

## Customize the Pen

---

### Textbook

---

## Customize the Pen



Now let's learn how to customize the pen as your turtle draws.

### Pen Thickness

Let's adjust how thick the pen drawing is.

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.pensize(5)
```

The higher the number in the parentheses, the thicker the drawing.

### Pen Color

Now let's adjust the color of the pen drawing.

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

## Turtle Speed

You can speed up or slow down your turtle.

```
1 turtle.speed(1)
2 turtle.forward(200)
3 turtle.speed(10)
4 turtle.forward(200)
```

The higher the number in the parentheses, the faster your turtle will go.

## Stamp

You can also leave a stamp of your turtle shape as you go! This is done with the `turtle.stamp()` command.

```
1 turtle.forward(50)
2 turtle.stamp()
3 turtle.forward(100)
4 turtle.stamp()
5 turtle.forward(50)
```

This will leave little stamps of your turtle shape.

## Checkpoint

---

### Customize the Pen

Practice customizing your pen!

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 pensize of your turtle to 5.
5. Change the pencolor of your turtle to `yellow`.
6. Move the turtle forward 30.
7. Change the speed to 10.
8. Stamp the turtle.
9. Rotate your turtle to the left 40.
10. Move your turtle forward 100.



### 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 pensize of your turtle to 5.
- Change the pencolor of your turtle to `yellow`.
- Move the turtle forward 30.
- Change the speed to 10.
- Stamp the turtle.
- Rotate your turtle to the left 40.
- Move your turtle forward 100.

## Questions (10)

**1. What effect does increasing the number in the parentheses of the following code have on the pen drawing?**

MULTIPLE CHOICE

```
turtle.pensize(5)
```

**Choose the correct answer:**

- A. Makes the pen color red.
- B. Changes the speed of the turtle.
- C. Decreases the thickness of the drawing.
- D. Increases the thickness of the drawing.

**2. How can you change the color of the pen using Turtles?**

MULTIPLE CHOICE

**Choose the correct answer:**

- A. `turtle.speed(1)`
- B. `turtle.pencolor("red")`
- C. `turtle.color(5)`
- D. `turtle.shade()`

**3. Explain the relationship between the number in the parentheses of the following code and the speed of the turtle.**

MULTIPLE CHOICE

**Example:**

```
1 turtle.speed(1)
```

```
turtle.speed(1)
```

**Choose the correct answer:**

- A. Higher number makes turtle move faster.
- B. Higher number makes turtle move slower.
- C. Changes pen color.
- D. Adjusts pen thickness.

**4. What is the purpose of the following code in the context of the passage?****Example:**

```
1 turtle.stamp()
```

```
turtle.stamp()
```

**Choose the correct answer:**

- A. Changes pen color.
- B. Adjusts pen thickness.
- C. Leaves a stamp of the turtle shape.
- D. Speeds up the turtle.

**5. How would you describe the impact of a higher number in the parentheses of the following code?****Example:**

```
1 turtle.pensize()
```

```
turtle.pensize()
```

**Choose the correct answer:**

- A. Increases thickness.
- B. Changes color.
- C. Decreases thickness.
- D. Speeds up turtle.

**6. What command is used to leave a stamp of the turtle shape as it moves?****Choose the correct answer:**

- A. turtle.pensize(5)
- B. turtle.pencolor("red")
- C. turtle.speed(1)
- D. turtle.stamp()

### 7. Debug the following code:

[DEBUG CODE](#)

#### Code to Debug:

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.pen.size(5)
```

### 8. Debug the following code:

[DEBUG CODE](#)

#### Code to Debug:

```
1 import turtles
2 turtle.getscreen()
3
4 turtle.pencolor("purple")
```

### 9. Debug the following code:

[DEBUG CODE](#)

#### Code to Debug:

```
1 import turtle
2 turtle.getscreen()
3
4 turtlespeed(5)
5 turtle.forward(20)
6 turtle.speed(10)
7 turtle.forward(20)
```

### 10. Debug the following code:

[DEBUG CODE](#)

