

## Draw Shapes with Python Turtles

---

### Textbook

---

## Draw Shapes with Python Turtles



Now that we can move the turtle different directions, let's try drawing a shape!

### Draw a Square

Try drawing a square with a Python Turtle.

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.forward(50)
5 turtle.left(90)
6 turtle.forward(50)
7 turtle.left(90)
8 turtle.forward(50)
9 turtle.left(90)
10 turtle.forward(50)
```

As the turtle moves forward and turns to the left, it will draw a square. Try adjusting the lengths of the sides to make different sizes of squares. See if you can draw a rectangle!



# Algorithms

An [algorithm](#) is like a set of step-by-step instructions that tells you exactly what to do to solve a problem or complete a task. It's like a recipe for a computer or a person to follow, making things easier and more organized. Algorithms help us solve problems in a clear and systematic way.

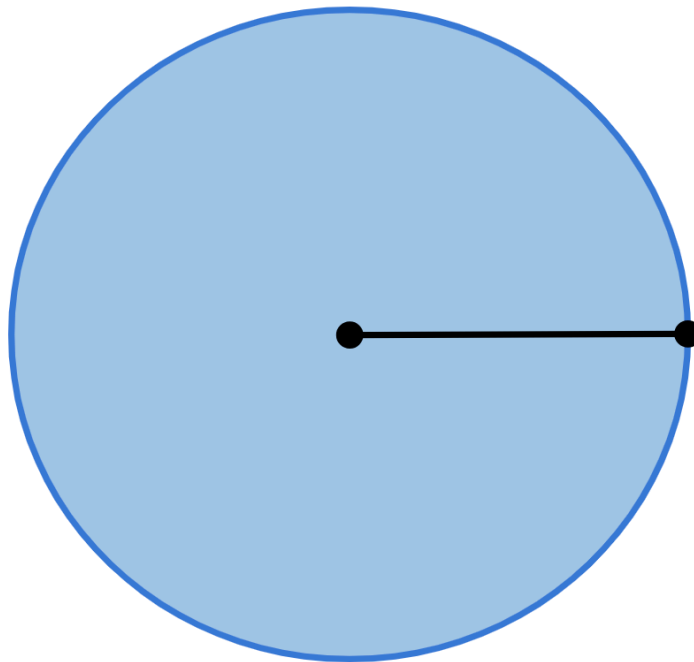
The set of instructions we added to create a square is an example of an algorithm.

## Draw a Circle

There's actually a preset way to draw circles to make it easier.

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.circle(50)
```

The value inside the parentheses is the [radius](#) of the circle. The radius is the distance from the center of the circle to the edge of the circle. In the diagram below, the black line is the radius.



Try changing the radius to see what happens!

## Dot

You can also fill in the circle. This is done with the `turtle.dot()` command.

```
1 import turtle
2 turtle.getscreen()
3
```



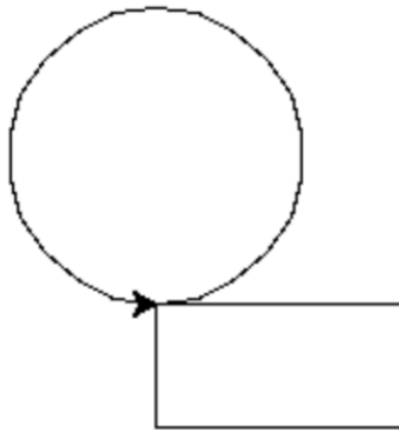
## Checkpoint

---

### Draw Shapes with Python Turtles

Draw a circle and then a rectangle.

1. Include the necessary code to start up a python screen (import the library and generate a screen).
2. Draw a circle with a radius of 60.
3. Move the turtle forward 100.
4. Rotate the turtle to the right 90.
5. Move the turtle forward 50.
6. Rotate the turtle to the right 90.
7. Move the turtle forward 100.
8. Rotate the turtle to the right 90.
9. Move the turtle forward 50.
10. Rotate the turtle to the right 90.



