

Warmup: Stairs

Big Board

Square Dance

Safe Pick

Safe Move

Find The End

Two Rows

Week 2

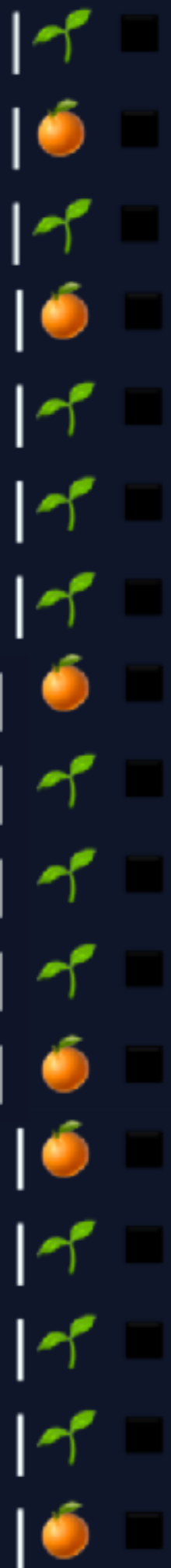
CS106R

Sabri **Eyuboglu** & Geoffrey **Angus**

PyBot

🍊 Collected: 0





?



Loops & Conditional Statements

But first... a **review**.

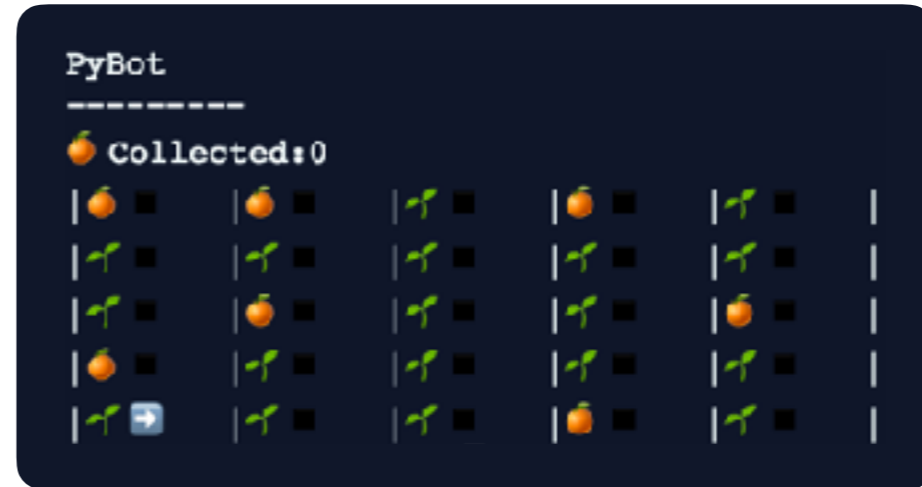
Review

```
1
2
3 def main():
4     ### Start Your Code ###
5     pass # Delete this line when you begin
6
7
8
9     ### End Your Code ###
10
11
12
13 ### Do not edit the code below this line ###
14 if __name__ == "__main__":
15     main()
```

move()

turn_right()

pick_fruit()



The "def" keyword

The function name + "()" + ":"

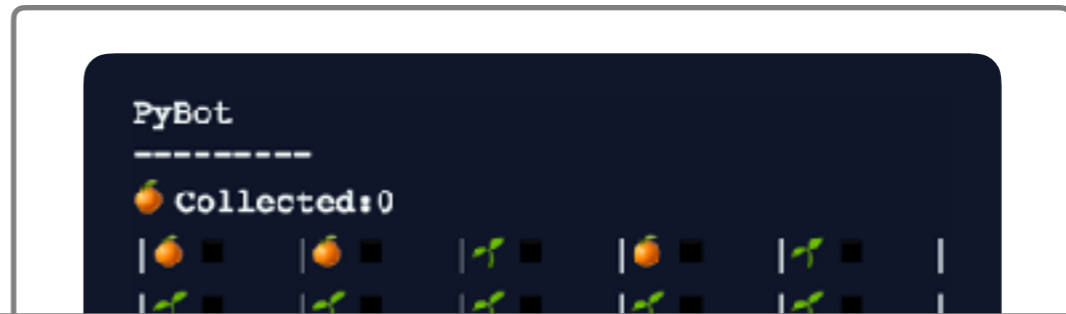
```
def this_is_a_function():
    """
    This is an example function for the class notes.
    """
    if not front_is_blocked():
        move()
    turn_right()
    turn_right()
    move()
    move()
```