#### Code to Debug:

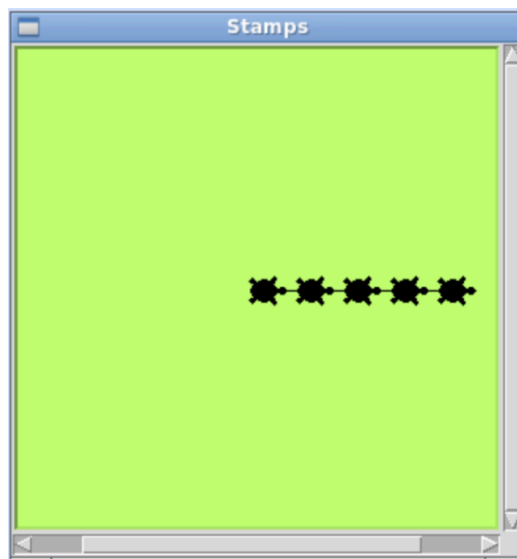
```
1 import turtle
2 turtle.getscreen()
3
4 turtle.forward(20)
5 turtle.stamp("turtle")
```

## Challenges (5)

### 1. Stamps

Practice creating turtle stamps!

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 `turtle`.
5. Add at least 5 stamp commands.
6. Add at least 4 forward commands.



#### 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 `turtle`.
- Add at least 5 stamp commands.
- Add at least 4 forward commands.

## 2. Circles

Practice seeing the difference in pen size by drawing circles!

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. Add a custom pencolor.
5. Set the pensize to 2.
6. Draw a circle sized 40.
7. Rotate your turtle to the right 30.
8. Repeat steps 5-7 a total of 5 times, increasing the pensize by 2 every time.



### Requirements:

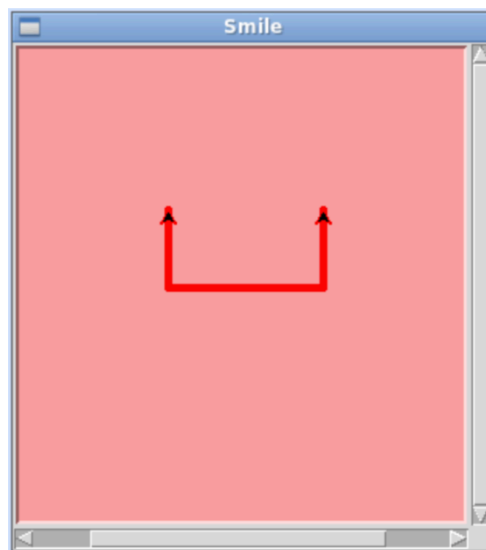
- Include the necessary code to start up a Python screen (import the library and generate a screen).
- Add a background color with a hexadecimal value.
- Add a custom title to your turtle screen.
- Add a custom pencolor.
- Set the pensize to 2.
- Draw a circle sized 40.
- Rotate your turtle to the right 30.
- Repeat steps 5-7 a total of 5 times, increasing the pensize by 2 every time.



### 3. Smile

Practice moving two turtles! Use two turtles to draw a smile shape.

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. Create one Python turtle variable named `henry`.
5. Make the pensize for the turtle named `henry` 5.
6. Make the pencolor for the turtle named `henry` `red`.
7. Move the turtle named `henry` forward 50.
8. Rotate the turtle named `henry` to the left 90 degrees.
9. Move the turtle named `henry` forward 50.
10. Make the pensize for the default turtle 5.
11. Make the pencolor for the default turtle `red`.
12. Move the default turtle variable named `turtle` backward 50.
13. Rotate the default turtle variable named `turtle` to the left 90 degrees.
14. Move the default turtle variable named `turtle` forward 50.



#### Requirements:

- Import the turtle library and generate a turtle screen.
- Create a variable named `henry` and assign it to a turtle from the Python turtle library.
- Move the turtle named `henry` forward 50.
- Rotate the turtle named `henry` to the left 90 degrees.
- Move the turtle named `henry` forward 50.

- Move the default turtle variable named `turtle` backward 50.
- Rotate the default turtle variable named `turtle` to the left 90 degrees.
- Move the default turtle variable named `turtle` forward 50.
- Make the pensize for the turtle named `henry` 5.
- Make the pencolor for the turtle named `henry` `red` .
- Make the pensize for the default turtle 5.
- Make the pencolor for the default turtle `red` .

#### 4. Target

Draw a target with red and white dots!

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. Make the pencolor `red`.
5. Draw a dot sized 100.
6. Make the pencolor `white`.
7. Draw a dot sized 80.
8. Make the pencolor `red`.
9. Draw a dot sized 60.
10. Make the pencolor `white`.
11. Draw a dot sized 40.
12. Make the pencolor `red`.
13. Draw a dot sized 20.



