

# **Learning Python with CS Circles, CEMC's Amazing Online Tool for Students (lab)**

Peter Beens  
@pbeens  
pbeens@gmail.com

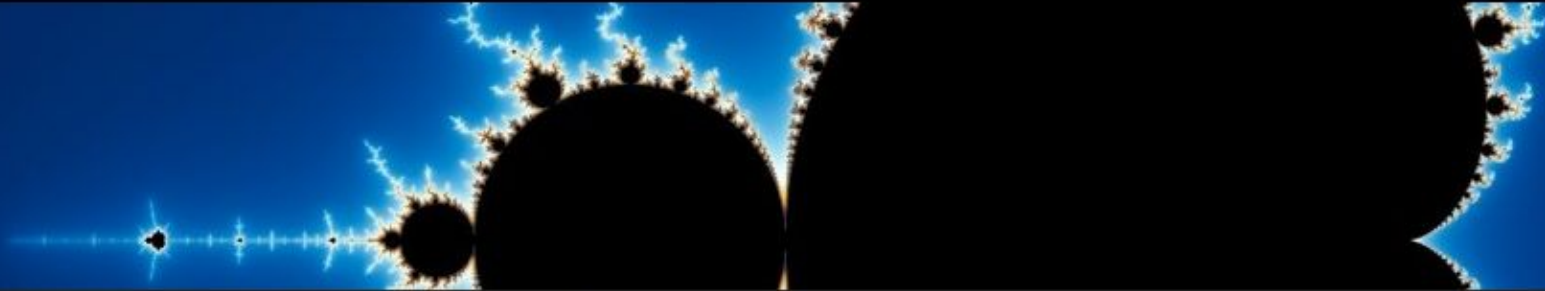
**<https://goo.gl/ZEeg1t>**

**www.acse.net**

**2.5 hours =  
0.25 hours (max) talk +  
2.25 hours play**

- Fantastic, interactive tool for learning Python 3
- All you need is your web browser (Python does not need to be installed)
- “Guru” feature allows you to monitor your students’ progress and assist them remotely
- Self-paced, for students with differing abilities
- Text is written in plain, non-academic language (i.e., for novices)
- Ideal for flipped classroom use (lessons at home, practical work at school)
- The technical support is fantastic!

Why CS Circles?



[Using This Website](#) [Console](#) [Visualize](#) [Python At Home](#) [Resources](#) [Cheatsheet](#) [Contact](#) [Thanks](#) **[My Progress](#)**

[0: Hello!](#) [1: Variables](#) [2: Functions](#) [3: Comments, Quotes](#) [4: Types](#) [5: Input](#) [6: If](#) [6D: Debug](#) [7: Rich](#) [8: Remix](#)

[9: Else, And, Or, Not](#) [10: def](#) [11](#) [12: Tips](#) [13: Lists](#) [14: Methods](#) [15](#) [16: Recursion](#) [17: Is](#) [18: Efficiency](#)

Overview of CS Circles Concepts



Using This Website   Console   Visualize   Python At Home   Resources   **Cheatsheet**   Contact   Thanks   **My Progress**

0: Hello!   1: Variables   2: Functions   3: Comments, Quotes   4: Types   5: Input   6: If   6D: Debug   7: Rich   8: Remix

9: Else, And, Or, Not   10: def   11   12: Tips   13: Lists   14: Methods   15   16: Recursion   17: Is   18: Efficiency

Keep the cheatsheet open in another tab as you learn!

Here is the first line of **Python** code in our lessons.

### Example

This is a sample **Python** program. Press the **Run program** button to see what it does.

```
print("Hello, World!")
```

Run program

When you run a program you also get to see the *output*. The example program above only has one command, `print("Hello, World!")` and it created one line of output,

```
Hello, World!
```

Here is an analysis of this first program:

- `print` is the name of a Python command, which sends messages to output.
- The parentheses `()` after the `print` command are used to contain what you want to print.
- The quote marks `" "` are used as a container for the text `Hello, World!` inside. Without quotes, Python would think that `Hello` was meant to be a command. This would be cause an error since no such command exists.

Lots of easy-to-understand explanations to learn the basics



## Exercise

Our first exercise is given below. It asks you to write a program similar to the first one above. Type your program into the box, and when you want to test whether your program correctly performs the task, press the **Run program** button. If you don't get it correct on the first try, edit the program and try running it again.

*Coding Exercise: Bonjour* 

Write a program which prints

Bonjour, tout le monde!

(Hint: if you get stuck, copy the first program above and then edit it.)

Click on this box and type to enter your code

Run program

History

Help

Lots of exercises to develop skills

If you assign a project and get  
back 30 of the exact same  
thing, that's not a project,  
that's a recipe.

Chris Lehmann

(graphic from Alfred Thompson's blog post entitled "Projects and Recipes in  
Computer Science Classes" at <http://goo.gl/iQjFQc>)

<http://www2.beens.org/ics/python/cs-circles-tutorial>

(<http://goo.gl/QyGFKF>)

My approach to keeping CS Circles from being too much like a recipe...