The function body

8 lines :

1 line :D

Review



```
1
2
3 def main():
4     ### Start Your Code ###
5     pass # Delete this line when you begin
6
7
8
9     ### End Your Code ###
10
11
12
13 ### Do not edit the code below this line ###
14 if __name__ == "__main__":
15     main()
```

```
if __name__ == '__main__':
    main()
```

Review

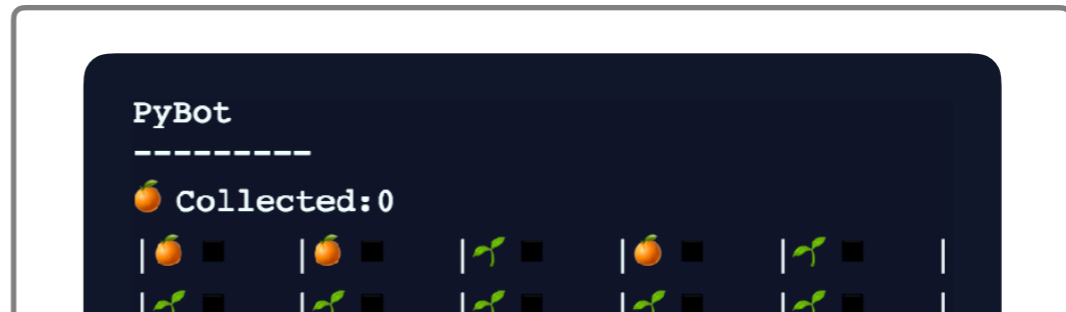
PyBot

🍊 Collected: 0



```
def main():  
    # ...  
    if __name__ == '__main__':  
        main()
```

Review



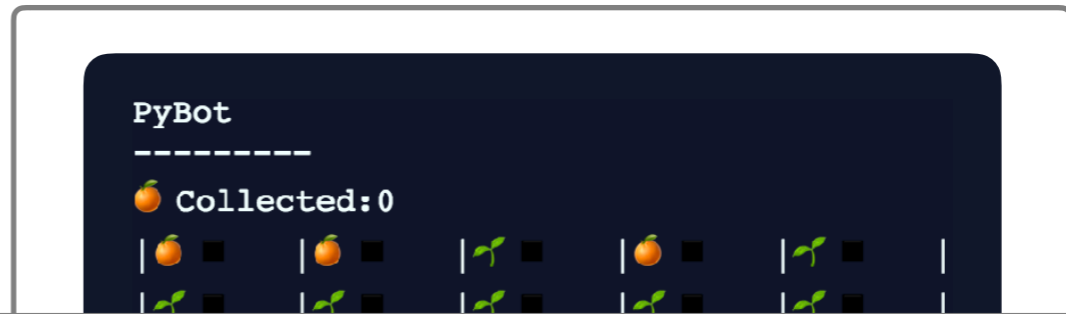
`move()`

`turn_right()`

`pick_fruit()`



Review



The “**def**” keyword

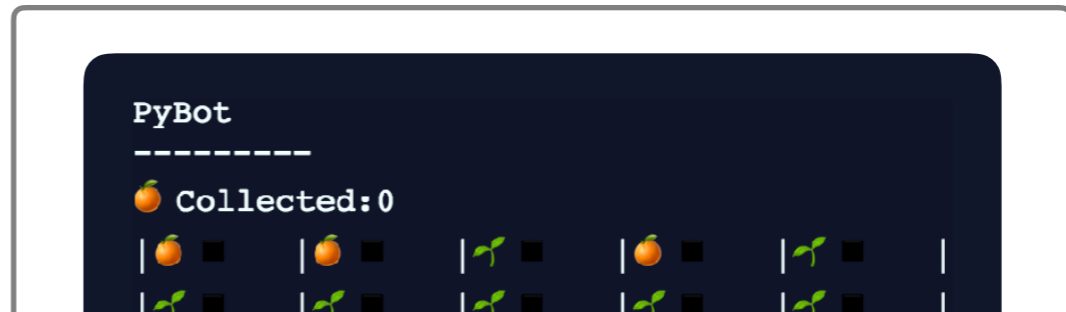
The function **name** + “ () ” + “ : ”

```
def this_is_a_function():  
    """  
    This is an example function for the class notes.  
    """  
    if not front_is_blocked():  
        move()  
    turn_right()  
    turn_right()  
    move()  
    move()
```