### Requirements:

- Include the necessary code to start up a python screen (import the library and generate a screen).
- Draw a circle with a radius of 60.



- Move the turtle forward 100.
- Rotate the turtle to the right 90.
- Move the turtle forward 50.
- Rotate the turtle to the right 90.
- Move the turtle forward 100.
- Rotate the turtle to the right 90.
- Move the turtle forward 50.
- Rotate the turtle to the right 90.

## Questions (9)

**1. What programming concept is introduced in the passage that involves a set of step-by-step instructions to solve a problem or complete a task?**

MULTIPLE CHOICE

**Choose the correct answer:**

- A. Looping
- B. Algorithms
- C. Variables
- D. Libraries

**2. How can you adjust the size of a square in the Python Turtle program?**

MULTIPLE CHOICE

**Choose the correct answer:**

- A. By changing the turtle's color.
- B. By adjusting the angle of the turtle.
- C. By modifying the lengths of the sides.
- D. By changing the background color.

**3. Do algorithms only apply when working on math problems?**

MULTIPLE CHOICE

**Choose the correct answer:**

- A. False
- B. True



**4. In the code provided, what does the value inside the parentheses represent when drawing a circle?**

MULTIPLE CHOICE

```
turtle.circle(50)
```

**Choose the correct answer:**

- A. Diameter
- B. Circumference
- C. Radius
- D. Perimeter

**5. What is the purpose of the line `turtle.getscreen()` in the Python Turtle program?**

MULTIPLE CHOICE

**Choose the correct answer:**

- A. It initializes the turtle graphics window.
- B. It closes the turtle graphics window.
- C. It changes the turtle's color.
- D. It has no impact on the program.

**6. Debug the following code:**

DEBUG CODE

**Code to Debug:**

```
1 import turtle
2 turtle.getscreen()
3
4 turtl.circle(50)
```

**7. Debug the following code:**

DEBUG CODE

**Code to Debug:**

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.circle(40)
5 turtle.dot20)
```



## 8. Debug the following code:

DEBUG CODE

### Code to Debug:

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.forward(20)
5 turtle.left(90)
6 turtle.forward(20)
7 turtle.left(90)
8 turtle.forward(20)
9 turtle.left(90)
10 turtle.forward(20)
```

