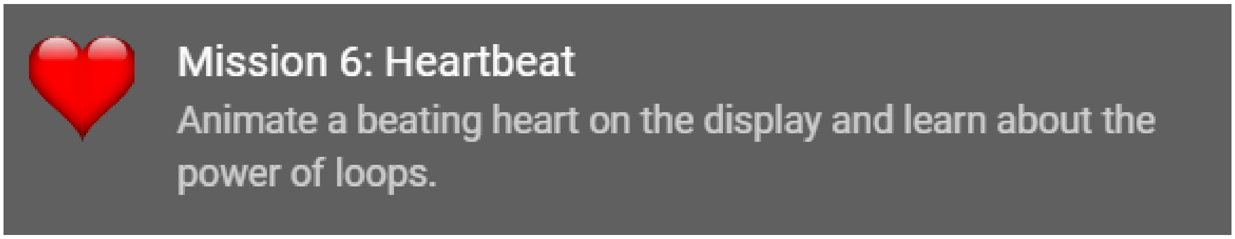
Mission 6:
Heartbeat

Student Workbook

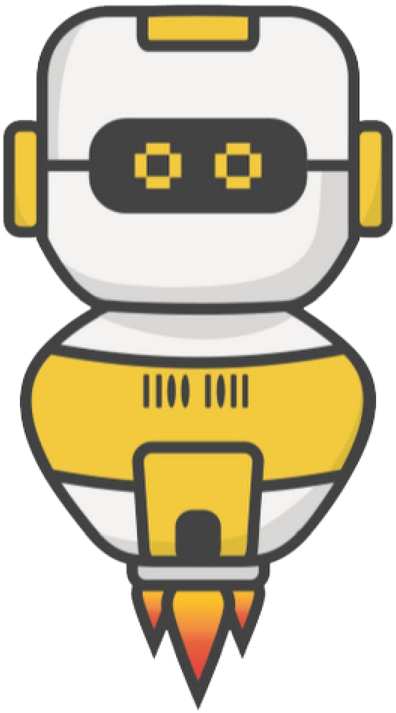
FIRIA LABS



Mission 6: Heartbeat

Animate a beating heart on the display and learn about the power of loops.

Let’s keep it going



Your programs are cool, but they only run one time. Then they stop. Let’s keep the code going by repeating the code in a loop. Also, you will use buttons to control the code.

You have probably seen flashing traffic road signs or traffic lights.

Go to the Mission 6 Log and fill out the Pre-Mission preparation.

- Make a list of blinking indicators (like flashing traffic road signs)

Mission 6: Heartbeat

In this project you'll give the CodeX a *beating heart.*

Okay, not a *real* heart – that would be a little too messy!

But using the display you can give the CodeX its own *digital* heart, and even make it speed up and slow down just like your own heart does.

Mission 6: Get started

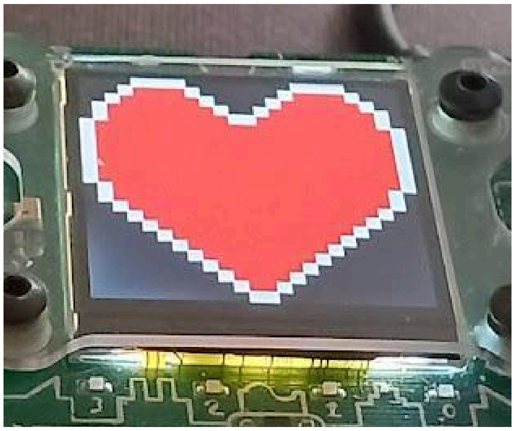
- Go to https://sim.firialabs.com/ and log in.
- Go to Mission 6 ❤
- Click NEXT and start Mission 6.

Objective #1: Lots of heart

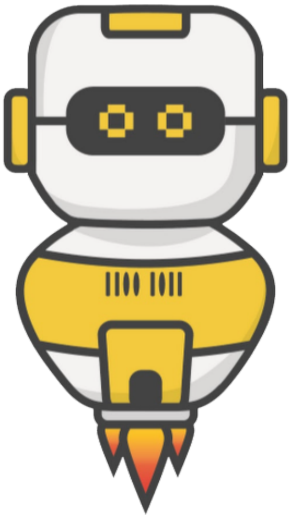
Start by reviewing programming concepts from your earlier missions.

Show a heart image on the screen.

- You might recognize this as the same code as your first project.
- Don't worry, you're going to add a *lot* of new features soon!

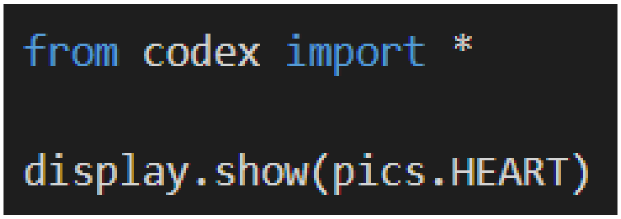


Objective #1: Lots of heart

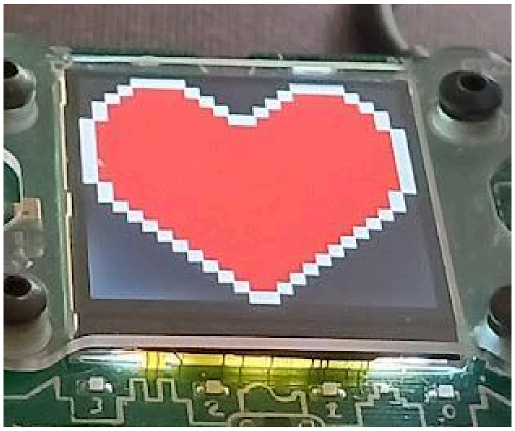


DO THIS:

- Start a new file named **Heart2**
- Show pics.HEART on the CodeX display
  - ○ Use CodeTrek if you need help



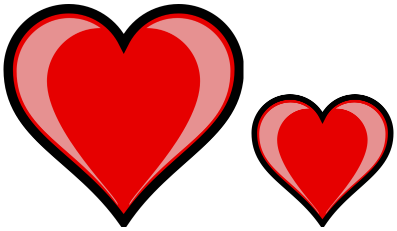
```
from codex import *

display.show(pics.HEART)
```

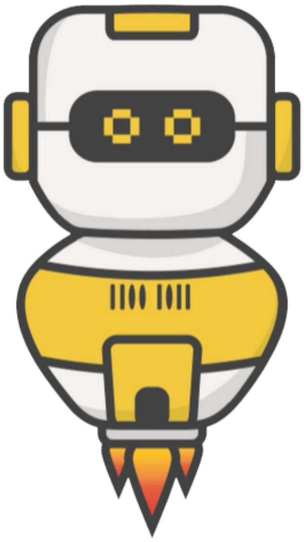


Objective #2: Pump it UP

Now you will animate the heart to make it look like it is beating.

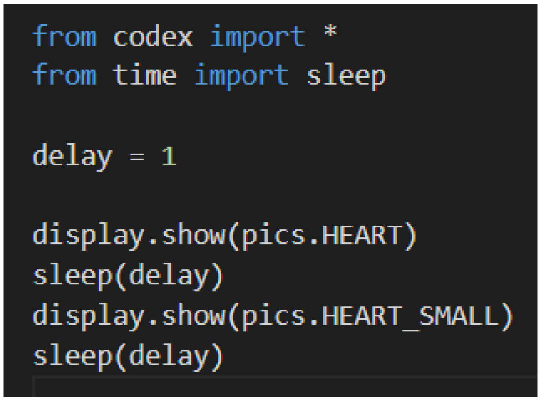
- You will need two heart images:
  - **pics.HEART**
  - **pics.HEART_SMALL**
- You need a **delay** variable to see both images
- You need to **import sleep** to use the **delay** variable





DO THIS:

- From time, import sleep
- Define a delay variable
- Show the first heart
  - Then sleep
- Show the second heart
  - Then sleep



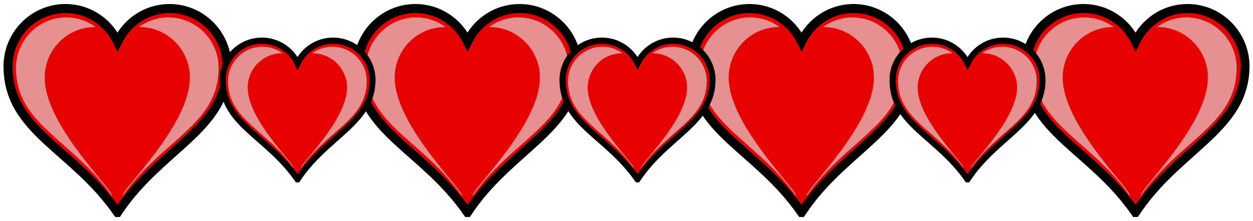
```
from codex import *
from time import sleep

delay = 1

display.show(pics.HEART)
sleep(delay)
display.show(pics.HEART_SMALL)
sleep(delay)
```

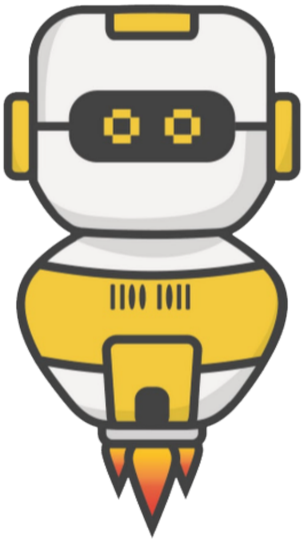
Objective #3: Repeat da beat

Now you have a heartbeat.

But one heartbat isn't an animation.

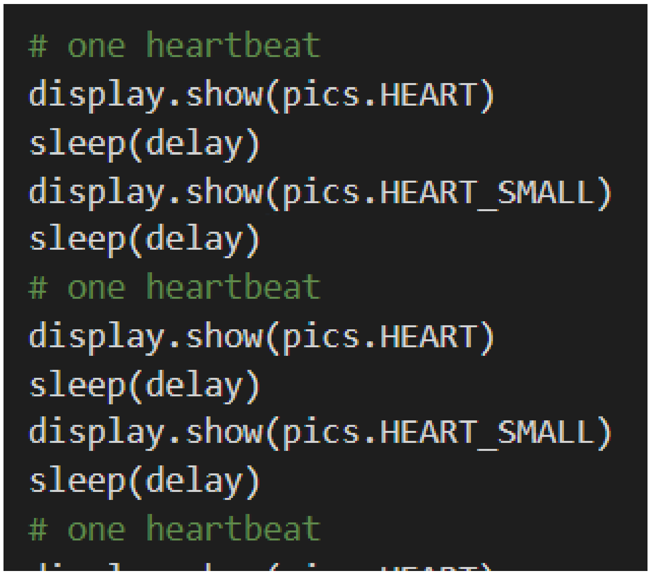
- You can repeat the code to repeat the heartbeat several times





