)	<pre>def filter_list():     for item in numlist:         if item &lt; 20:             filteredlist.append(item)</pre>
Printing on the CodeX display	<pre>def print_list():     display.print("From your random")     display.print("list of " + str(len(numlist)))     display.print("numbers, " + str(len(filteredlist)))     display.print("were less than ")     display.print(str(20))     display.print()     display.print(filteredlist)</pre>

Filter a list by comparing it to another list

```
def filter_list():  
    filteredlist = []  
    for item in my_animals:  
        if item in farm_animals:  
            filteredlist.append(item)  
    display.print("Farm Animals you listed:")  
    display.print(filteredlist)
```