WELCOME TO

TECHNOVATION

Week 3 - October 16

MICHIGAN STATE UNIVERSITY

Agenda

- Spotlight 10 - 10:15
- Lesson 2 Continued 10:15 - 10:45
  - Review
  - Variables
  - User Inputs
- Coding challenges 10:45 - 11:15
- Standup 11:15 - 11:30
  - Temperature Check

Spotlight

What Most Schools Don’t Teach

- Why do you want to learn to code?
- How do you hope to use computer science in the future?

Feedback!

Review - Week 1

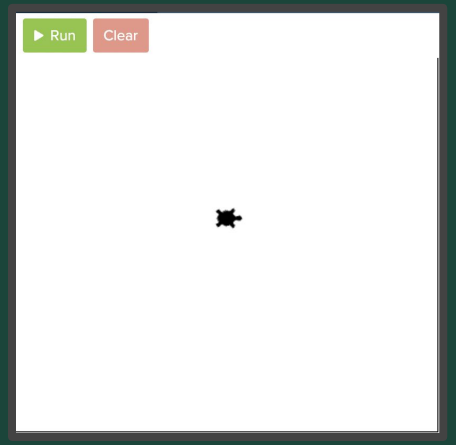
Command	What does it do?
`forward(distance)`	Moves Tracy forward a specified ***distance***
`circle(radius)`	Draws a circle with a specified ***radius***
`backward(distance)`	Moves Tracy backward a specified ***distance***
`penup()`	Stops Tracy from leaving a trail
`pendown()`	Has Tracy start drawing a trail
`left(num)`	Turns Tracy ***num*** degrees to the left
`right(num)`	Turns Tracy ***num*** degrees to the right

Review - Week 2

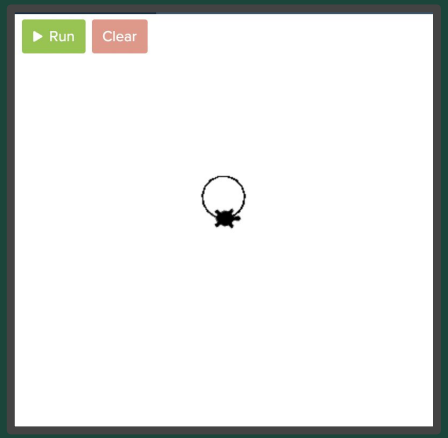
Command	What does it do?
`color("color name")`	Changes Tracy's trail color
`pensize(number)`	Changes Tracy's trail thickness
`begin_fill()`	Starts tracking closed shapes
`end_fill()`	Fills & stops tracking closed shapes
`setposition(x, y)`	Moves Tracy to the input coordinates
`speed(number)`	Sets how fast Tracy executes commands
`name = value`	Saves the value in the variable

Tracy command: Assignment

```
name = value
```



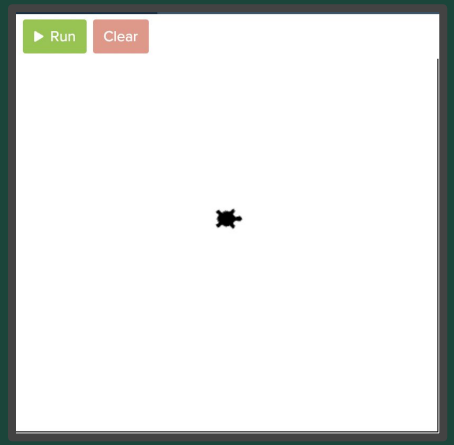

```
radius = 20
circle(radius)
```



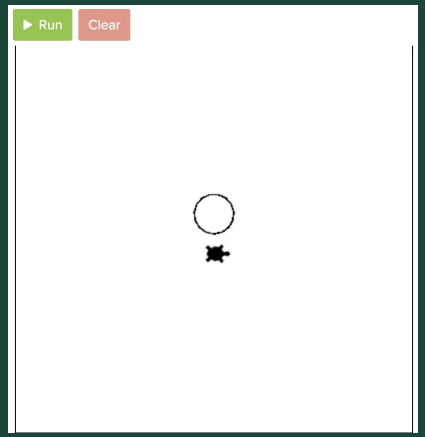
More About Variables

A **variable** is like a “box” for a “value” (number or text)

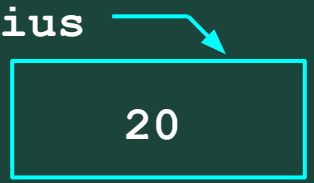
What's the Deal With Variables?



