

## Customize Your Turtle

---

### Textbook

---

## Customize Your Turtle



The Python Turtles library has some fun ways you can customize your turtle. Let's explore a few of them in this lesson.

### Change Turtle Size

You can adjust the size of your turtle by using the following code command.

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.shapesize(10,1,5)
```

The 3 items inside the parentheses are called [arguments](#). Each argument affects a different aspect of the turtle.

1. The first argument affects the stretch length.

2. The second argument affects the stretch width.
3. The third argument affects the outline width.

Try adjusting these values to make different sizes of turtles!

## Change Turtle Color



Make your turtle bigger before changing your turtle colors so that you can see the changes.

To change the outline of the turtle use the following code.

```
1 turtle.pencolor("red")
```

To change the fill color of the turtle use the following code.

```
1 turtle.fillcolor("blue")
```

You can add any color inside the parentheses here as well as hexadecimal values.

If you want to change both the color and the outline, you can do it on one line of code like this.

```
1 turtle.color("red", "blue")
```

1. The first argument will be the outline of the turtle.

2. The second argument will be the fill color.

## Different Turtle Shapes

You can also change the shape of your turtle. Use the following code.

```
1 turtle.shape("turtle")
```

This will change the shape of the turtle from an arrow to a turtle shape.

Try the following shapes out!

**square, arrow, circle, turtle, triangle, classic**

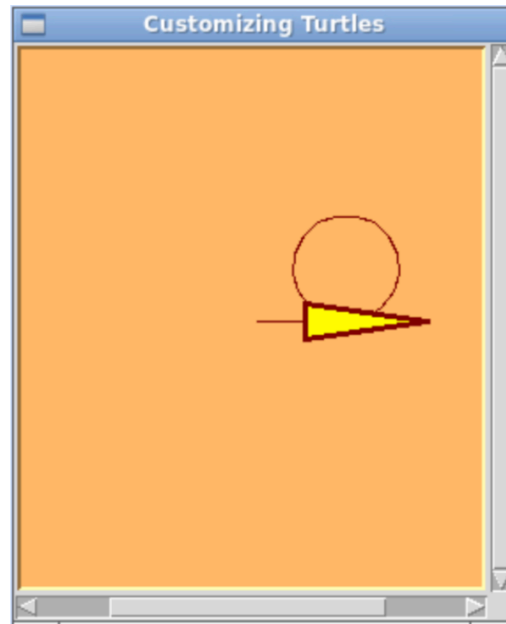
## Checkpoint

---

### Customize Your Turtle

Practice customizing your turtle.

1. Include the necessary code to start up a Python screen (import the library and generate a screen).
2. Add a background color with a hexadecimal value.
3. Add a custom title to your turtle screen.
4. Change the shape of your turtle to a `triangle`.
5. Change the size of your turtle to have the values of `1,4,3`.
6. Change the `pencolor` to `maroon`.
7. Change the `fillcolor` to `yellow`. *(For this challenge, do this and step 6 by using separate lines of code.)*
8. Move the turtle forward `50`.
9. Make a circle with a radius of `30`.



## Requirements:

- Include the necessary code to start up a Python screen (import the library and generate a screen).
- Add a background color with a hexadecimal value.
- Add a custom title to your turtle screen.
- Change the shape of your turtle to a `triangle`.
- Change the size of your turtle to have the values of `1,4,3`.
- Change the `pencolor` to `maroon`.
- Change the `fillcolor` to `yellow`. *(For this challenge, do this and step 6 by using separate lines of code.)*
- Move the turtle forward `50`.
- Make a circle with a radius of `30`.

## Questions (10)

### 1. How can you adjust the size of your turtle in the Python Turtles library?

MULTIPLE CHOICE

Choose the correct answer:

- A. `turtle.shape()`
- B. `turtle.getscreen()`
- C. `turtle.shapesize()`
- D. `turtle.pencolor()`

## 2. In the code below, what does the second argument inside the parentheses affect?

MULTIPLE CHOICE

Example:

```
1 turtle.shapesize(10,1,5)
```

```
turtle.shapesize(10,1,5)
```

Choose the correct answer:

- A. Stretch length.
- B. Stretch width.
- C. Outline width.
- D. Fill color.

## 3. What does the following code do?

MULTIPLE CHOICE

```
turtle.pencolor("red")
```

Choose the correct answer:

- A. Changes the entire turtle to red.
- B. Changes the outline color of the turtle.
- C. Changes the fill color of the turtle.
- D. Adjusts the turtle's size.

## 4. How can you change both the color and the outline of the turtle in one line of code?

MULTIPLE CHOICE

Choose the correct answer:

- A. `turtle.pencolor("red")`
- B. `turtle.fillcolor("blue")`
- C. `turtle.color("red", "blue")`
- D. `turtle.shape("turtle")`

**5. In the following code, what part of the turtle will be red?**

```
turtle.color("red", "blue")
```

**Choose the correct answer:**

- A. The outline of the turtle.
- B. The fill color of the turtle.
- C. The entire turtle.
- D. None of the turtle.

**6. What does the following code do?**

```
turtle.shape("turtle")
```

**Choose the correct answer:**

- A. Changes the turtle to a turtle shape.
- B. Adjusts the turtle's size.
- C. Changes the fill color of the turtle.
- D. Changes the outline color of the turtle.

**7. To make different sizes of turtles, which command should you use?****Choose the correct answer:**

- A. `turtle.size()`
- B. `turtle.shapesize()`
- C. `turtle.widthlength()`
- D. `turtle.scale()`

**8. What type of values can be added inside the parentheses of the following code?**

```
turtle.fillcolor()
```

**Choose the correct answer:**

- A. Only numerical values.
- B. Only hexadecimal values.
- C. Only color names.
- D. Both color names and hexadecimal values.

### 9. Debug the following code:

[DEBUG CODE](#)

#### Code to Debug:

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.shape(circle)
```

### 10. Debug the following code:

[DEBUG CODE](#)

#### Code to Debug:

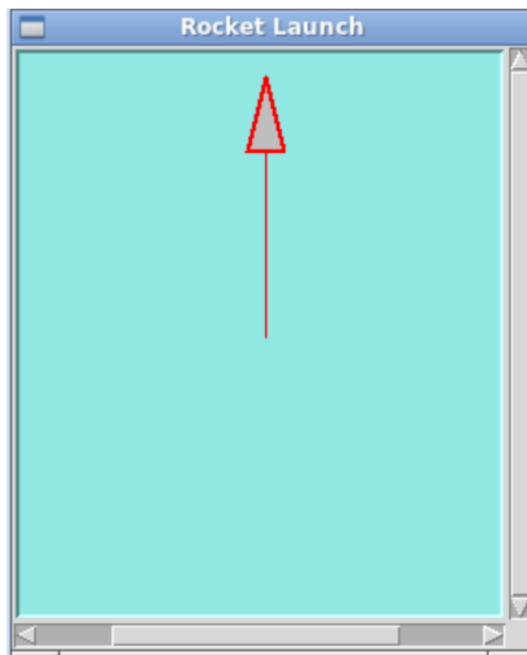
```
1 import turtle
2 turtle.getscreen()
3
4 turtle.color("red" "blue")
```

## Challenges (5)

### 1. Rocket Launch

Make a rocket launch!

1. Include the necessary code to start up a Python screen (Import the library and generate a screen. Create a Python turtle variable as well).
2. Add a custom background color.
3. Add a custom title to your turtle screen.
4. Change the shape of your turtle to an `arrow`.
5. Change the size of your turtle to have the values of `1,4,2`.
6. Change the pencolor and the fillcolor. *(For this challenge, do both of these on the same line of code.)*
7. Rotate your variable to the left 90 degrees and forward 100.