The function **body**

```
if __name__ == '__main__':  
    main()
```

Review



8 lines :(

```
def main():  
    pick_fruit()  
    move()  
    pick_fruit()  
    move()  
    pick_fruit()  
    move()  
    pick_fruit()  
    move()  
    turn_right()  
    turn_right()  
    turn_right()  
    pick_fruit()  
    move()  
    pick_fruit()  
    move()  
    pick_fruit()  
    move()  
    turn_right()  
    turn_right()  
    turn_right()  
  
if __name__ == '__main__':  
    main()
```

```
def turn_left():  
    turn_right()  
    turn_right()  
    turn_right()  
  
def pick_and_move():  
    pick_fruit()  
    move()  
  
def pick_fruit_across():  
    pick_and_move()  
    pick_and_move()  
    pick_and_move()  
    pick_and_move()  
  
def main():  
    pick_fruit_across()  
    turn_left()  
    pick_fruit_across()  
    turn_left()  
  
if __name__ == '__main__':  
    main()
```

1 line :D

Review

```
1
2
3 def main():
4     ### Start Your Code ###
5     pass # Delete this line when you begin
6
7
8
9     ### End Your Code ###
10
11
12
13 ### Do not edit the code below this line ###
14 if __name__ == "__main__":
15     main()
```

move()

turn_right()

pick_fruit()



The "def" keyword

The function name + "()" + ":"

```
def this_is_a_function():
    """
    This is an example function for the class notes.
    """
    if not front_is_blocked():
        move()
        turn_right()
        turn_right()
        move()
        move()
```

The function body

8 lines :

```
def main():
    pick_fruit()
    move()
    pick_fruit()
    move()
    pick_fruit()
    move()
    turn_right()
    turn_right()
    turn_right()
    pick_fruit()
    move()
    pick_fruit()
    move()
    pick_fruit()
    move()
    turn_right()
    turn_right()
    turn_right()
    if __name__ == '__main__':
        main()
```

```
def turn_left():
    turn_right()
    turn_right()
    turn_right()

def pick_and_move():
    pick_fruit()
    move()

def pick_fruit_across():
    pick_and_move()
    pick_and_move()
    pick_and_move()
    pick_and_move()

def main():
    pick_fruit_across()
    turn_left()
    pick_fruit_across()
    turn_left()

if __name__ == '__main__':
    main()
```

