

# Python Math Library

## Textbook

# Python Math Library



One of the most useful aspects of the Python language is that Python has many libraries with code that already exists. Instead of writing your own code, you can just reference these libraries. This can streamline the coding process and allows you to create programs faster and easier!

Python has an extensive [math library](#) which you can easily access. All you have to include at the top of your program is the following code:

```
import math
```

Once we have imported the [math library](#), we can easily use the following code examples.

## Rounding Up

The code `math.ceil( )` will round the value up to the next whole number no matter what the decimal value. You can think of **ceil** to mean ceiling.

```
1 import math
2 x = math.ceil(3.2)
```

```
3 print(x)
```

Try it!

## Rounding Down

The code `math.floor( )` will round the value down to the next whole number no matter what the decimal value.

```
1 import math
2 x = math.floor(3.2)
3 print(x)
```

Try it!

## The Value of Pi

**Pi** is an important number in geometry and math. The true value of pi has an infinite number of digits that come after the decimal place. To access the value of pi, you can simply use the following code.

```
1 import math
2 x = math.pi
3 print(x)
```

Try it!

## Square Root

The square root helps you figure out what value multiplied by itself gives you your target value. For example:  $5 \times 5 = 25$ . So the square root of 25 is 5.

The following code can help you find the square root using Python code. You can think of `sqrt` to mean square root.

```
1 import math
2 x = math.sqrt(100)
3 print(x)
```

Try it!

## Multiply by a Power

Often in math, finding a value raised to a certain power is important. For example, 3 to the power of 4 is equal to  $3 \times 3 \times 3 \times 3 = 81$ .

The following code can help you easily multiply a value by any power.

```
1 import math
2 x = pow(2, 4)
```

```
3 print(x)
```

Try it!

The first value in the parentheses is the value you want to raise to a power. The second value is the power you want to raise it to. So this code could also be written as

**2 × 2 × 2 × 2**

## Give Attribution

It's important to give proper credit when using outside libraries. This can be done with a comment in your code such as this:

```
#Math concepts used from the Python Math Library
```

## Importing Everything from a Module or Library

If you want to import everything from the math library, this allows you to streamline your code. To do this, use the following line at the top instead of import.

```
from math import *
```

The asterisk is used to mean "import all."

This makes it so you don't have to put the library math before it's modules.

For example, your code can go from this

```
1 import math
2 x = math.sqrt(100)
3 print(x)
```

Try it!

to this

```
1 from math import *
2 x = sqrt(100)
3 print(x)
```

Try it!

See how you no longer need `math.sqrt()` ? This streamlines your code to just `sqrt()`

## Further Learning

The [Python math library](#) has extensive capabilities. You can find the sine, cosine and tangent. You can convert from radians to degrees. You can find factorials and absolute values. There's no limit to what you can do and create with the Python language. For this lesson, we will not go into these more advanced math topics.

Here's a shortened version focusing on the key actions for students:

---

## Invitation: Let's Talk Math!

In our math class, we're not just finding answers; we're exploring *how* we think mathematically. Your ideas matter, and we learn best when we share and question together.

## Your Math Discussion Challenges:

### 1. "Show Your Work & Tell Your Story":

- When you solve a problem, **clearly explain your method and the math ideas** you used. Help others understand *your* thinking process.

### 2. "Be a Math Detective":

- When classmates share, **analyze their math thinking**. Ask questions like, "How did you get from here to there?" or "What does this part mean?"

### 3. "The Efficiency Expert":

- Look at different ways to solve a problem. **Compare which methods are more efficient or clearer**, and explain *why*.

### 4. "Learn from Our Errors":

- If you spot a mistake (yours or someone else's), gently **point it out and suggest how to correctly solve it**. Errors are opportunities to learn!

### 5. "Prove It!":

- When you present your results, **justify them with evidence** from your work. Build strong arguments and be ready to explain your 'whys.'