#### Requirements:

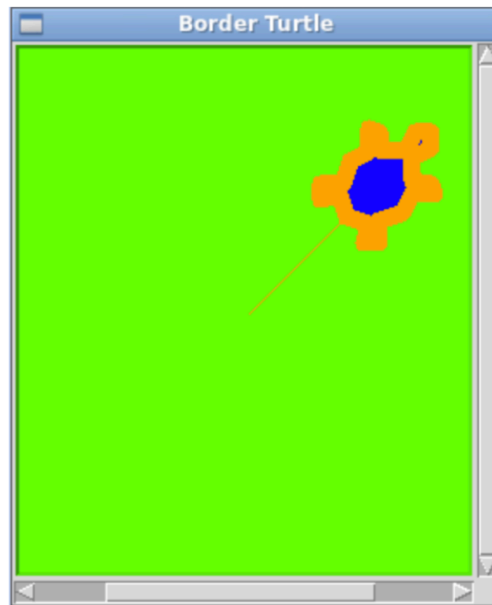
- Include the necessary code to start up a Python screen (Import the library and generate a screen).
- Add a custom background color.
- Add a custom title to your turtle screen.
- Change the shape of your turtle to an `arrow`.
- Change the size of your turtle to have the values of `1,4,2`.
- Change the pencolor and the fillcolor. *(For this challenge, do both of these on the same line of code.)*
- Rotate your variable to the left 90 degrees and forward 100.



## 2. Turtle Outline

Create a turtle with a super thick outline.

1. Include the necessary code to start up a Python screen (import the library and generate a screen).
2. Add a custom background color.
3. Add a custom title to your turtle screen.
4. Change the shape of your turtle to a `turtle` .
5. Change the size of your turtle to have the values of `3,3,10` .



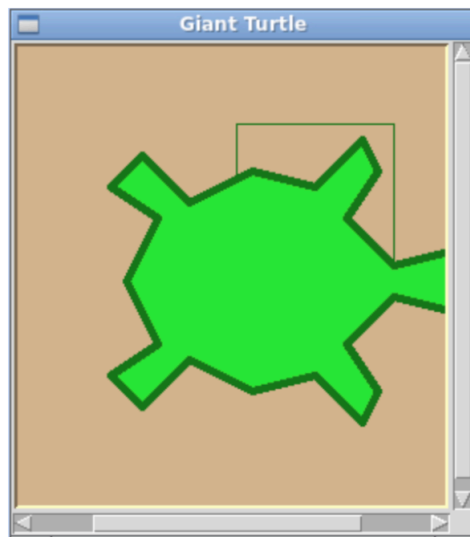
### Requirements:

- Include the necessary code to start up a Python screen (import the library and generate a screen).
- Add a custom background color.
- Add a custom title to your turtle screen.
- Change the shape of your turtle to a `turtle` .
- Change the size of your turtle to have the values of `3,3,10` .

### 3. Giant Turtle

Make a gigantic turtle.

1. Include the necessary code to start up a Python screen (import the library and generate a screen).
2. Add a custom background color.
3. Add a custom title to your turtle screen.
4. Change the shape of your turtle to a `turtle` .
5. Change the size of your turtle to have the values of `10,10,5` .
6. Change the pencolor to a dark green color and the fillcolor to a light green color. *(For this challenge, do both of these on the same line of code.)*
7. Draw a square with your turtle. Include 4 forward movements and 4 left turns of 90 degrees.



#### Requirements:

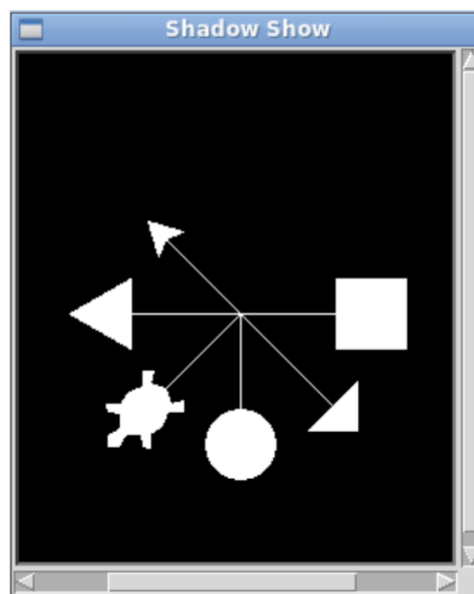
- Include the necessary code to start up a Python screen (import the library and generate a screen).
- Add a custom background color.
- Add a custom title to your turtle screen.
- Change the shape of your turtle to a `turtle` .
- Change the size of your turtle to have the values of `10,10,5` .
- Change the pencolor to a dark green color and the fillcolor to a light green color. *(For this challenge, do both of these on the same line of code.)*
- Draw a square with your turtle. Include 4 forward movements and 4 left turns of 90 degrees.

