

# Check Your Understanding - Part 1



Name: \_\_\_\_\_

Problem #1 🌶️	
What would you call the function?	def ending (answers can vary)
What are the variables needed?	count
What are the parameters?	count
What are the local variables?	none
Does it need a return?	no
Write a function call:	ending(count) or ending(3) or ending(4)

```
# Ending message
if count == 4:
    display.clear()
    display.draw_text("You WON", scale=4,
else:
    display.clear()
    display.draw_text("You LOST", scale=4,
```

Problem #2 🌶️	
What would you call the function?	def display_dice (answers vary)
What are the variables needed?	num
What are the parameters?	num
What are the local variables?	none
Does it need a return?	no
Write a function call:	display_dice(num) or display_dice(3) – or any number 1-6 for the argument

```
if buttons.was_pressed(BTN_B):
    # Reset the board for each game
    reset()
    # Select first random number
    num1 = random.randrange(6) + 1
    if num == 1:
        one_roll()
    elif num == 2:
        two_roll()
    elif num == 3:
        three_roll()
    elif num == 4:
        four_roll()
    elif num == 5:
        five_roll()
    else:
        six_roll()
    sleep(delay)
```

Problem #3 🌶️	
What would you call the function?	def instruction (answers vary)
What are the variables needed?	delay
What are the parameters?	delay
What are the local variables?	none
Does it need a return?	no
Write a function call:	instruction(delay)

```
pixels.set(3, BLACK)
if buttons.was_pressed(BTN_A):
    audio.mp3("sounds/welcome")
if buttons.was_pressed(BTN_B):
    display.show(pics.HAPPY)
sleep(delay)
display.fill(BLACK)
display.show("Press a Button!")
sleep(delay)
```

## Check Your Understanding - Part 1



Problem #4 🌶️🌶️	
What would you call the function?	<code>def turn_off (answers vary)</code>
What are the variables needed?	<code>lite, how_many</code>
What are the parameters?	<code>how_many</code>
What are the local variables?	<code>lite</code>
Does it need a return?	<code>no</code>
Write a function call:	<code>turn_off(3) or turn_off(how_many)</code> — or any number 1-4 as the argument

```
red = random.randrange(0, 255)
green = random.randrange(0, 255)
blue = random.randrange(0, 255)
color = (red, green, blue)

pixels.set(1, color)

how_many = 4
# turn off pixel LEDs
for lite in range(how_many):
    pixels.set(lite, BLACK)
```

Problem #5 🌶️🌶️	
What would you call the function?	<code>def random_color (answers vary)</code>
What are the variables needed?	<code>red, green, blue, color</code>
What are the parameters?	<code>none</code>
What are the local variables?	<code>red, green, blue, color</code>
Does it need a return?	<code>Yes – color</code>
Write a function call:	<code>color = random_color()</code>

```
while True:
    red = random.randrange(0, 255)
    green = random.randrange(0, 255)
    blue = random.randrange(0, 255)
    color = (red, green, blue)

    pixels.set(0, color)

    red = random.randrange(0, 255)
    green = random.randrange(0, 255)
    blue = random.randrange(0, 255)
    color = (red, green, blue)
```

Problem #6 🌶️🌶️	
What would you call the function?	<code>def display_image (answers vary)</code>
What are the variables needed?	<code>my_image, choice</code> ( <code>my_list</code> is a list and is automatically available throughout the program)
What are the parameters?	<code>choice</code>
What are the local variables?	<code>my_image</code>
Does it need a return?	<code>no</code>
Write a function call:	<code>display_image(choice)</code>

```
if buttons.was_pressed(BTN_L):
    choice = 4
if buttons.was_pressed(BTN_R):
    choice = 5

my_image = my_list[choice]

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