DO THIS:

- Repeat the code for the large and small hearts at least **4** times



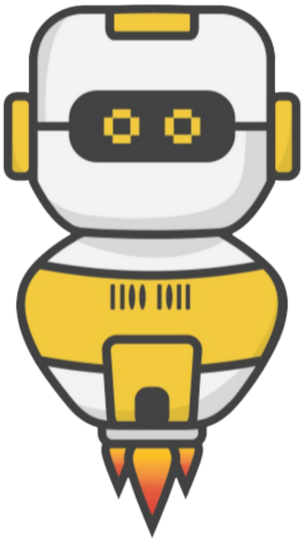
```
# one heartbeat
display.show(pics.HEART)
sleep(delay)
display.show(pics.HEART_SMALL)
sleep(delay)
# one heartbeat
display.show(pics.HEART)
sleep(delay)
display.show(pics.HEART_SMALL)
sleep(delay)
# one heartbeat
display.show(pics.HEART)
```

Objective #4: Hearts forever

Four beats is a good animation, but it ends.

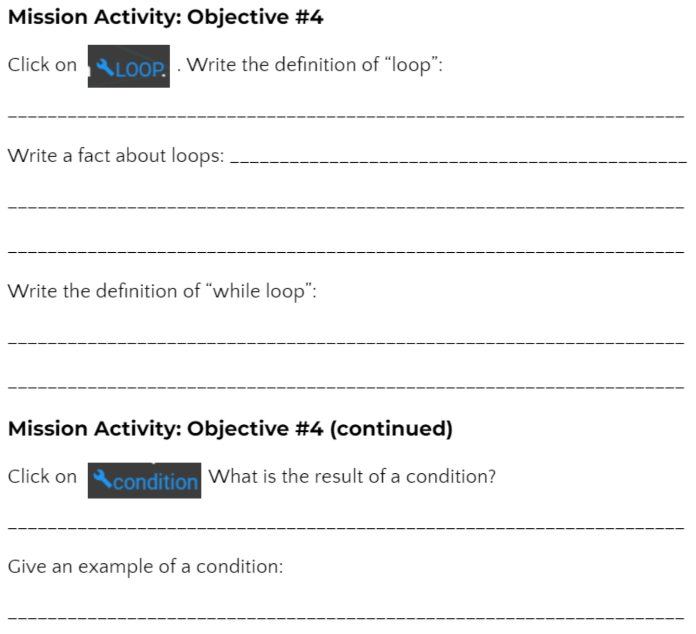
We want our heart animation to run forever.

- You can't just copy millions of times
- Instead, tell the computer to repeat the code
- Repeating code without retyping is called a LOOP



DO THIS:

- Click on 🔧LOOP
- Go to your Mission Log and answer the questions for Objective #4
- Click on 🔧condition
- Go to your Mission Log and answer the questions for Objective #4



Mission Activity: Objective #4

Click on 🔧LOOP . Write the definition of "loop":

____

Write a fact about loops: ____

____

____

Write the definition of "while loop":

____

____

Mission Activity: Objective #4 (continued)

Click on 🔧condition What is the result of a condition?

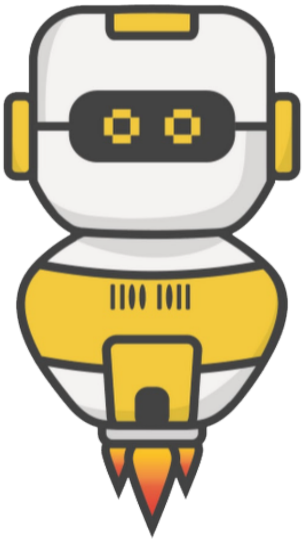
____

Give an example of a condition:

____

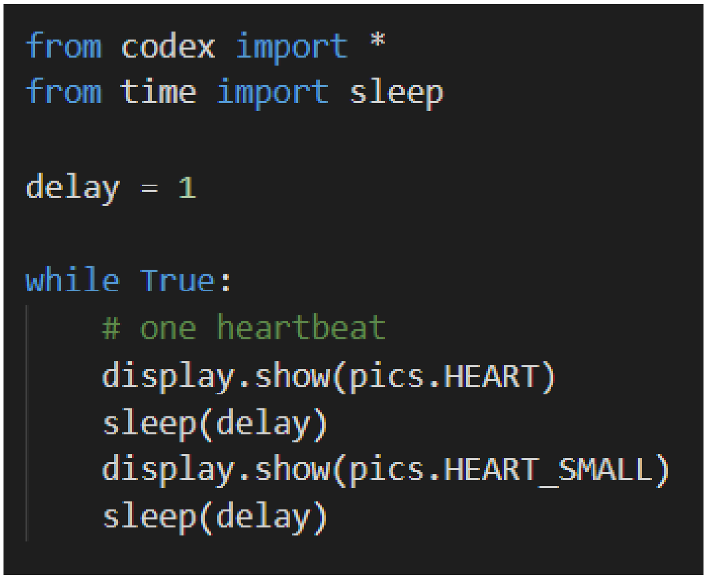
Objective #4: Hearts forever

Modify your code



DO THIS:

- Delete all the repeated code except the first heartbeat
- Add a while loop to the code
  - Type a colon (:) at the end of the line
  - Indent the heartbeat code
  - Use the TAB key to indent your heartbeat code
- Run the code
  - You will need to click “STOP RUNNING” ■ to end the code.



```
from codex import *
from time import sleep

delay = 1

while True:
    # one heartbeat
    display.show(pics.HEART)
    sleep(delay)
    display.show(pics.HEART_SMALL)
    sleep(delay)
```

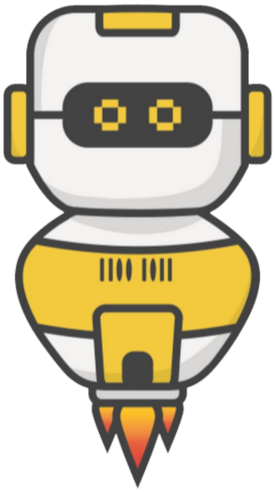
Objective #5: Stop it!