#### 4. Shadow Show

Show off all the turtle shapes! Draw in white on a black background.

1. Include the necessary code to start up a Python screen (import the library and generate a screen).
2. Create six Python turtle variables named `first` , `second` , `third` , `fourth` , `fifth` , `sixth` .
3. Add a background color with a value of `black` .
4. Add a custom title to your turtle screen.
5. Assign each variable to a different shape. Make sure to use all the shapes. `square` , `arrow` , `circle` , `turtle` , `triangle` , `classic` .
6. Give each variable a shapesize of `2,2,1` .
7. Make each variable a color of `white` fill and a `white` border/pen color.
8. Rotate each variable a different amount. Move each variable forward `75` .

*Note: Each `variable.forward(75)` will need to be on the line below `variable.left()` or `variable.right()`*



#### Requirements:

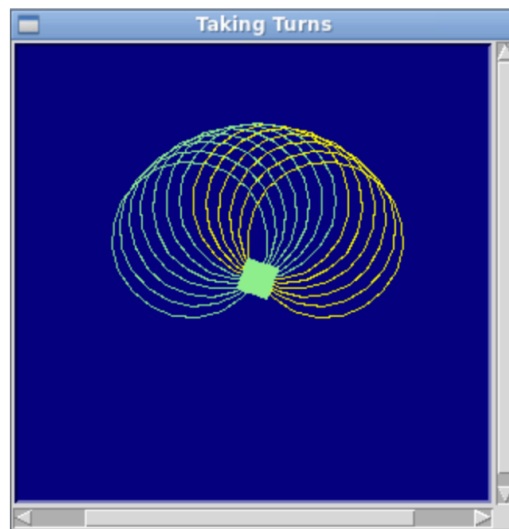
- Include the necessary code to start up a Python screen (import the library and generate a screen).
- Create six Python turtle variables named `first` , `second` , `third` , `fourth` , `fifth` , `sixth` .
- Add a background color with a value of `black` .
- Add a custom title to your turtle screen.
- Assign each variable to a different shape. Make sure to use all the shapes. `square` , `arrow` , `circle` , `turtle` , `triangle` , `classic` .
- Give each variable a shapesize of `2,2,1` .

- Make each variable a color of `white` fill and a `white` border/pen color.
- Rotate each variable a different amount. Move each variable forward `75` .

## 5. Taking Turns

Create a cool design that takes turns from one turtle to the next.

1. Include the necessary code to start up a Python screen (import the library and generate a screen).
2. Create a Python turtle variable named `first` and another variable named `second`.
3. Add a background color with a value of `navy`.
4. Add a custom title to your turtle screen.
5. Change the shape of the turtle named `first` to a `circle`.
6. Change the shape of the turtle named `second` to a `square`.
7. Change the `shapesize` of the turtle named `first` to `1,1,1`.
8. Change the `shapesize` of the turtle named `second` to `1,1,1`.
9. Move the turtle named `first` in a circle with a radius of 50, then rotate it to the right 10 degrees.
10. Move the turtle named `second` in a circle with a radius of 50, then rotate it to the left 10 degrees.
11. Repeat steps 9-10 seven times.



### Requirements:

- Include the necessary code to start up a Python screen (import the library and generate a screen).
- Create a Python turtle variable named `first` and another variable named `second`.
- Add a background color with a value of `navy`.
- Add a custom title to your turtle screen.
- Change the shape of the turtle named `first` to a `circle`.
- Change the shape of the turtle named `second` to a `square`.

- Change the `shapesize` of the turtle named `first` to `1,1,1` .
- Change the `shapesize` of the turtle named `second` to `1,1,1` .
- Move the turtle named `first` in a circle with a radius of 50, then rotate it to the right 10 degrees.
- Move the turtle named `second` in a circle with a radius of 50, then rotate it to the left 10 degrees.
- Repeat steps 9-10 seven times.