## Create an Account

In the top right corner, you'll see a link to create an account. Click on this link and create an account using your proper name and class email address (your class Google ID).

Add "pbeens2015" (without the quotes) as your guru's username. This will enable me monitor your progress and help when needed.

## 0. Hello

Read and do the exercises in this section. When complete, see Mr. Beens to ensure a) that you have properly added him as your "guru", and b) that your mark has been properly recorded and shared.

## 1. Variables

Read and do the exercises in sections 1 and 1E, then see your instructor to have your mark recorded.

## 2. Functions

Read and do the exercises in this sections 2 and 2X, then see your instructor to have your mark recorded.

## Challenges

- [Text Messaging \(POTW\)](#) (math) ([doctest](#))
- [Will This Hamper Your Thinking? \(POTW\)](#) (math)

From my website -- typical instructions for my students. Note the additional challenges.

## Create an Account

In the top right corner, you'll see a link to create an account. Click on this link and create an account using your proper name and class email address (your class Google ID).

Add "pbeens2015" (without the quotes) as your guru's username. This will enable me monitor your progress and help when needed.

## 0. Hello

Read and do the exercises in this section. When complete, see Mr. Beens to ensure a) that you have properly added him as your "guru", and b) that your mark has been properly recorded and shared.

## 1. Variables

Read and do the exercises in sections 1 and 1E, then see your instructor to have your mark recorded.

## 2. Functions

Read and do the exercises in this sections 2 and 2X, then see your instructor to have your mark recorded.

## Challenges

- [Text Messaging \(POTW\)](#) (math) (doctest)
- [Will This Hamper Your Thinking? \(POTW\)](#) (math)

Students adding you as “guru” lets you monitor and assist them.

### List of all problems (with #submissions)

0: Hello! (✓)

1: Variables (✓) (✓) (✓) (✓)

1E: Errors   

2: Functions (✓) (✓) (✓) (✓) (✓)

2X: Extra Practice (✓) (✓) (✓) (✓) ( )

### 3: Comments and Quotes

4: Types (✓) (✓)

5: Input 

6: If  <sub>2</sub> <sub>1</sub> <sub>1</sub>

## 6D: Design, Debugging and Donuts

7A: Strings      

7B: Math        

7C: Loops (✓) (✓) (✓) (✓) (✓)

[REDACTED]

62 completed

563 submissions

[REDACTED]

37 completed

315 submissions

[REDACTED]

34 completed

284 submissions

[REDACTED]

69 completed

535 submissions

[REDACTED]

69 completed

509 submissions

[REDACTED]

35 completed

209 submissions

[REDACTED]

71 completed

936 submissions

[REDACTED]

37 completed

288 submissions



## List of all problems (with #completed)

Click on the ( ) to drill down.

0: Hello!	( ) 8
1: Variables	( ) ( ) ( ) ( ) 8 8 8 8
1E: Errors	( ) ( ) ( ) 8 8 8
2: Functions	( ) ( ) ( ) ( ) ( ) 8 8 8 8 8
2X: Extra Practice	( ) ( ) ( ) ( ) ( ) 8 8 8 8 8
3: Comments and Quotes	( ) ( ) ( ) 8 8 7
4: Types	( ) ( ) 8 8
5: Input	( ) 8
6: If	( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) 8 8 8 8 8 8 8 8
6D: Design, Debugging and Donuts	( ) 3
7A: Strings	( ) ( ) ( ) ( ) ( ) ( ) 4 4 4 4 4 4
7B: Math	( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) 4 4 4 4 4 4 4 4
7C: Loops	( ) ( ) ( ) ( ) ( ) 4 4 4 4 3



```
1 oldList = ["first", "second", "third"]
2 newList = list(oldList)      # create a copy
3 oldList.reverse()
4 print(newList, oldList)
```

[Edit code](#)

<< First < Back Program terminated Forward > Last >>

line that has just executed next line to execute

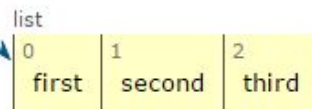
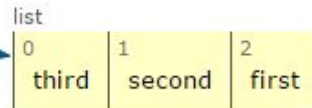
Program output:

```
['first', 'second', 'third'] ['third', 'second',
```

Frames



Objects



Generate URL

<http://cscircles.cemc.uwaterloo.ca/visualize/#code=oldList+%3D+%5B%22first%22,+%22second%22,+%22third%22%5D>

Excellent visualizer

## Console

### Console

You can test programs in this console.

```
1 n = 2
2 print (n + 3 * 4)
```

[Run program](#)[Enter input](#)[Simple editor](#)[Visualize](#)

More actions... ▾

Program executed without crashing.

Program gave the following output:

14

</>

Console lets you try (and visualize) your own test programs

# Time to play!

- <http://cscircles.cemc.uwaterloo.ca>  
(<http://goo.gl/YfWMYa>) (or Google “CS Circles”)
- Create an account to track your progress, and optionally add pbeens2015 as your guru (can be changed or removed later)
- This presentation (with links) can be found at  
<http://www2.beens.org/teacher-pd/cs-circles-2015>  
(<http://goo.gl/Hvn2ip>)