## 9. Debug the following code:

DEBUG CODE

### Code to Debug:

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.forward(20)
5 turtle.left(90)
6 turtle.forward(20)
7 turtle.left(90)
8 turtle.forward(20)
9 turtle.left(90)
10 turtle.foward(20)
```



## Challenges (5)

### 1. Backward Square

Now draw a square backwards! **Only move your turtle backwards for your square.**

1. Include the necessary code to start up a python screen (import the library and generate a screen).
2. Include 4 sides that are 75 long.
3. Turn your turtle 90 degrees a total of 4 times.



#### Requirements:

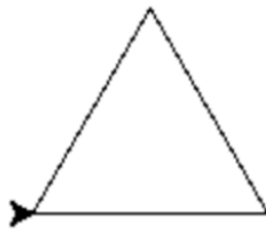
- Include the necessary code to start up a python screen (import the library and generate a screen).
- Include 4 sides that are 75 long.
- Turn your turtle 90 degrees a total of 4 times.



## 2. Triangle

Draw a triangle!

1. Include the necessary code to start up a python screen (import the library and generate a screen).
2. Move your turtle forward 100.
3. Rotate your turtle to the left 120.
4. Move your turtle forward 100.
5. Rotate your turtle to the left 120.
6. Move your turtle forward 100.
7. Rotate your turtle to the left 120.



### Requirements:

- Include the necessary code to start up a python screen (import the library and generate a screen).
- Move the turtle forward 100.
- Rotate the turtle to the left 120.
- Move the turtle forward 100.
- Rotate the turtle to the left 120.
- Move the turtle forward 100.
- Rotate the turtle to the left 120.



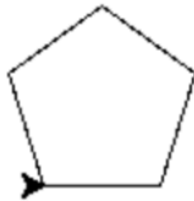
### 3. Pentagon

Draw a pentagon! What do you think the angle needs to be for a pentagon? Try out some angles and see what happens!

A pentagon has 5 sides.

*Hint: The angle is 360 divided by 5*

1. Include the necessary code to start up a python screen (import the library and generate a screen).
2. Include 5 sides of 50 each.
3. Turn the turtle a total of 5 times with the appropriate angle.



#### Requirements:

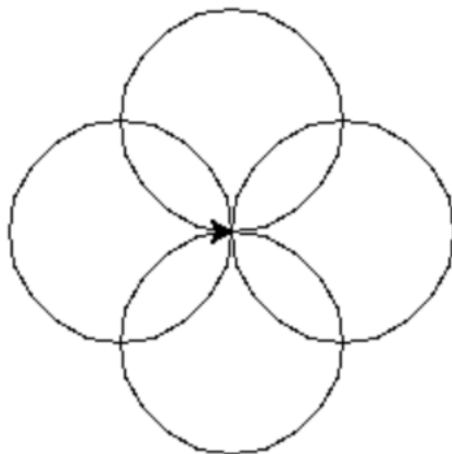
- Include the necessary code to start up a python screen (import the library and generate a screen).
- Move the turtle forward 50 a total of 5 times.
- Turn the turtle a total of 5 times with the necessary angle.



#### 4. Clover Shape

Draw a clover shape!

1. Include the necessary code to start up a python screen (import the library and generate a screen).
2. Draw a circle with a radius of 50.
3. Rotate the turtle to the right 90 degrees.
4. Draw a second circle with a radius of 50.
5. Rotate the turtle to the right 90 degrees.
6. Draw a third circle with a radius of 50.
7. Rotate the turtle to the right 90 degrees.
8. Draw a fourth circle with a radius of 50.
9. Rotate the turtle to the right 90 degrees.



#### Requirements:

- Include the necessary code to start up a python screen (import the library and generate a screen).
- Draw a circle with a radius of 50.
- Rotate the turtle to the right 90 degrees.
- Draw a second circle with a radius of 50.
- Rotate the turtle to the right 90 degrees.
- Draw a third circle with a radius of 50.
- Rotate the turtle to the right 90 degrees.

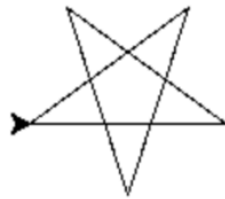


- Draw a fourth circle with a radius of 50.

## 5. Star

Draw a star!

1. Include the necessary code to start up a python screen (import the library and generate a screen).
2. Move the turtle forward 100.
3. Rotate the turtle to the left 144 degrees.
4. Repeat steps 2-3 five times.



### Requirements:

- Include the necessary code to start up a python screen (import the library and generate a screen).
- Move the turtle forward 100. Add this step a total of 5 times.
- Rotate the turtle to the left 144 degrees. Add this step a total of 5 times.



## Answer Keys & Solutions

### Checkpoint Solutions

#### Draw Shapes with Python Turtles

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.circle(60)
5
6 turtle.forward(100)
7 turtle.right(90)
8 turtle.forward(50)
9 turtle.right(90)
10 turtle.forward(100)
11 turtle.right(90)
12 turtle.forward(50)
13 turtle.right(90)
```

### Questions

1. What programming concept is introduced in the passage that involves a set of step-by-step instructions to solve a problem or complete a task?

MULTIPLE CHOICE

Correct Answer:

- |               |             |
|---------------|-------------|
| A. Looping    | ✗ Incorrect |
| B. Algorithms | ✓ Correct   |
| C. Variables  | ✗ Incorrect |
| D. Libraries  | ✗ Incorrect |

#### Explanation:

This term refers to the total list of instructions.

2. How can you adjust the size of a square in the Python Turtle program?

MULTIPLE CHOICE

Correct Answer:



A. By changing the turtle's color.

✗ Incorrect

B. By adjusting the angle of the turtle.

✗ Incorrect

C. By modifying the lengths of the sides.

✓ Correct

D. By changing the background color.

✗ Incorrect

#### Explanation:

Changing the length of the sides of the square will change the overall size of the square.