## Answer Keys & Solutions

### Checkpoint Solutions

#### Customize Your Turtle

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.bgcolor("#ffb866")
5 turtle.title("Customizing Turtles")
6
7
8 turtle.shape("triangle")
9
10 turtle.shapesize(1,4,3)
11 turtle.pencolor("maroon")
12 turtle.fillcolor("yellow")
13
14 turtle.forward(50)
15 turtle.circle(30)
```

### Questions

#### 1. How can you adjust the size of your turtle in the Python Turtles library?

MULTIPLE CHOICE

##### Correct Answer:

- A. turtle.shape() ✗ Incorrect
- B. turtle.getscreen() ✗ Incorrect
- C. turtle.shapesize() ✓ Correct
- D. turtle.pencolor() ✗ Incorrect

##### Explanation:

You adjust the shape and the size

#### 2. In the code below, what does the second argument inside the parentheses affect?

MULTIPLE CHOICE

**Correct Answer:**

- A. Stretch length. ✗ Incorrect
- B. Stretch width. ✓ Correct
- C. Outline width. ✗ Incorrect
- D. Fill color. ✗ Incorrect

**Explanation:**

The first value determines length and the the second value determines outline thickness.

MULTIPLE CHOICE

**3. What does the following code do?**

**Correct Answer:**

- A. Changes the entire turtle to red. ✗ Incorrect
- B. Changes the outline color of the turtle. ✓ Correct
- C. Changes the fill color of the turtle. ✗ Incorrect
- D. Adjusts the turtle's size. ✗ Incorrect

**Explanation:**

Pencolor affects the outline of the turtle

**4. How can you change both the color and the outline of the turtle in one line of code?**

MULTIPLE CHOICE

**Correct Answer:**

- A. `turtle.pencolor("red")` ✗ Incorrect
- B. `turtle.fillcolor("blue")` ✗ Incorrect
- C. `turtle.color("red", "blue")` ✓ Correct
- D. `turtle.shape("turtle")` ✗ Incorrect

**Explanation:**



Both colors are on one line

### 5. In the following code, what part of the turtle will be red?

MULTIPLE CHOICE

**Correct Answer:**

- A. The outline of the turtle. ✓ Correct
- B. The fill color of the turtle. ✗ Incorrect
- C. The entire turtle. ✗ Incorrect
- D. None of the turtle. ✗ Incorrect

#### Explanation:

The first value determines the outline of the turtle

### 6. What does the following code do?

MULTIPLE CHOICE

**Correct Answer:**

- A. Changes the turtle to a turtle shape. ✓ Correct
- B. Adjusts the turtle's size. ✗ Incorrect
- C. Changes the fill color of the turtle. ✗ Incorrect
- D. Changes the outline color of the turtle. ✗ Incorrect

#### Explanation:

You can change the shape of the turtle

### 7. To make different sizes of turtles, which command should you use?

MULTIPLE CHOICE

**Correct Answer:**

- A. `turtle.size()` ✗ Incorrect
- B. `turtle.shapesize()` ✓ Correct
- C. `turtle.widthlength()` ✗ Incorrect

D. turtle.scale()

✗ Incorrect

#### Explanation:

Both the shape and the size are adjusted

### 8. What type of values can be added inside the parentheses of the following code?

MULTIPLE CHOICE

#### Correct Answer:

A. Only numerical values.

✗ Incorrect

B. Only hexadecimal values.

✗ Incorrect

C. Only color names.

✗ Incorrect

D. Both color names and hexadecimal values.

✓ Correct

#### Explanation:

You can add whatever value determines a color

### 9. Debug the following code:

DEBUG CODE

#### Incorrect Code:

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.shape(circle)
```

#### Correct Solution:

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.shape("circle")
```

#### Explanation:

This code is missing quotation marks

DEBUG CODE

## 10. Debug the following code:

### Incorrect Code:

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.color("red" "blue")
```