A while loop that goes forever without stopping is an **infinite loop**. This is because the condition is always True.

- Right now, the only way to stop the loop is to click the "**STOP**" button ⏹

The heartbeat changes at the same speed, forever.

- You can change the value of **delay** to change the speed of the heartbeat

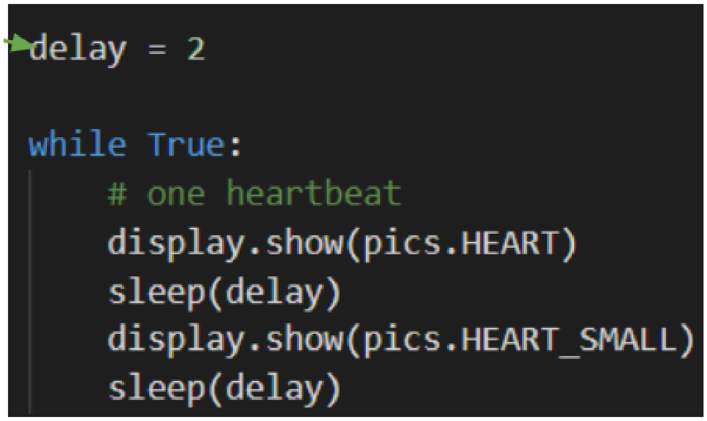


DO THIS:

- Go to your Mission Log and write the definition for **infinite loop**.

Modify your code

- If your code is still running, click "STOP"
- Change the value of **delay** to 2



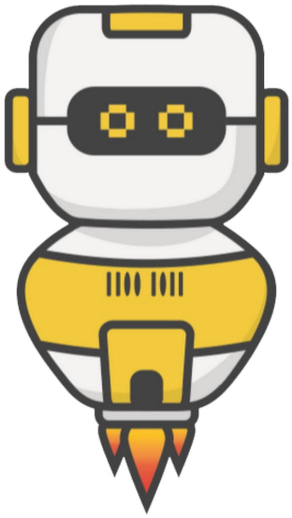
```
delay = 2

while True:
    # one heartbeat
    display.show(pics.HEART)
    sleep(delay)
    display.show(pics.HEART_SMALL)
    sleep(delay)
```

- Run the code
- Click "STOP"
- Change the value of **delay** to 0.5
- Run the code
- Click "STOP"
- OPTIONAL: try different values for **delay**

Objective #6: Heart break

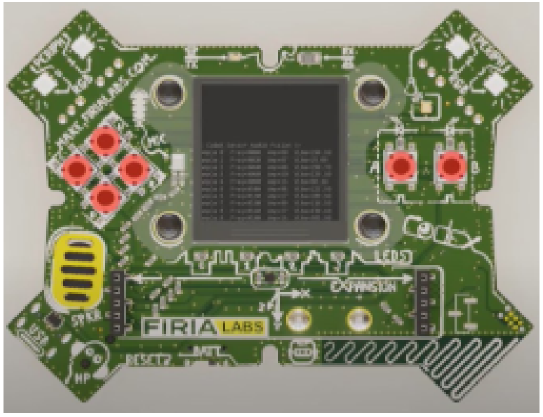
You still have an infinite loop.

- Instead of clicking the “STOP” button, write code to stop the loop
- Use a **break** command
- The **break** command is used in an **if** statement
- Press a button to break out of the loop and stop the program



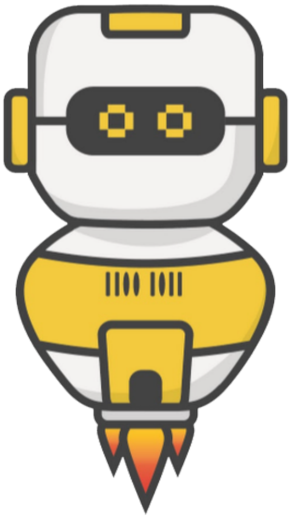
DO THIS:

- Click on CodeX buttons
- Go to your Mission Log and answer the questions for Objective #6



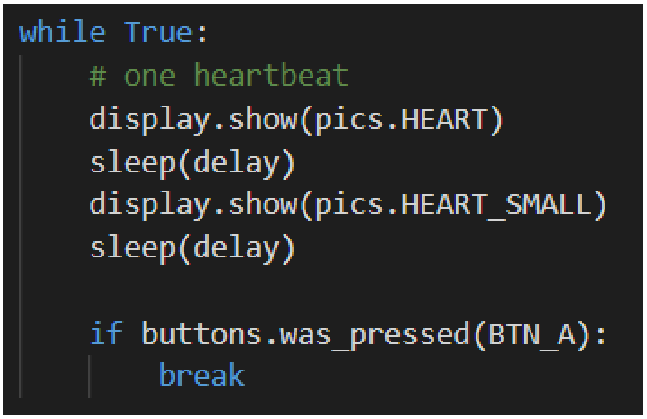
Objective #6: Heart break

Modify your code



DO THIS:

- Add an if statement to the while loop to break out of the loop
  - Be careful with the indenting
  - Make sure to add a colon (:) after the if statement
- Run the code
- Press the “A” button to stop the code



```
while True:
    # one heartbeat
    display.show(pics.HEART)
    sleep(delay)
    display.show(pics.HEART_SMALL)
    sleep(delay)

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

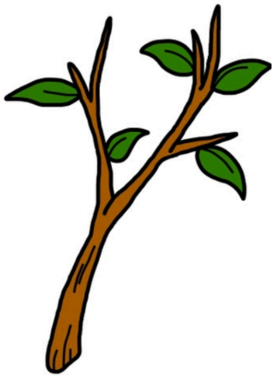
Objective #7: Explore the beat

Now your CodeX is interactive!

And your coding skills are growing.

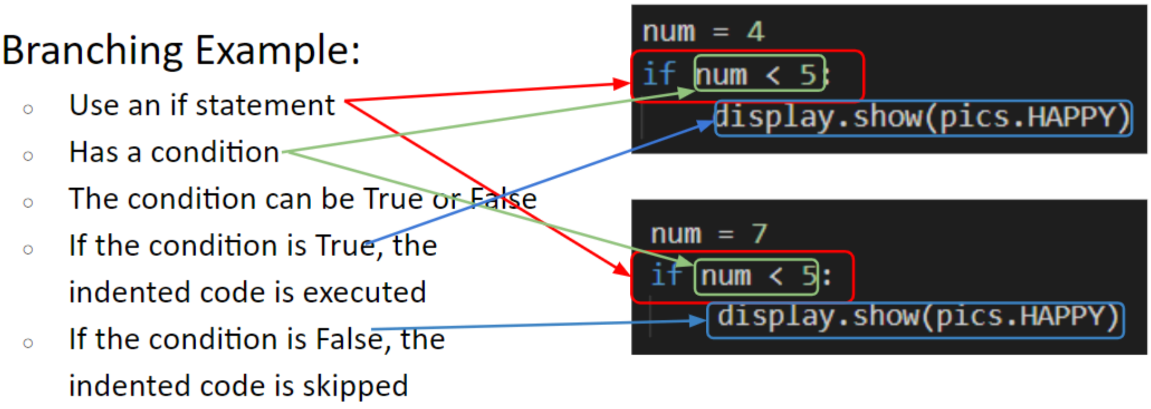
- You learned about
  - Input -- using a button press
  - Branching -- if statements with a condition
- Now you CodeX can do something different when a button is pressed

Review the concepts



- Branching
  - Use an if statement
  - Has a condition
  - The condition can be True or False
  - If the condition is True, the indented code is executed
  - If the condition is False, the indented code is skipped

- Branching Example:

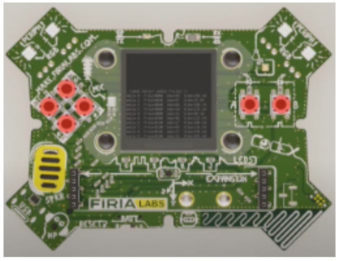


- Use an if statement
- Has a condition
- The condition can be True or False
- If the condition is True, the indented code is executed
- If the condition is False, the indented code is skipped

```
num = 4
if num < 5:
    display.show(pics.HAPPY)
```

```
num = 7
if num < 5:
    display.show(pics.HAPPY)
```

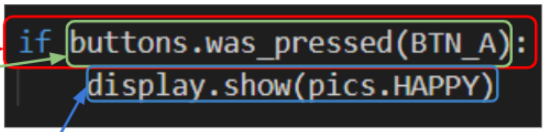
Objective #7: Explore the beat



Review the concepts

- CodeX button input
    - Two different functions check for a CodeX button press
    - **buttons.was_pressed(BTN_A)**
        - Checks to see if BTN_A was pressed since the last check
    - **buttons.is_pressed(BTN_A)**
        - Checks to see if BTN_A is currently pressed
    - Both functions are a condition
        - They evaluate to True or False
    - Both functions can be used in an if statement

- Branching Example with input:
    - Use an if statement
    - Has a condition
    - The condition can be True or False

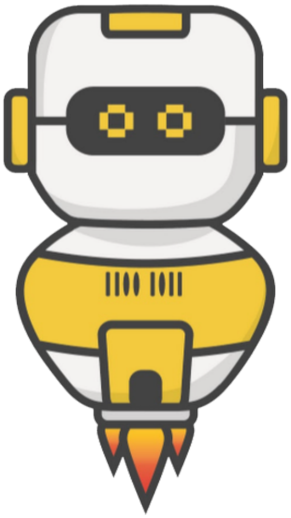