#### Requirements:

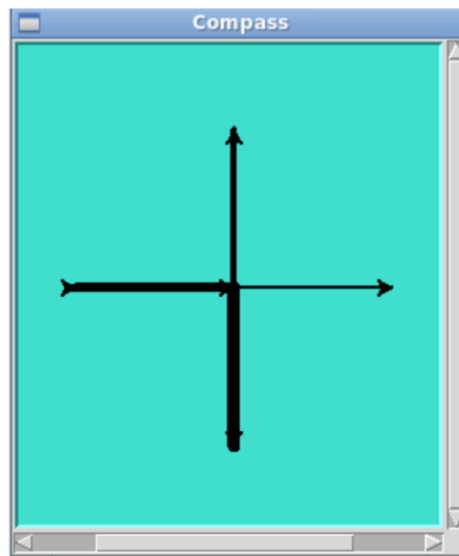
- Include the necessary code to start up a Python screen (import the library and generate a screen).
- Add a background color with a hexadecimal value.
- Add a custom title to your turtle screen.
- Make the pencolor `red`.
- Draw a dot sized 100.
- Make the pencolor `white`.
- Draw a dot sized 80.

- Make the pencolor `red` .
- Draw a dot sized 60.
- Make the pencolor `white` .
- Draw a dot sized 40.
- Make the pencolor `red` .
- Draw a dot sized 20.

## 5. Compass

Create four turtles who move in the four opposite directions.

1. Include the necessary code to start up a Python screen (import the library and generate a screen).
2. Create 4 Python turtle variables. Name the variables `first` , `second` , `third` , and `fourth` .
3. Give the turtle screen a background color of `turquoise` .
4. Give a title of `Compass` to your turtle screen.
5. Include 4 different pen sizes. One for each turtle.
6. Move `first` forward 100.
7. Rotate `second` to the left 90.
8. Move `second` forward 100.
9. Move `third` backward 100.
10. Rotate `fourth` to the right 90.
11. Move `fourth` forward 100.



### Requirements:

- Include the necessary code to start up a Python screen (import the library and generate a screen).
- Create 4 Python turtle variables. Name the variables `first` , `second` , `third` , and `fourth` .
- Give the turtle screen a background color of `turquoise` .
- Give a title of `Compass` to your turtle screen.
- Move `first` forward 100.
- Rotate `second` to the left 90.
- Move `second` forward 100.
- Move `third` backward 100.
- Rotate `fourth` to the right 90.

- Move `fourth` forward 100.
- Include 4 different pen sizes. One for each turtle.

## Answer Keys & Solutions

### Checkpoint Solutions

#### Customize the Pen

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.bgcolor("#001d4d")
5 turtle.title("Checkpoint")
6 turtle.pensize(5)
7 turtle.pencolor("yellow")
8
9 turtle.forward(30)
10 turtle.speed(10)
11 turtle.stamp()
12 turtle.left(40)
13 turtle.forward(100)
```

### Questions

1. What effect does increasing the number in the parentheses of the following code have on the pen drawing?

MULTIPLE CHOICE

Correct Answer:

- A. Makes the pen color red. ✗ Incorrect
- B. Changes the speed of the turtle. ✗ Incorrect
- C. Decreases the thickness of the drawing. ✗ Incorrect
- D. Increases the thickness of the drawing. ✓ Correct

#### Explanation:

The higher the number, the thicker the line

2. How can you change the color of the pen using Turtles?

MULTIPLE CHOICE

Correct Answer:

A. `turtle.speed(1)`

✗ Incorrect

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

✓ Correct

C. `turtle.color(5)`

✗ Incorrect

D. `turtle.shade()`

✗ Incorrect

**Explanation:**

The command is `pencolor`.

**3. Explain the relationship between the number in the parentheses of the following code and the speed of the turtle.**

MULTIPLE CHOICE

**Correct Answer:**

A. Higher number makes turtle move faster.

✓ Correct

B. Higher number makes turtle move slower.

✗ Incorrect

C. Changes pen color.

✗ Incorrect

D. Adjusts pen thickness.

✗ Incorrect

**Explanation:**

10 will be faster than 5.

**4. What is the purpose of the following code in the context of the passage?**

MULTIPLE CHOICE

**Correct Answer:**

A. Changes pen color.

✗ Incorrect

B. Adjusts pen thickness.

✗ Incorrect

C. Leaves a stamp of the turtle shape.

✓ Correct

D. Speeds up the turtle.

✗ Incorrect

**Explanation:**



Stamp leaves a little shape of the turtle at that spot.

## 5. How would you describe the impact of a higher number in the parentheses of the following code?

MULTIPLE CHOICE

Correct Answer:

- A. Increases thickness. ✓ Correct
- B. Changes color. ✗ Incorrect
- C. Decreases thickness. ✗ Incorrect
- D. Speeds up turtle. ✗ Incorrect

Explanation:

10 will be thicker than 5.

## 6. What command is used to leave a stamp of the turtle shape as it moves?

MULTIPLE CHOICE

Correct Answer:

- A. `turtle.pensize(5)` ✗ Incorrect
- B. `turtle.pencolor("red")` ✗ Incorrect
- C. `turtle.speed(1)` ✗ Incorrect
- D. `turtle.stamp()` ✓ Correct

Explanation:

`stamp()` leaves a stamp of the variable it's attached to

## 7. Debug the following code:

DEBUG CODE

Incorrect Code:

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.pen.size(5)
```