## Checkpoint

---

## Python Math Library

Practice using the Python Math Library! Create a program that uses the following elements from the library:

- Import math
- Use `ceil()` and `floor()`
- Use `sqrt`
- Use `pow( )`

### Requirements:

- Import the Python Math Library
- Use `ceil()`
- Use `floor()`
- Use `sqrt()`
- Use `pow()`

## Questions (7)

1. Which of the following is used to access the math library?

MULTIPLE CHOICE

Choose the correct answer:

- A. import math
- B. import math library
- C. access math
- D. math library

2. True or False: The math library is the only library in Python.

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

3. Which of the following are capabilities of the Python math library? Select all that apply.

SELECT MULTIPLE

Select all that apply:

- A. rounding up
- B. rounding down
- C. using the value of pi
- D. square root
- E. multiplying by a power

4. What will the following code print out?

MULTIPLE CHOICE

```
import math x = math.ceil(4.4) print(x)
```

Choose the correct answer:

- A. 4.4
- B. 4
- C. 4.5
- D. 5

**5. What will the following code print out?**

```
import math
x = math.floor(3.8)
print(x)
```

**Choose the correct answer:**

- A. 3
- B. 3.7
- C. 4
- D. 3.9

**6. Edit the text box below to debug (fix) the code:****Code to Debug:**

```
1 import math
2 x = math-sqrt(9)
3 print(x)
```

**7. Edit the text box below to debug (fix) the code:****Code to Debug:**

```
1 access math
2 x = math.floor(6.9)
3 print(x)
```

## Challenges (4)

### 1. Biking Town

You have just moved to a new town and notice that there are a lot of people who ride bikes! Everywhere you go there are people biking to the park, to the store, and to work. You wonder how many bikes there are in this town compared to how many people there are. You decide to write a program that can calculate this for you.

Create a program that takes in two inputs: the number of people in the town, and the number of bikes in the town. Your program will find out how many bikes there are compared to people and return the following statement: `In this town, for every bike that exists there are ____ people.`

You realize as you are building this program, that with different numbers you will inevitably end up with fractions. Since you don't want to consider a fraction of a person, if there is a decimal, you will round up to the nearest whole number.

For this challenge use the ceiling function imported from the Python Math library.

For this challenge, create a variable and assign it to the ceiling function.

For example:

Input 1 (number of people in the town): 100

Input 2 (number of bikes in the town): 50

Output: `In this town, for every bike that exists there are 2 people.`

Another example:

Input 1: `50`

Input 2: `75`

Output: `In this town, for every bike that exists there are 1 people.`

### 2. Cube Volume

Create a program that will find the volume of a cube!

The program will take in the length of a side of the cube as an input. Using the `pow()` function, calculate the volume of the cube.

For example:

Input: `5`

Output: `125`

Another example:

Input: `3`

Output: `27`

Hint: Volume of a cube is the length of a side raised to the power of 3.

*Note: For this challenge use `pow()`, not `math.pow()`*

### 3. Is it a Good Investment?

Did you know that you can invest in the stock market? This means that you can sometimes make money with very little work!

Sometimes the value of stocks goes up and sometimes the value goes down. The more experience you have with stocks, the better you are able to guess if a stock will become more or less valuable!

Create a program that can help you know if a stock was a good investment or not!

The formula for the rate of return on a stock over 2 years can be found below.

x = starting price

y = ending price after 2 years

$$\text{rate} = \sqrt{y / x} - 1$$

Let's say that you bought a stock for \$100, then after 2 years it was worth \$196

$$\text{rate} = \sqrt{196 / 100} - 1$$

$$\text{rate} = \sqrt{1.96} - 1$$

$$\text{rate} = 1.4 - 1$$

$$\text{rate} = .4$$

Since the decimal .4 is the same as 40%, this calculation shows that this investment had an annual return of 40% which is a really good investment. Generally speaking, any investment that has 10% or more annual return is considered a good investment.

Create a program that will take in 2 inputs: The value of the stock at the beginning, and the value of the stock at the end of 2 years. It will then print out the rate rounded to 2 decimal places.

For example:

Input 1 (value of stock at the beginning): 100

Input 2 (value of stock at the end): 200

Output (rate): 0.41

Another example:

Input 1: 50

Input 2: 150

Output: 0.73

**Hint:** round(x, 2)



## 4. Right Triangle

Write a program that can find the length of the diagonal of a right triangle! Remember that a right triangle has one angle that is 90 degrees.