```
if buttons.was_pressed(BTN_A):
    display.show(pics.HAPPY)
```

- If BTN_A was recently pressed, the indented code is executed
- If a different button was recently pressed, the indented code is skipped

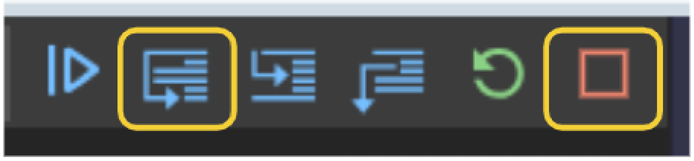
Objective #7: Explore the beat

Experiment with the code



DO THIS:

- Click the debugger button
- Use the Step Over button to watch the branching
- You must step at least 8 times
- Go through the while loop one time
- Then press BTN_A and go through the loop again
- The code should break out of the loop and stop



Step Over

Stop

Mission Quiz: Break-fast

Test your skills by **taking the quiz**.

Objective #8: Half a sleep

You can make the heart beat faster!

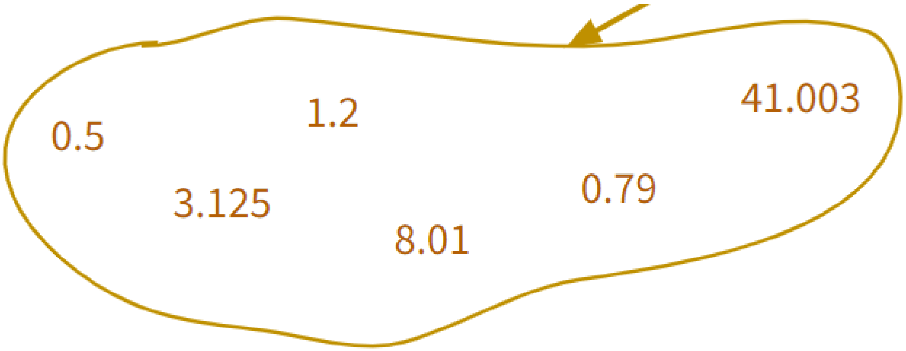
What controls the speed of the beat so far?

- The **delay** variable
- sleep()

To beat faster, the **delay** variable needs to be a smaller value.

But what is smaller than delay = 1?

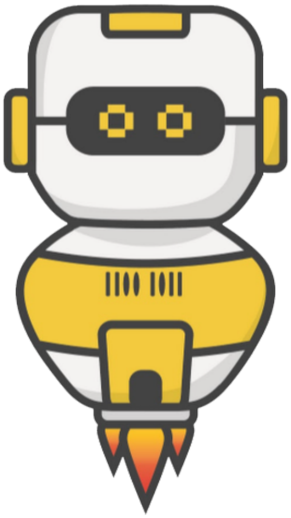
- There isn't an integer less than 1 that can be used for delay
- You need to use a real number, or decimal, like 0.5 or 0.75
- The decimal in the number is called a "floating point"
- The data type for a real number is **float**



Objective #8: Half a sleep

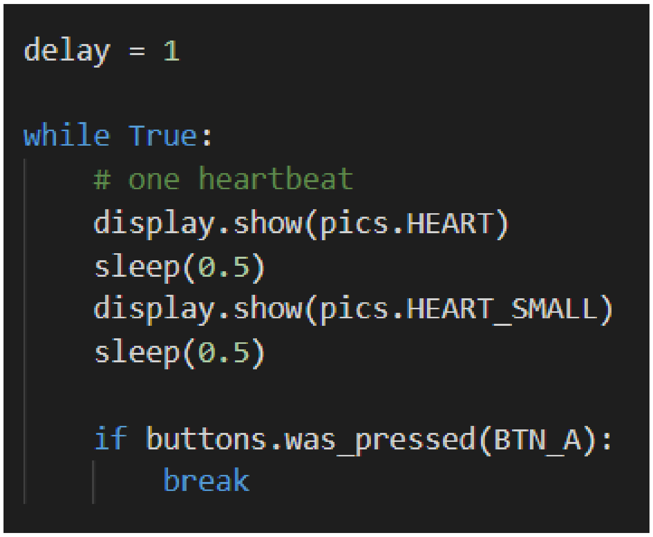
Now you know five data types:

- Integer (Examples: 1, 54, 720)
- CodeX image (Examples: pics.HEART, pics.MUSIC)
- String (Examples: “Hello”, “Press A”, “cake”)
- Boolean (Values: True, False)
- Float (Examples: 0.5, 3.125, 49.02)



DO THIS:

- Set the value of delay to 1
- Use the float value 0.5 in the two sleep() commands
- Run the code
- After a few fast heartbeats, press BTN_A to stop the program

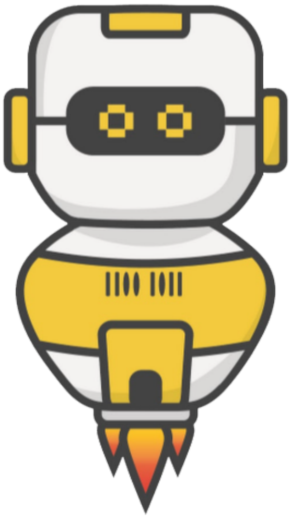


```
delay = 1

while True:
    # one heartbeat
    display.show(pics.HEART)
    sleep(0.5)
    display.show(pics.HEART_SMALL)
    sleep(0.5)

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

Objective #9: Variable speed control

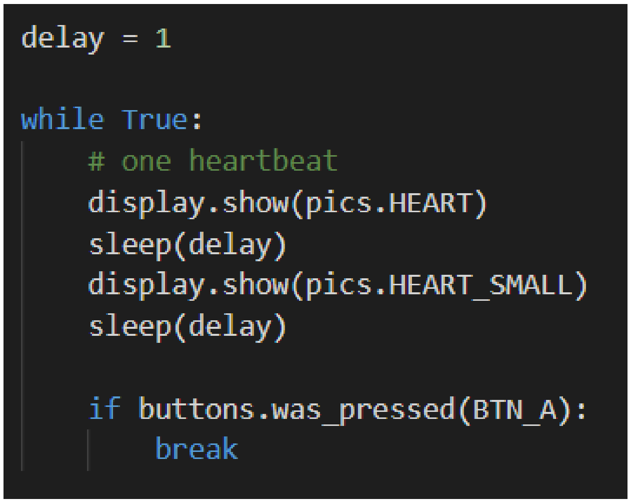
You can change the speed of the heartbeat by changing the value of the variable **delay**.