### Correct Solution:

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.color("red", "blue")
```

### Explanation:

This code is missing a comma

## Challenges

### 1. Rocket Launch

#### Solution:

```
1 import turtle
2 turtle.getscreen()
3 rocket = turtle.Turtle()
4
5 turtle.bgcolor("#92e8e2")
6 turtle.title("Rocket Launch")
7
8 rocket.shape("arrow")
9
10 rocket.shapesize(1,4,2)
11 rocket.color("red", "silver")
12
13 rocket.left(90)
14 rocket.forward(100)
```

### 2. Turtle Outline

#### Solution:

```
1 import turtle
2 turtle.getscreen()
3 josh = turtle.Turtle()
4
5
6 turtle.bgcolor("#63ff0f")
7 turtle.title("Border Turtle")
8
9
10 josh.shape("turtle")
```

```
11
12
13 josh.shapesize(3,3,10)
14 josh.color("orange", "blue")
15
16 josh.left(45)
17 josh.forward(100)
```

### 3. Giant Turtle

**Solution:**

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.bgcolor("tan")
5 turtle.title("Giant Turtle")
6
7
8 turtle.shape("turtle")
9
10 turtle.shapesize(10,10,5)
11 turtle.color("#1c781e", "#34e537")
12
13 turtle.forward(100)
14 turtle.left(90)
15 turtle.forward(100)
16 turtle.left(90)
17 turtle.forward(100)
18 turtle.left(90)
19 turtle.forward(100)
20 turtle.left(90)
```

### 4. Shadow Show

**Solution:**

```
1 import turtle
2 turtle.getscreen()
3 first = turtle.Turtle()
4 second = turtle.Turtle()
5 third = turtle.Turtle()
6 fourth = turtle.Turtle()
7 fifth = turtle.Turtle()
8 sixth = turtle.Turtle()
9
10 turtle.bgcolor("black")
11 turtle.title("Shadow Show")
12
13
14 first.shape("square")
15 second.shape("arrow")
16 third.shape("circle")
17 fourth.shape("turtle")
18 fifth.shape("triangle")
```

```
19 sixth.shape("classic")
20
21 first.shapesize(2,2,1)
22 first.color("white", "white")
23
24 second.shapesize(2,2,1)
25 second.color("white", "white")
26
27
28 third.shapesize(2,2,1)
29 third.color("white", "white")
30
31 fourth.shapesize(2,2,1)
32 fourth.color("white", "white")
33
34 fifth.shapesize(2,2,1)
35 fifth.color("white", "white")
36
37 sixth.shapesize(2,2,1)
38 sixth.color("white", "white")
39
40 first.forward(75)
41
42 second.right(45)
43 second.forward(75)
44
45 third.right(90)
46 third.forward(75)
47
48 fourth.right(135)
49 fourth.forward(75)
50
51 fifth.right(180)
52 fifth.forward(75)
53
54 sixth.right(225)
55 sixth.forward(75)
```

## 5. Taking Turns

### Solution:

```
1 import turtle
2 turtle.getscreen()
3 first = turtle.Turtle()
4 second = turtle.Turtle()
5
6 turtle.bgcolor("navy")
7 turtle.title("Taking Turns")
8
9
10 first.shape("circle")
11 second.shape("square")
12
13 first.shapesize(1,1,1)
```

```
14 first.color("yellow", "yellow")
15
16 second.shapesize(1,1,1)
17 second.color("lightgreen", "lightgreen")
18
19
20 first.circle(50)
21 first.right(10)
22 second.circle(50)
23 second.left(10)
24
25 first.circle(50)
26 first.right(10)
27 second.circle(50)
28 second.left(10)
29
30 first.circle(50)
31 first.right(10)
32 second.circle(50)
33 second.left(10)
34
35 first.circle(50)
36 first.right(10)
37 second.circle(50)
38 second.left(10)
39
40 first.circle(50)
41 first.right(10)
42 second.circle(50)
43 second.left(10)
44
45 first.circle(50)
46 first.right(10)
47 second.circle(50)
48 second.left(10)
49
50
51 first.circle(50)
52 first.right(10)
53 second.circle(50)
54 second.left(10)
```