Let's assume that the length of the two sides next to the 90 degree angle are named a and b. To find the length of the diagonal, we use what is called the Pythagorean Theorem.

$$a^2 + b^2 = c^2$$

Create a program that will take in two inputs: the length of side a and the length of side b. It will then print out the length of side c.

Round to 2 decimal places with `round(x, 2)`

*For example:*

Input 1 (length of side a): `3`

Input 2 (length of side b): `4`

Output (length of side c): `5.0`

*Another example:*

Input 1: `4`

Input 2: `8`

Output: `8.94`

## Answer Keys & Solutions

### Checkpoint Solutions

#### Python Math Library

```
1 import math
2 x = 3.6
3 y = 100
4 z = 5
5
6 print(math.ceil(x))
7 print(math.floor(x))
8 print(math.sqrt(y))
9 print(pow(z, 3))
```

### Questions

1. Which of the following is used to access the math library?

MULTIPLE CHOICE

Correct Answer:

- A. import math ✓ Correct
- B. import math library ✗ Incorrect
- C. access math ✗ Incorrect
- D. math library ✗ Incorrect

#### Explanation:

To use libraries in Python, we use the word "import"

2. True or False: The math library is the only library in Python.

MULTIPLE CHOICE

Correct Answer:

- A. True ✗ Incorrect
- B. False ✓ Correct

#### Explanation:

Random is an example of accessing a Python library. Datetime is another example of accessing Python library.

### 3. Which of the following are capabilities of the Python math library? Select all that apply.

SELECT MULTIPLE

#### Correct Answers:

- A. rounding up ✓ Correct
- B. rounding down ✓ Correct
- C. using the value of pi ✓ Correct
- D. square root ✓ Correct
- E. multiplying by a power ✓ Correct

#### Explanation:

The Python math library can do all of these answers and more.

### 4. What will the following code print out?

MULTIPLE CHOICE

#### Correct Answer:

- A. 4.4 ✗ Incorrect
- B. 4 ✗ Incorrect
- C. 4.5 ✗ Incorrect
- D. 5 ✓ Correct

#### Explanation:

`math.ceil()` will always round up, no matter what the decimal value.

### 5. What will the following code print out?

MULTIPLE CHOICE

#### Correct Answer:

- A. 3 ✓ Correct

B. 3.7

× Incorrect

C. 4

× Incorrect

D. 3.9

× Incorrect

### Explanation:

`math.floor( )` will always round down no matter what the decimal value

## 6. Edit the text box below to debug (fix) the code:

DEBUG CODE

### Incorrect Code:

```
1 import math
2 x = math-sqrt(9)
3 print(x)
```

### Correct Solution:

```
1 import math
2 x = math.sqrt(9)
3 print(x)
```

### Explanation:

They hyphen needs to be a different symbol.

## 7. Edit the text box below to debug (fix) the code:

DEBUG CODE

### Incorrect Code:

```
1 access math
2 x = math.floor(6.9)
3 print(x)
```

### Correct Solution:

```
1 import math
2 x = math.floor(3.2)
3 print(x)
```

### Explanation:

The word "access" is incorrect.

## Challenges

### 1. Biking Town

Solution:

```
1 import math
2 population = int(input("Population"))
3 bikes = int(input("Bikes"))
4
5 total = population/bikes
6 rounded = math.ceil(total)
7
8 print(f"In this town, for every bike that exists there are {rounded} people.")
```

### 2. Cube Volume

Solution:

```
1 import math
2 side = int(input("What is the length of the side?"))
3
4 print(pow(side, 3))
```

### 3. Is it a Good Investment?

Solution:

```
1 import math
2 beginning = int(input("What is the value of the stock at the beginning?"))
3 end = int(input("What is the value of the stock at the end of 2 years?"))
4
5 division = end / beginning
6
7 squareroot = math.sqrt(division)
8
9 rate = squareroot - 1
10
11 print(round(rate, 2))
```

### 4. Right Triangle

Solution:

```
1 import math
2 a = int(input("What is the length of side a?"))
3 b = int(input("What is the length of side b?"))
4
```

```
5 asquared = pow(a,2)
6 bsquared = pow(b,2)
7
8 csquared = asquared + bsquared
9
10 c = math.sqrt(csquared)
11 final = round(c, 2)
12
13 print(final)
```