- You will need your **delay** variable
- You will need to use it in the **sleep()** command



DO THIS:

- Use the delay variable in the 2 sleep() commands
- Run the code
- After a few fast heartbeats, press BTN_A to stop the program



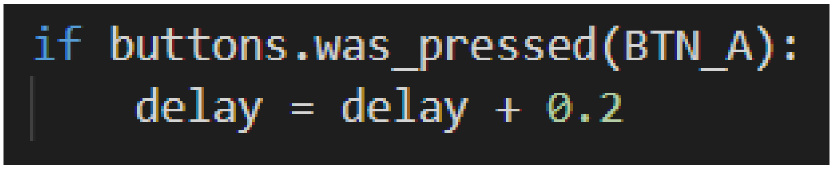
```
delay = 1

while True:
    # one heartbeat
    display.show(pics.HEART)
    sleep(delay)
    display.show(pics.HEART_SMALL)
    sleep(delay)

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

Objective #10: Brake! Not break

With a variable, your heartbeat speed is easy to change.

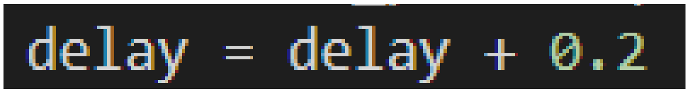
- You can change the speed while the program is running
- Use the buttons to change the value of delay

Look at the code below:



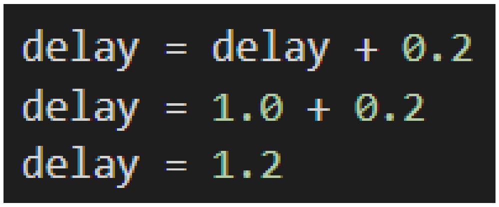
```
if buttons.was_pressed(BTN_A):
    delay = delay + 0.2
```

Look carefully at the indented code:



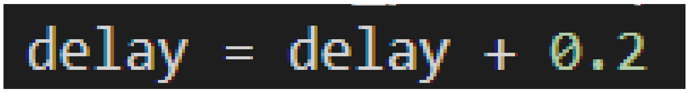
```
delay = delay + 0.2
```

- This doesn't make sense in math, but it does make sense to a computer
- The original value of **delay** is 1, or 1.0
- Then 0.2 is added to the current value
- The new value is assigned to **delay**, like this:



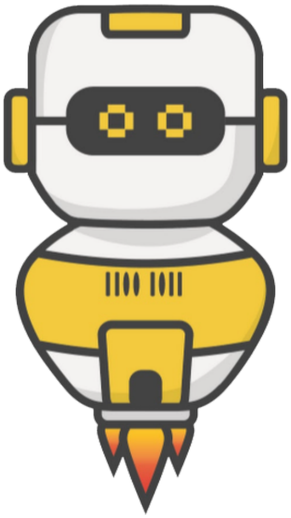
```
delay = delay + 0.2
delay = 1.0 + 0.2
delay = 1.2
```

Objective #10: Brake! Not break



```
delay = delay + 0.2
```

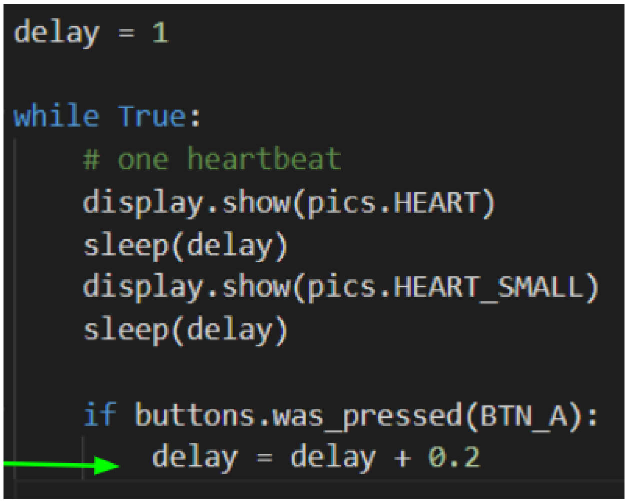
- Adding a set amount, like 0.2, to a variable is called “**increment**”
- In this code, every time BTN_A is pressed, the delay will increase by 0.2
- A larger number for delay will slow down the heartbeat



Ready to try? Your first goal is to slow down the heartbeat.

DO THIS:

- Change the if statement
- Remove “break”
- Replace it by incrementing delay
- Run the code and press BTN_A a few times



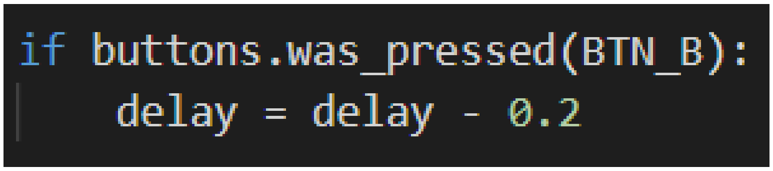
```
delay = 1

while True:
    # one heartbeat
    display.show(pics.HEART)
    sleep(delay)
    display.show(pics.HEART_SMALL)
    sleep(delay)

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

- The heartbeat should slow down each time
- Press the STOP button to stop the program

Objective #11: Variable speed heart

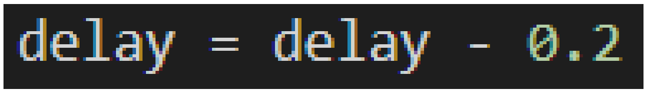
You added code to slow down the heartbeat. Now add code to speed up the heartbeat.

