

# CSC 111:

# Intro to Computer Science through Programming

Spring 2017  
Prof. Sara Mathieson



# Getting started today

- + Fill out anonymous feedback form (put in cardboard box)
- + Continue working on the `quadratic.py` program from Friday
- + Homework 2 due Tuesday night on Moodle
- + Office hours today 3-5pm (Ford 355)

# Outline: 2/13

- + Continue quadratic example (if/elif/else)
- + Practice with modules (math/random)
- + More string methods
- + (if time) Random colors exercise

Recap

# Solving a quadratic equation

Solve for x:

$$ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Example of factoring:  $x^2 - 4 = 0$

$$(x + 2)(x - 2) = 0$$

$$x + 2 = 0 \quad \text{or} \quad x - 2 = 0$$

$$x = -2$$

$$x = 2$$

# Module practice (math)

# Math module online documentation

## Table Of Contents

- 9.2. **math** — Mathematical functions
  - 9.2.1. Number-theoretic and representation functions
  - 9.2.2. Power and logarithmic functions
  - 9.2.3. Trigonometric functions
  - 9.2.4. Angular conversion
  - 9.2.5. Hyperbolic functions
  - 9.2.6. Special functions

## 9.2. **math** — Mathematical functions

This module is always available. It provides access to the mathematical functions defined by the C standard.

These functions cannot be used with complex numbers; use the functions of the same name from the **cmath** module if you require support for complex numbers. The distinction between functions which support complex numbers and those which don't is made since most users do not want to learn quite as much mathematics as required to understand complex numbers. Receiving an exception instead of a complex result allows earlier detection of the unexpected complex number used as a parameter, so that the programmer can determine how and why it was generated in the first place.

The following functions are provided by this module. Except when explicitly noted otherwise, all return values are floats.

### 9.2.7. Constants

**math.pi**

The mathematical constant  $\pi = 3.141592\dots$ , to available precision.

**math.e**

The mathematical constant  $e = 2.718281\dots$ , to available precision.

# Your modules have the same structure

```
>>> import my_math
>>> my_math.my_round(14.54,10)
10
>>> import math
>>> math.sqrt(9)
3.0
```



# Your modules have the same structure

```
>>> import my_math
>>> my_math.my_round(14.54,10)
10
>>> import math
>>> math.sqrt(9)
3.0
```

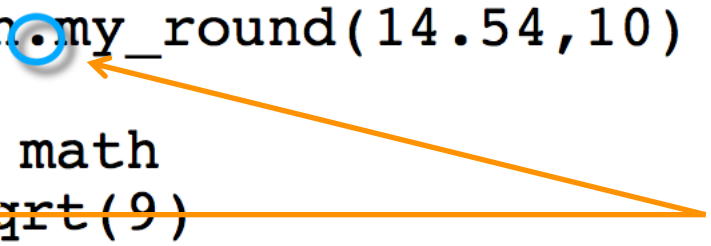
Module name



# Your modules have the same structure

```
>>> import my_math
>>> my_math.my_round(14.54,10)
10
>>> import math
>>> math.sqrt(9)
3.0
```

Period



# Your modules have the same structure

```
>>> import my_math
>>> my_math.my_round(14.54,10)
10
>>> import math
>>> math.sqrt(9)
3.0
```

Function name



Module practice (random)

# Random challenge:

1) Find the Python random module documentation online

2) Produce a random number between 0 and 1

0.10302849886915533

3) In a loop, produce 10 random numbers between 0 and 1

0.32874304980941216  
0.4278950868503729  
0.25948033275795135  
0.13088128774551644  
0.527463290117301  
0.21974575348546377  
0.5776857680193639  
0.05896275613911728  
0.3383859124009375  
0.6232418744867241

4) Produce a random number in the range 0,1,2,3,4 inclusive

4 4 0 4 1 4 1 0 0 1 2 4 1 4 2 4 0 1 1 2 4 0 3 0 1

# String methods

# Split a string based on space

Input:

```
>>> sentence = "Tomorrow is valentines day"
```

Challenge 1:

```
Tomorrow!  
is!  
valentines!  
day!
```

Challenge 2:

```
Tomorrow! is! valentines! day!
```

# Split a string based on space

Input:

```
>>> sentence = "Tomorrow is valentines day"
```

Challenge 1:

```
Tomorrow!  
is!  
valentines!  
day!
```

Solution 1:

```
>>> for word in sentence.split():  
    print(word + "!")
```

Challenge 2:

```
Tomorrow! is! valentines! day!
```



# Split a string based on space

Input:

```
>>> sentence = "Tomorrow is valentines day"
```

Challenge 1:

```
Tomorrow!  
is!  
valentines!  
day!
```

Solution 1:

```
>>> for word in sentence.split():  
    print(word + "!")
```

Challenge 2:

```
Tomorrow! is! valentines! day!
```

Solution 2:

```
>>> for word in sentence.split():  
    print(word + "!",end=" ")
```

# Replace characters in a string

Input:

```
sentence = "I was snowed in at home yesterday."
```

Challenge 1: 'I was snowed in at home yesterday?'

Challenge 2: 'I was snowEd in at home yEstErday.'

# Replace characters in a string

Input:

```
sentence = "I was snowed in at home yesterday."
```

Challenge 1: 'I was snowed in at home yesterday?'

Solution 1: `sentence.replace(".", "?")`

Challenge 2: 'I was snowEd in at home yEstErday.'

# Replace characters in a string

Input:

```
sentence = "I was snowed in at home yesterday."
```

Challenge 1: 'I was snowed in at home yesterday?'

Solution 1: `sentence.replace(".", "?")`

Challenge 2: 'I was snowEd in at home yEstErday.'

Solution 2: `sentence.replace("e", "E")`

Putting it all together:

random colors

# Pick a random color program

- + Ask the user for a list of their favorite colors
- + Color should be separated by commas (no space)
- + Choose a color from this list at random and print it

## Examples:

```
>>> ===== RESTART =====
>>>
Enter your favorite colors, separated by commas: pink,red,aqua,blue
pink
>>>
>>> ===== RESTART =====
>>>
Enter your favorite colors, separated by commas: pink,red,aqua,blue
blue
```