1 line :D

PyBot

🍊 Collected: 0



PyBot

🍊 Collected: 0

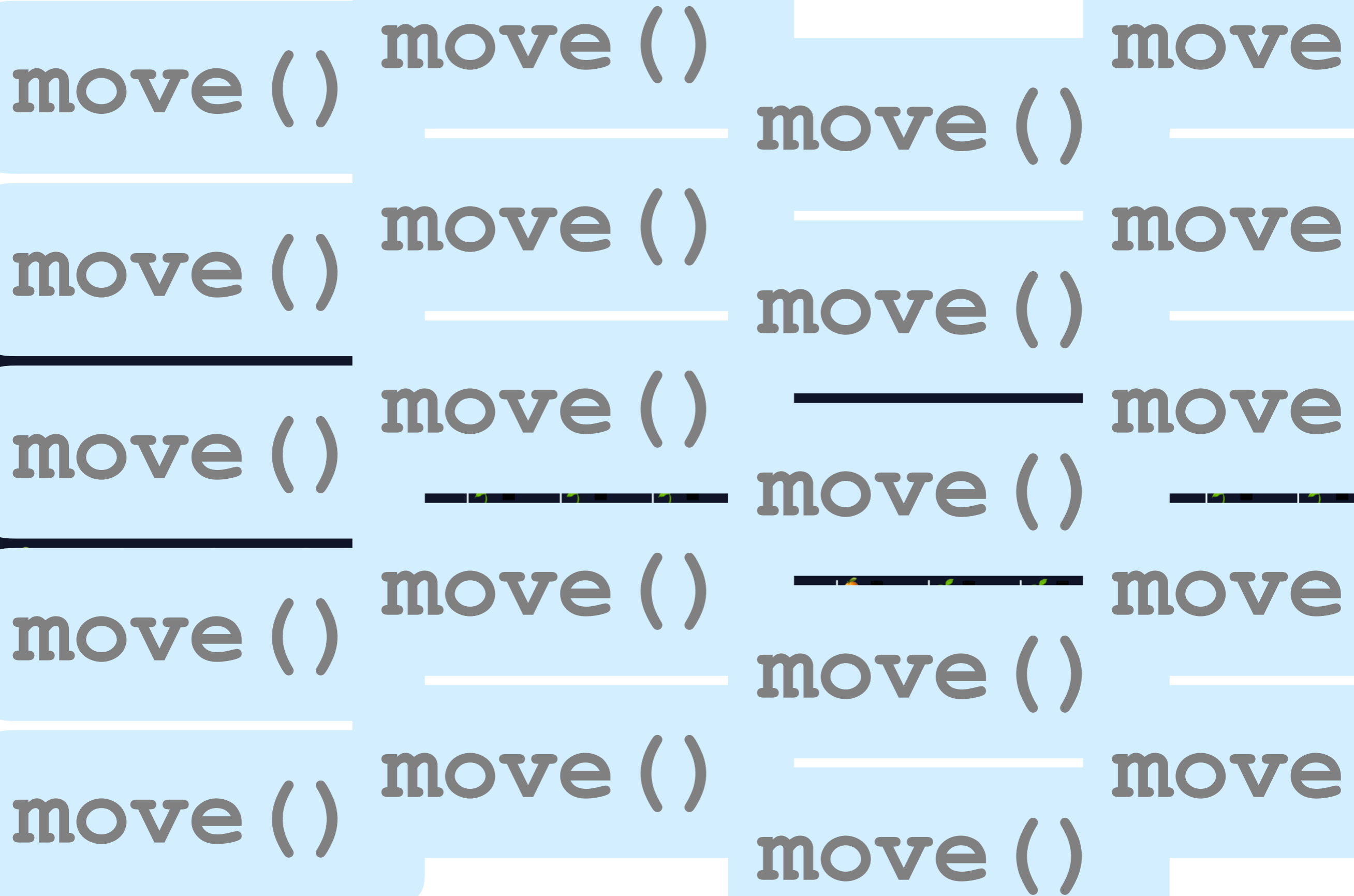


????

PyBot

🍊 Collected: 0





Or...

For Loops

```
def move_across_99():  
    for i in range(99):  
        move()
```


Example: Far Away Fruit

Today's Exercises

Big Field

Square Dance

Safe Pick

Safe Move

Find The End

Two Rows

Today's Exercises

Big Field

Dança do Quadrado

Rogue Fruit

Face North

Land's End

Change Row

`if` Statements

Introducing **GeoffBot...**

GeoffBot Action Functions

`go_to_store()`

`buy_coca()`

`buy_guarana()`

`buy_sprite()`

If Statements

“GeoffBot, go to the store.

If they have Guarana, buy some.”

If Statements

“GeoffBot, go to the store.

If they have Guarana, buy some.”

```
go_to_store()
```

```
buy_guarana()
```

If Statements

“GeoffBot, go to the store.

If they have Guarana, buy some.”



GeoffBot Conditional Functions

```
has_coca()
```

```
has_sprite()
```

```
has_guarana()
```


If Statements

“GeoffBot, go to the store.

If they have Guarana, buy some.”



If Statements

“GeoffBot, go to the store.

If they have Guarana, buy some.”

```
go_to_store()
```

```
if has_guarana() :
```

```
    buy_guarana()
```

If Statements

“GeoffBot, go to the store.