- Use BTN_B
- The code will be almost the same as the if statement for BTN_A



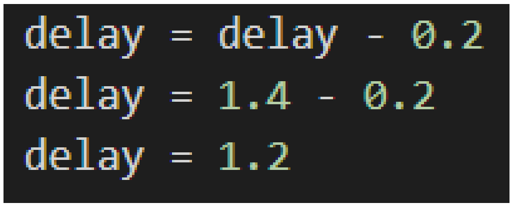
```
if buttons.was_pressed(BTN_B):
    delay = delay - 0.2
```

Look carefully at the indented code:



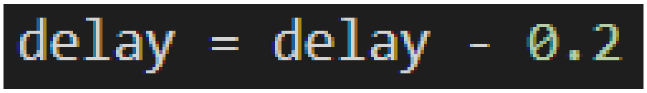
```
delay = delay - 0.2
```

- The original value of **delay** could be 1.0, or 1.2, or 1.4
- In this code, 0.2 is subtracted from the current value
- Then the new value is assigned to **delay**, like this:



```
delay = delay - 0.2
delay = 1.4 - 0.2
delay = 1.2
```

Objective #11: Variable speed heart



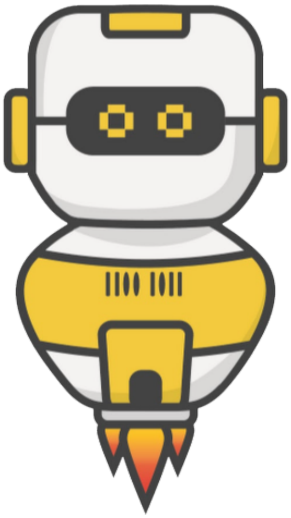
```
delay = delay - 0.2
```

- Subtracting a set amount, like 0.2, to a variable is called “decrement”
- In this code, every time BTN_B is pressed, the delay will decrease by 0.2
- A smaller number for delay will speed up the heartbeat

When you add the if statement, you can speed up and slow down the heartbeat with the press of two buttons

- But … be careful!
- Pressing BTN_B several times can give a 0 or negative value for delay
- The sleep() function must have a positive value!
- So, an error will occur if BTN_B is pressed too many times.

Objective #11: Variable speed heart

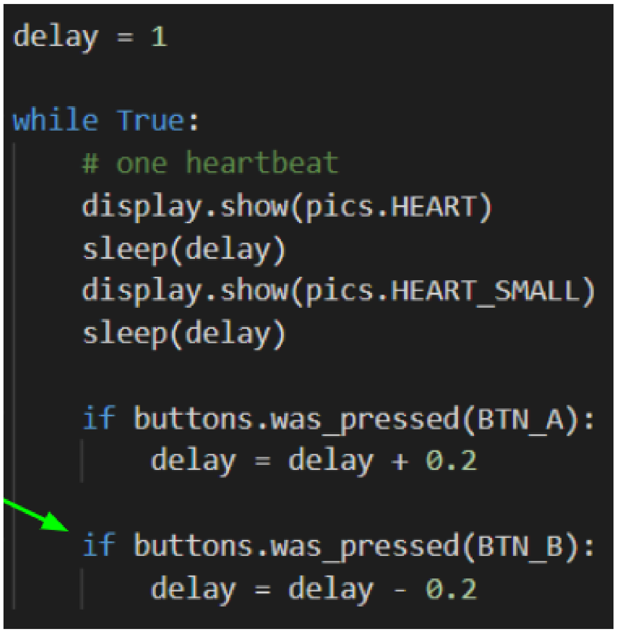


DO THIS:

- Go to your Mission Log and complete the questions for Objective #10 and Objective #11

Complete the second goal to speed up the heartbeat.

- Add an if statement for BTN_B to decrement delay
- Run the code and press BTN_A a few times and BTN_B a few times
- The heartbeat should slow down and speed up
- Press BTN_B enough times to cause an error and stop the program



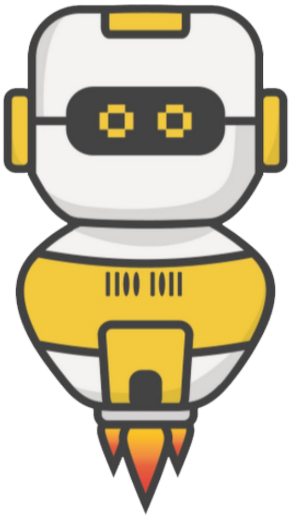
```
delay = 1

while True:
    # one heartbeat
    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):
        delay = delay - 0.2
```

Mission Quiz: Heartfelt Recap

Test your skills by **taking the quiz**.

Mission Complete

You have completed the sixth mission.



Do this:

- Read your "Completed Mission" message
- Complete your Mission 6 Log
  - Post-Mission Reflection
- Get ready for your next mission!

Wait! Before you go ... Clear the CodeX

Go to FILE -- BROWSE FILES

Select the "**Clear**" file and open it

Run the program to clear the CodeX

Okay. Now you can go.