### 3. Do algorithms only apply when working on math problems?

MULTIPLE CHOICE

Correct Answer:

A. False

✓ Correct

B. True

✗ Incorrect

### 4. In the code provided, what does the value inside the parentheses represent when drawing a circle?

MULTIPLE CHOICE

Correct Answer:

A. Diameter

✗ Incorrect

B. Circumference

✗ Incorrect

C. Radius

✓ Correct

D. Perimeter

✗ Incorrect

#### Explanation:

It represents the distance from the center of the circle to the edge of the circle.

### 5. What is the purpose of the line `turtle.getscreen()` in the Python Turtle program?

MULTIPLE CHOICE

Correct Answer:



A. It initializes the turtle graphics window.

✓ Correct

B. It closes the turtle graphics window.

✗ Incorrect

C. It changes the turtle's color.

✗ Incorrect

D. It has no impact on the program.

✗ Incorrect

### Explanation:

The code starts a screen for the turtle to walk on.

## 6. Debug the following code:

DEBUG CODE

### Incorrect Code:

```
1 import turtle
2 turtle.getscreen()
3
4 turtl.circle(50)
```

### Correct Solution:

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.circle(50)
```

### Explanation:

turtle is spelled wrong

## 7. Debug the following code:

DEBUG CODE

### Incorrect Code:

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.circle(40)
5 turtle.dot20)
```

### Correct Solution:

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.circle(40)
5 turtle.dot(20)
```



### Explanation:

This code is missing a parenthesis

## 8. Debug the following code:

DEBUG CODE

### Incorrect Code:

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.forward(20)
5 turtle.left(90)
6 turtle.forward(20)
7 turtl.left(90)
8 turtle.forward(20)
9 turtle.left(90)
10 turtle.forward(20)
```

### Correct Solution:

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.forward(20)
5 turtle.left(90)
6 turtle.forward(20)
7 turtle.left(90)
8 turtle.forward(20)
9 turtle.left(90)
10 turtle.forward(20)
```

## 9. Debug the following code:

DEBUG CODE

### Incorrect Code:

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.forward(20)
5 turtle.left(90)
6 turtle.forward(20)
7 turtle.left(90)
8 turtle.forward(20)
9 turtle.left(90)
10 turtle.foward(20)
```

### Correct Solution:

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



```
5 turtle.left(90)
6 turtle.forward(20)
7 turtle.left(90)
8 turtle.forward(20)
9 turtle.left(90)
10 turtle.forward(20)
```

**Explanation:**

This code has a spelling mistake

## Challenges

### 1. Backward Square

**Solution:**

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.backward(75)
5 turtle.right(90)
6 turtle.backward(75)
7 turtle.right(90)
8 turtle.backward(75)
9 turtle.right(90)
10 turtle.backward(75)
11 turtle.right(90)
```

### 2. Triangle

**Solution:**

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.forward(100)
5 turtle.left(120)
6 turtle.forward(100)
7 turtle.left(120)
8 turtle.forward(100)
9 turtle.left(120)
```

### 3. Pentagon

**Solution:**

```
1 import turtle
2 turtle.getscreen()
3
```



```
4 turtle.forward(50)
5 turtle.left(72)
6 turtle.forward(50)
7 turtle.left(72)
8 turtle.forward(50)
9 turtle.left(72)
10 turtle.forward(50)
11 turtle.left(72)
12 turtle.forward(50)
13 turtle.left(72)
```

## 4. Clover Shape

**Solution:**

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.circle(50)
5 turtle.right(90)
6 turtle.circle(50)
7 turtle.right(90)
8 turtle.circle(50)
9 turtle.right(90)
10 turtle.circle(50)
11 turtle.right(90)
```

## 5. Star

**Solution:**

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.forward(100)
5 turtle.left(144)
6 turtle.forward(100)
7 turtle.left(144)
8 turtle.forward(100)
9 turtle.left(144)
10 turtle.forward(100)
11 turtle.left(144)
12 turtle.forward(100)
13 turtle.left(144)
```