### Correct Solution:

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.pensize(5)
```

### Explanation:

There's too many periods in this code

## 8. Debug the following code:

DEBUG CODE

### Incorrect Code:

```
1 import turtles
2 turtle.getscreen()
3
4 turtle.pencolor("purple")
```

### Correct Solution:

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.pencolor("purple")
```

### Explanation:

The first line has a bug

## 9. Debug the following code:

DEBUG CODE

### Incorrect Code:

```
1 import turtle
2 turtle.getscreen()
3
4 turtlespeed(5)
5 turtle.forward(20)
6 turtle.speed(10)
7 turtle.forward(20)
```

### Correct Solution:

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.speed(5)
5 turtle.forward(20)
6 turtle.speed(10)
7 turtle.forward(20)
```

**Explanation:**

This code is missing a period.

**10. Debug the following code:**

DEBUG CODE

**Incorrect Code:**

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.forward(20)
5 turtle.stamp("turtle")
```

**Correct Solution:**

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.forward(20)
5 turtle.stamp()
```

**Explanation:**

The `stamp()` command should be empty.

**Challenges****1. Stamps****Solution:**

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.bgcolor("#bfff70")
5 turtle.title("Stamps")
6 turtle.shape("turtle")
7
8 turtle.stamp()
9 turtle.forward(30)
10 turtle.stamp()
11 turtle.forward(30)
12 turtle.stamp()
13 turtle.forward(30)
14 turtle.stamp()
15 turtle.forward(30)
16 turtle.stamp()
```

## 2. Circles

**Solution:**

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.title("Circles")
5 turtle.bgcolor("#bfff70")
6 turtle.pencolor("#e64533")
7
8 turtle.pensize(2)
9 turtle.circle(40)
10 turtle.right(30)
11
12 turtle.pensize(4)
13 turtle.circle(40)
14 turtle.right(30)
15
16 turtle.pensize(6)
17 turtle.circle(40)
18 turtle.right(30)
19
20 turtle.pensize(8)
21 turtle.circle(40)
22 turtle.right(30)
23
24 turtle.pensize(10)
25 turtle.circle(40)
26 turtle.right(30)
```

## 3. Smile

**Solution:**

```
1 import turtle
2 turtle.getscreen()
3 turtle.bgcolor("#f99f9f")
4 turtle.title("Smile")
5 henry = turtle.Turtle()
6
7 henry.pensize(5)
8 henry.pencolor("red")
9 henry.forward(50)
10 henry.left(90)
11 henry.forward(50)
12
13 turtle.pensize(5)
14 turtle.pencolor("red")
15 turtle.backward(50)
16 turtle.left(90)
17 turtle.forward(50)
```

## 4. Target

Solution:

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.bgcolor("#0000FF")
5
6 turtle.title("Target")
7
8 turtle.pencolor("red")
9 turtle.dot(100)
10
11 turtle.pencolor("white")
12 turtle.dot(80)
13
14 turtle.pencolor("red")
15 turtle.dot(60)
16
17 turtle.pencolor("white")
18 turtle.dot(40)
19
20 turtle.pencolor("red")
21 turtle.dot(20)
```

## 5. Compass

Solution:

```
1 import turtle
2 turtle.getscreen()
3 first = turtle.Turtle()
4 second = turtle.Turtle()
5 third = turtle.Turtle()
6 fourth = turtle.Turtle()
7
8 turtle.bgcolor("turquoise")
9 turtle.title("Compass")
10
11 first.forward(100)
12 second.left(90)
13 second.forward(100)
14 third.backward(100)
15 fourth.right(90)
16 fourth.forward(100)
17 first.pensize(20)
18 second.pensize(50)
19 third.pensize(70)
20 fourth.pensize(10)
```