If they have Guarana, buy some.”

```
go_to_store()
```

```
if has_guarana() :
```

```
    buy_guarana()
```

If Statements



If Statements

```
PyBot
-----
🍊 Collected:0
| ? ■   | ? ■   | ? ■   | ? ■   | ? ■   |
| ? ■   | ? ■   | ? ■   | ? ■   | ? ■   |
| ? ■   | ? ■   | ? ■   | ? ■   | ? ■   |
| ? ■   | ? ■   | ? ■   | ? ■   | ? ■   |
| ? →   | ? ■   | ? ■   | ? ■   | ? ■   |
Hit ENTER to start: █
```

New PyBot Function

`has_fruit()`

Returns **True** if PyBot's current cell has an orange.



FALSE



TRUE

New PyBot Function

```
if has_fruit():  
    pick_fruit()
```

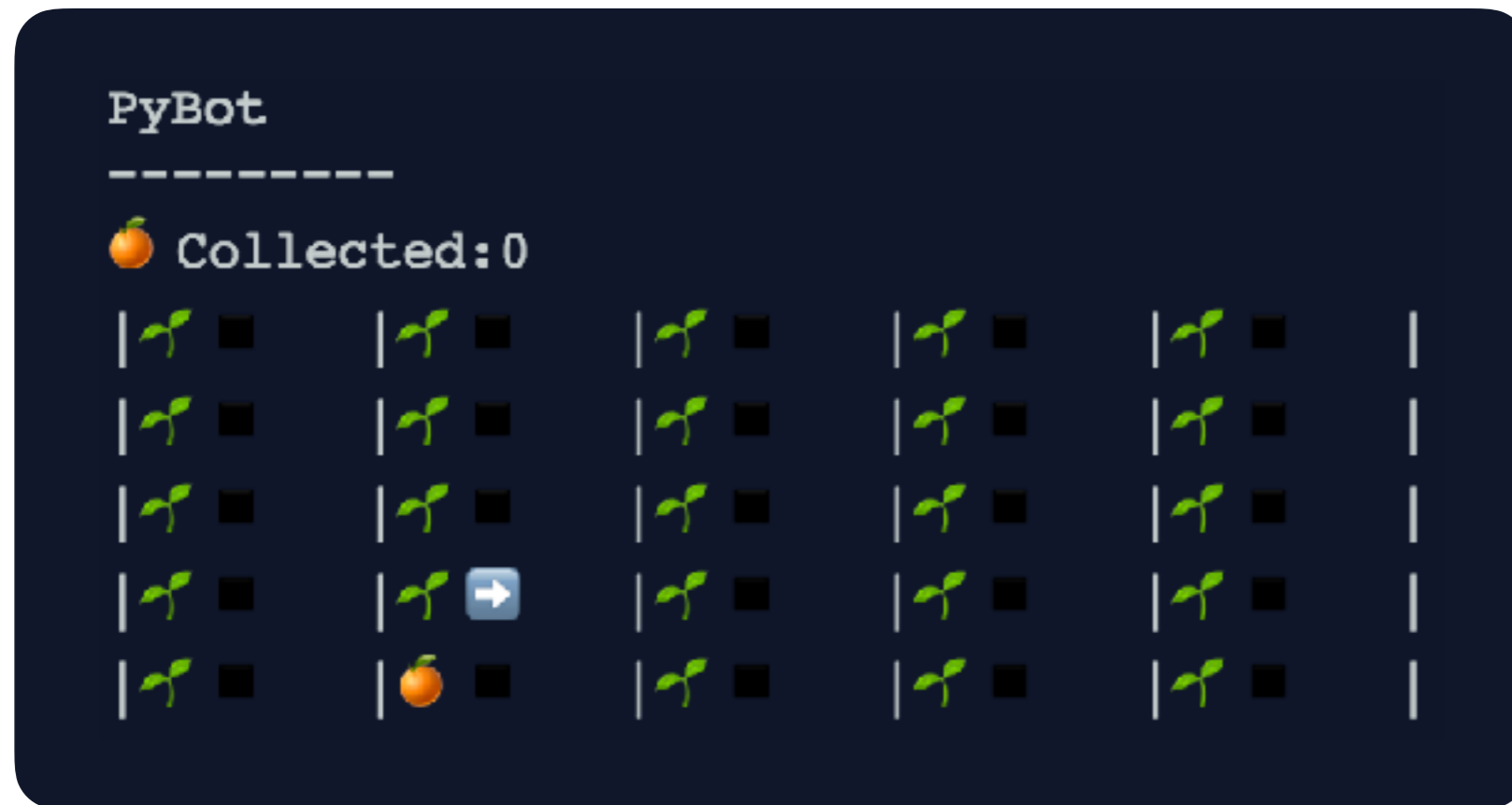
New PyBot Function



```
if True has_fruit():  
    pick_fruit()
```

