



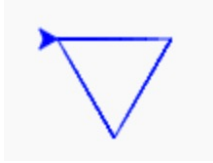
Variables & Libraries

storing data & quickly adding functionality

Challenge Overview

- **Day 1:** Getting started

Hello, World!

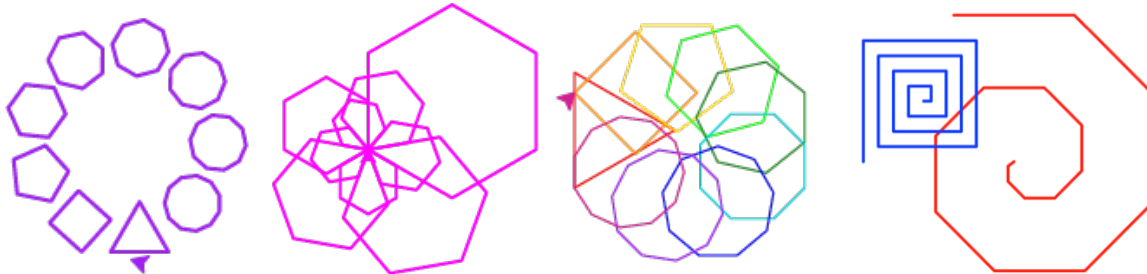
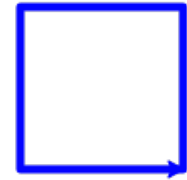


- **Day 2:** Variables & Turtle

- **Day 3:** Loops

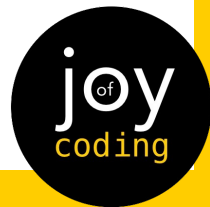
- **Day 4:** Writing Functions

- **Day 5:** Putting it all together!

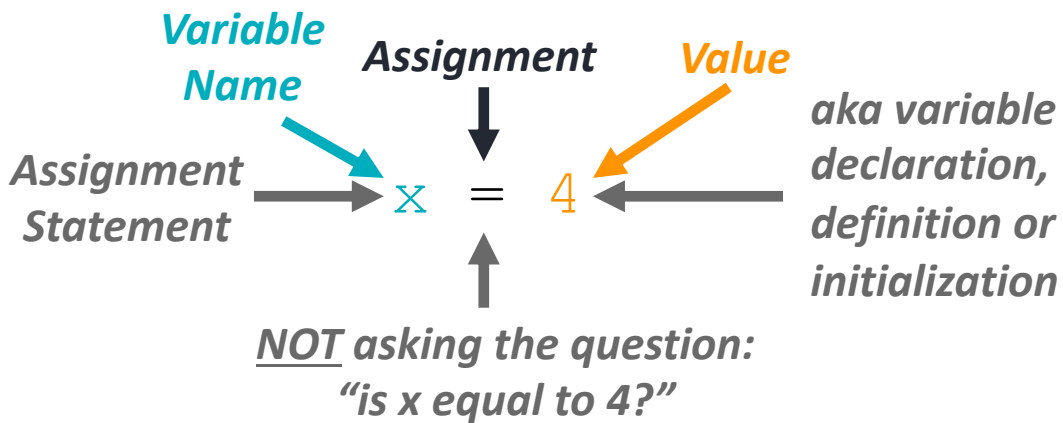


Why Variables?

- **Variables** allow us to **store data** in memory (or RAM) so that we can work with this information in our programs
- For example:
 - calculate the average run time or speed of 5 runners
 - calculate the amount of sales tax on a purchase
 - display text to the screen (like in “Hello, World!”)
- Help us make our code easier to change & modify
 - Make changes to one location & use in many
 - Saves time & frustration!



Defining Variables



Memory (aka RAM)

Variable	
Name	Value
x	4

Defining Variables

x = 4
y = 2 + 3
z = x + y
z += 1



Equivalent to:

z = z + 1

Memory (aka RAM)

Variable

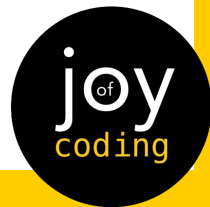
Name Value

x	4
z	10 + 1
y	5

Variables: Names & Types

- Variables have **names** (aka identifiers)
 - Have **meaningful** names (i.e., not x or y unless storing coordinates)
 - **Use** letters, numbers, or _
 - **Cannot** be a keyword like **for**, **def**, etc. or begin with a number
 - **Typically** begin with a lower case letter
- Variable values have **types**:

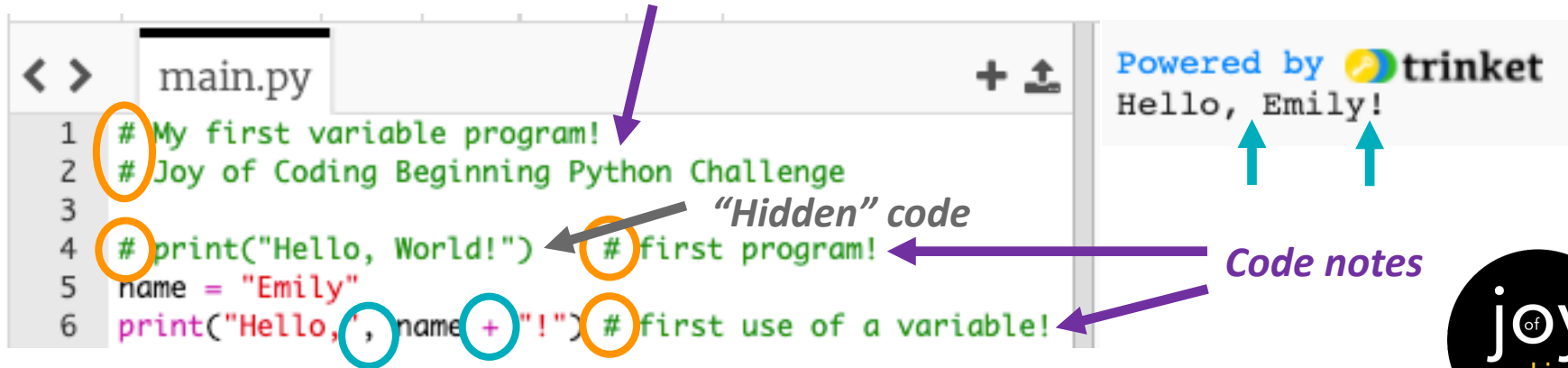
○ string	'hello'	"c"	"That's me!"		
○ int (integer)	4	237	-58	0	923,872,355
○ float (floating point)	-2.5	55.67	346.978001	0.00000017	
○ complex	4+6j	2+5j			
○ boolean	True	False			
○ objects or collections	lists	arrays	files	dictionaries	turtles



Comments


Comments allow us to write notes to ourselves and others about our code & save “hidden” code for later

*Best practice to begin
with a description of
your code & author info*



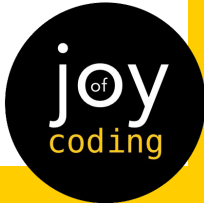
The screenshot shows a code editor window titled 'main.py'. The code contains several lines with comments. Annotations include orange circles around the first two lines of comments, a purple arrow pointing to the first comment line, a grey arrow pointing to the third comment line labeled '“Hidden” code', and a purple arrow pointing to the fourth comment line labeled 'Code notes'. The code itself includes a print statement, a variable assignment, and a formatted string print statement. The right side of the editor shows a preview of the output: 'Hello, Emily!'.

```
< > main.py + ↗  
1 # My first variable program!  
2 # Joy of Coding Beginning Python Challenge  
3  
4 # print("Hello, World!")  
5 name = "Emily"  
6 print("Hello, ", name + "!")  
# first program!  
# first use of a variable!
```

Powered by  trinket
Hello, Emily!

Why Libraries?

- The **fastest way to build programs** is by taking advantage of code that others have already written, tested, & debugged in **libraries**
 - Saves time & frustration!
 - Different time investment (learning & writing less vs writing more & testing)
- In python, libraries are organized into **modules**, like:
 - math *sqrt, floor, ceil, log, sin, cos, tan, pi, e*
 - statistics *mean, median, mode, stdev*
 - random *generate random numbers within a given range*
 - webbrowser *display web documents to user*
 - re *regular expression matching for processing text*
 - turtle *simple commands to draw shapes on screen*



Using Libraries: Import

In python, there are **two ways to import** library functions:

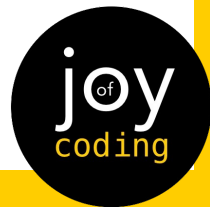
import module **VS** **from** module **import** function

```
import math
```

```
print(math.sqrt(25))
```

```
from math import sqrt
```

```
print(sqrt(25))
```



Using Libraries: Import

In python, there are **two ways to import** library functions:

import module **VS** **from** module **import** function

Keyword
↓
import

Module Name
↓
math

```
print (math.sqrt (25))
```

Module Name *Library Function Call* *Library Function*

↑ ↑ ↓

Keyword *Module Name* *Keyword* *Library Function*

↓ ↓ ↓ ↓

from math **import** sqrt

```
print (sqrt (25))
```

Library Function Call

↑



Let's try it!

- Go to trinket.io/turtle
 - Not working? Try repl.it/languages/python_turtle or skulpt.org
 - Remove any code that you didn't write
- Draw a triangle using python turtle!
- See worksheet for details & video solution:
bit.ly/beginning-python-challenge-day2
- See you in Day 3: Loops!