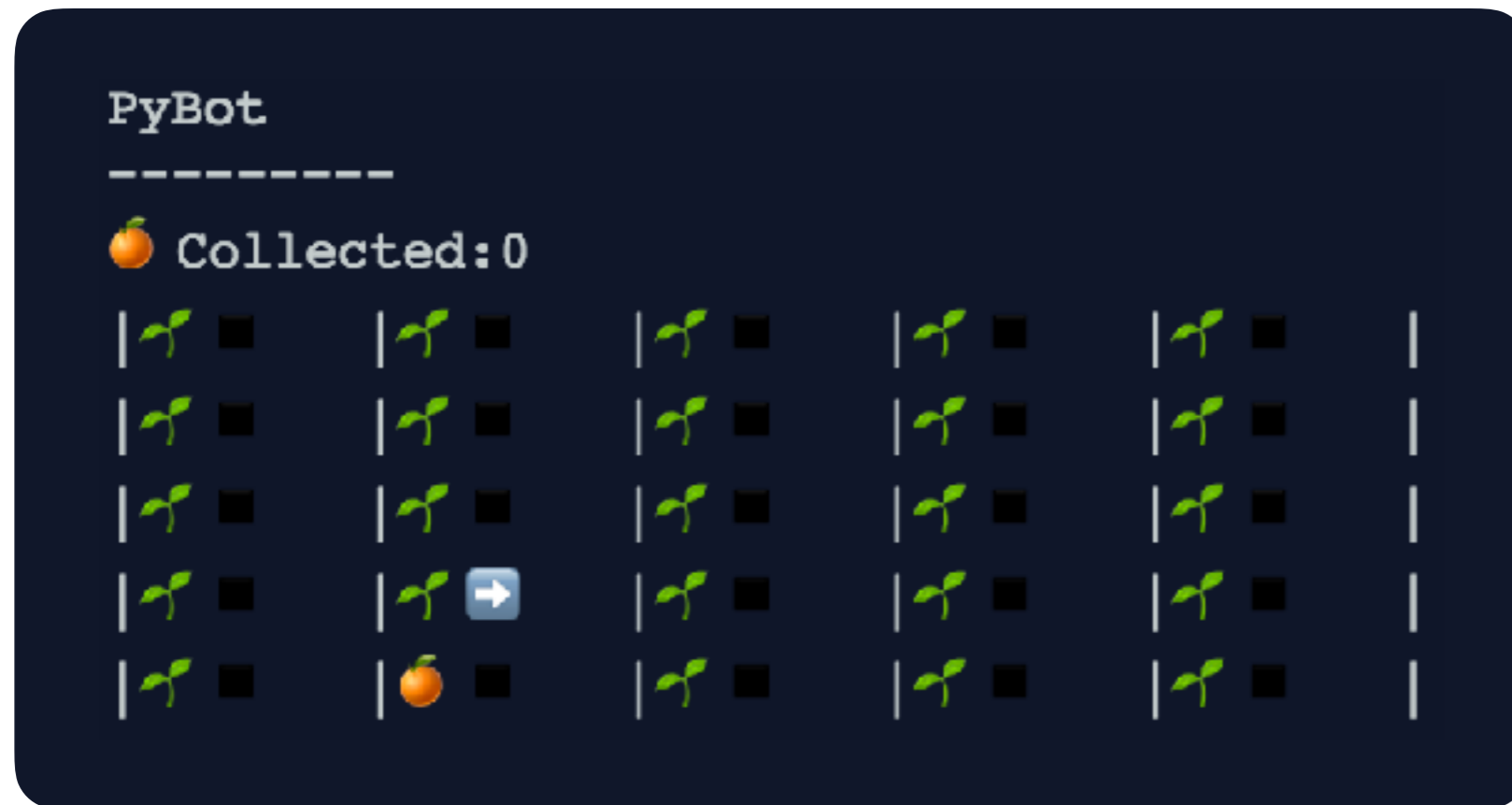


New PyBot Function



```
if False has_fruit():  
    pick_fruit()
```

New PyBot Function



```
if False has_fruit():  
    pick_fruit()
```

If Statements

```
(1) if (2) condition_function() (3) :  
    action_function()  
    action_function()  
    ...  
    (4)
```

Example: Safe Pick

Today's Exercises

Big Board

Square Dance

Rogue Fruit

Face North

Board's Edge

Two Rows

Conditional PyBot Functions

`is_facing_north()`

Returns **True** if PyBot is facing north.



FALSE



TRUE

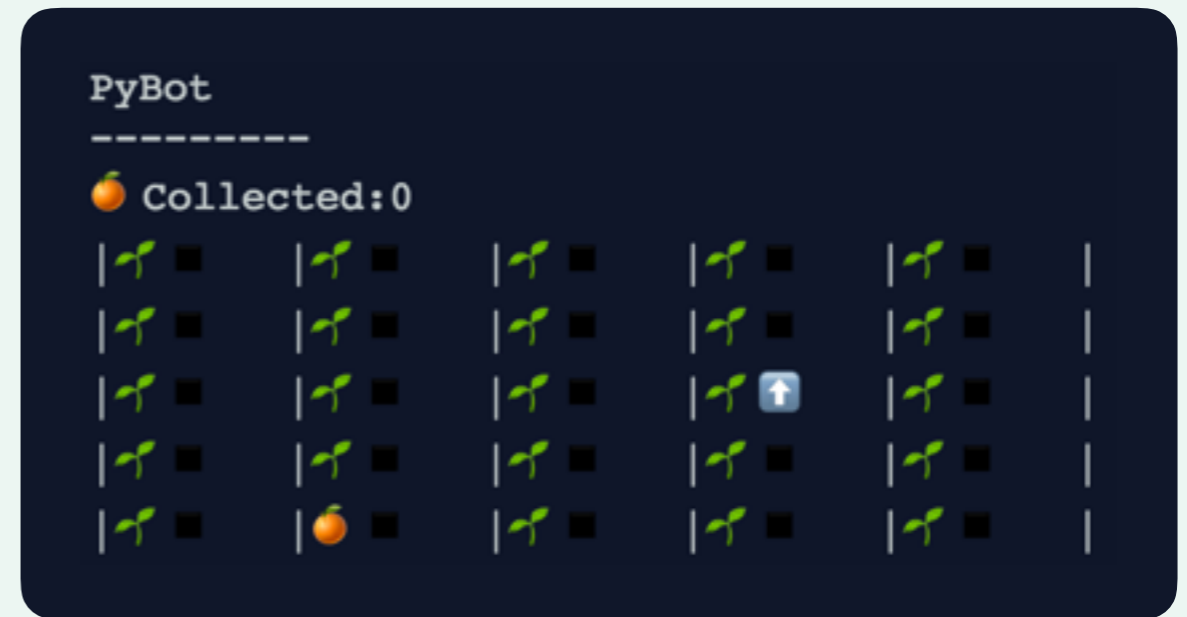
Conditional PyBot Functions

`is_facing_north()`

Returns **True** if PyBot is facing north.



FALSE



TRUE

`is_facing_east()`

`is_facing_south()`

`is_facing_west()`

What if instead of **skipping** instructions, we want PyBot to do something **totally different**?

`else` and `elif`
 (“else if”)

Else and Elif Statements

Let's bring back **GeoffBot...**

Else and Elif Statements

“GeoffBot, go to the store.

If they have Guarana, buy some.

Otherwise buy me some Coca-Cole.”

Else and Elif Statements

“GeoffBot, go to the store.

If they have Guarana, buy some.

Otherwise buy me some Coca-Cole.”

```
go_to_store()
```

```
if has_guarana() :
```

```
    buy_guarana()
```

```
else:
```

```
    buy_coca()
```

Definition

Pseudocode - *Code that looks like English (or Portuguese) that is structured like a program and uses programming vocabulary.*

Else and Elif Statements

“GeoffBot, go to the store.

If they have Guarana, buy some.

Otherwise, if they have Sprite,
buy me some. Otherwise buy
me some Coca-Cola.”

E Elif Statements

“GeoffBot, go to the store.

If they have Guarana, buy some. Otherwise buy me some Coca-Cole.”

```
go_to_store()
```

```
if has_guarana() :
```

```
    buy_guarana()
```

```
elif has_sprite() :
```

```
    buy_sprite()
```

```
else:
```

```
    buy_coca()
```

Today's Exercises

Big Board

Square Dance

Safe Pick

Move North

Find The End

Two Rows

While Loops

PyBot

🍊 Collected: 0



PyBot

🍊 Collected: 0

🍊	■	🍊	■	🌱	■	🍊	■	🌱	■	🌱	■	🍊	■	🌱	■	🌱	■	🌱	■
🌱	■	🌱	■	🌱	■	🌱	■	🌱	■	🌱	■	🌱	■	🌱	■	🌱	■	🌱	■
🌱	■	🍊	■	🌱	■	🌱	■	🍊	■	🌱	■	🍊	■	🌱	■	🌱	■	🌱	■
🍊	■	🌱	■	🌱	■	🌱	■	🌱	■	🌱	■	🌱	■	🌱	■	🌱	■	🌱	■
🌱	➡	🌱	■	🌱	■	🍊	■	🌱	■	🌱	■	🍊	■	🌱	■	🌱	■	🌱	■

Conditional PyBot Functions

`front_is_blocked()`

Returns **True** if PyBot is facing a wall.



FALSE



TRUE

```
def move_across_variable():  
    while not front_is_blocked():  
        move()
```

PyBot

🍊 Collected: 0

```
| 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ |  
| 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ |  
| 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ |  
| 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ |  
| 🌱 ➡ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🍊 ■ |
```

Hit ENTER to start: █

```
def move_across_variable():  
    while not front_is_blocked():  
        move()
```

PyBot

🍊 Collected: 0

```
| 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ |  
| 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ |  
| 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ |  
| 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ |  
| 🌱 ➡ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🍊 ■ |
```

Hit ENTER to start: █

```
def move_across_variable(): False  
while not front_is_blocked():  
    move()
```

PyBot

🍊 Collected: 0

```
| 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ |  
| 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ |  
| 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ |  
| 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ |  
| 🌱 ➡ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🍊 ■ |
```

Hit ENTER to start: █

```
def move_across_variable(): False  
while not front_is_blocked():  
    move()
```



```
def move_across_variable(): False
while not front_is_blocked():
    move()
```

is the same thing as...

```
def move_across_variable(): True
while not front_is_blocked():
    move()
```

PyBot

🍊 Collected: 0

```
| 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ |  
| 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ |  
| 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ |  
| 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🌱 ■ |  
| 🌱 ➡ | 🌱 ■ | 🌱 ■ | 🌱 ■ | 🍊 ■ |
```

Hit ENTER to start: █

```
def move_across_variable(): False  
while not front_is_blocked():  
    move()
```

PyBot

🍊 Collected:0



Hit ENTER to start: █

```
def move_across_variable(): True
while not front_is_blocked():
    move()
```

PyBot

🍊 Collected:0



Hit ENTER to start: █

```
def move_across_variable(): False
while not front_is_blocked():
    move()
```

Today's Exercises

Big Board

Square Dance

Safe Pick

Safe Move

Find The End

Two Rows

Today's Exercises

Big Board

Square Dance

Safe Pick

Safe Move

Find The End

Two Rows

Recap

repl.it = Where we will be coding.

PyBot = Your new best friend. Learn her set of commands!

Functions are little packages of code.

Implement functions to *decompose* and *make your life easier*.