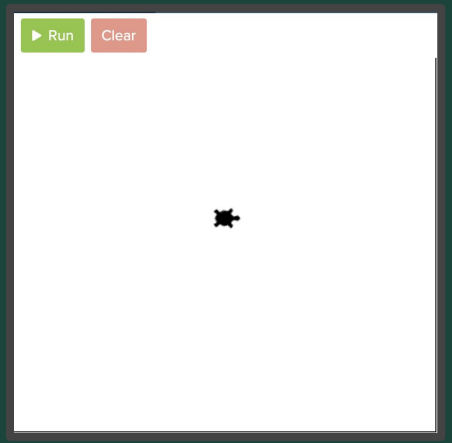

```
radius = 20
circle(radius)
penup()
setposition(0,-radius)
pendown()
```



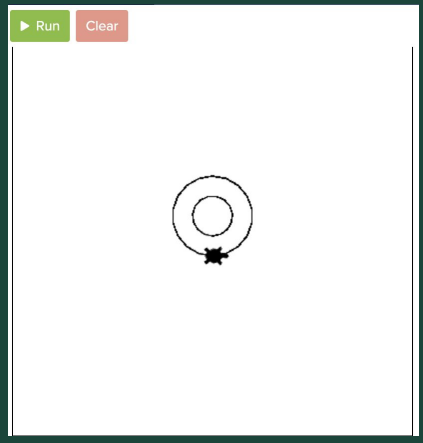
Variables let us set a value *once*, and then use that same value over and over!



What's the Deal With Variables?

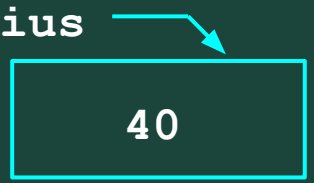



```
radius = 20
circle(radius)
penup()
setposition(0,-radius)
pendown()
radius = 2*radius
circle(radius)
```

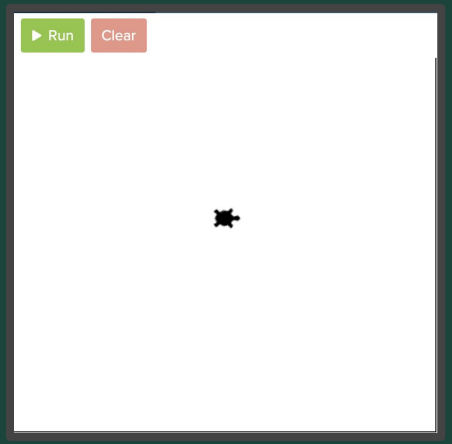


Variables let us set a value *once*, and then use that same value over and over!

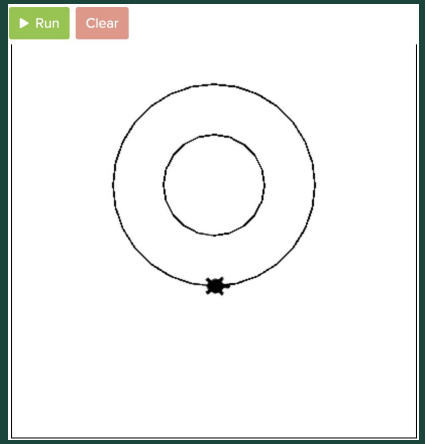
Variables can be modified to store different values over time!



What's the Deal With Variables?




```
radius = 50
circle(radius)
penup()
setposition(0,-radius)
pendown()
radius = 2*radius
circle(radius)
```



Variables let us set a value *once*, and then use that same value over and over!

Variables can be modified to store different values over time!

Variables make it easier to read and modify code!

Rules for Naming Variables

A V**ariable Name**:

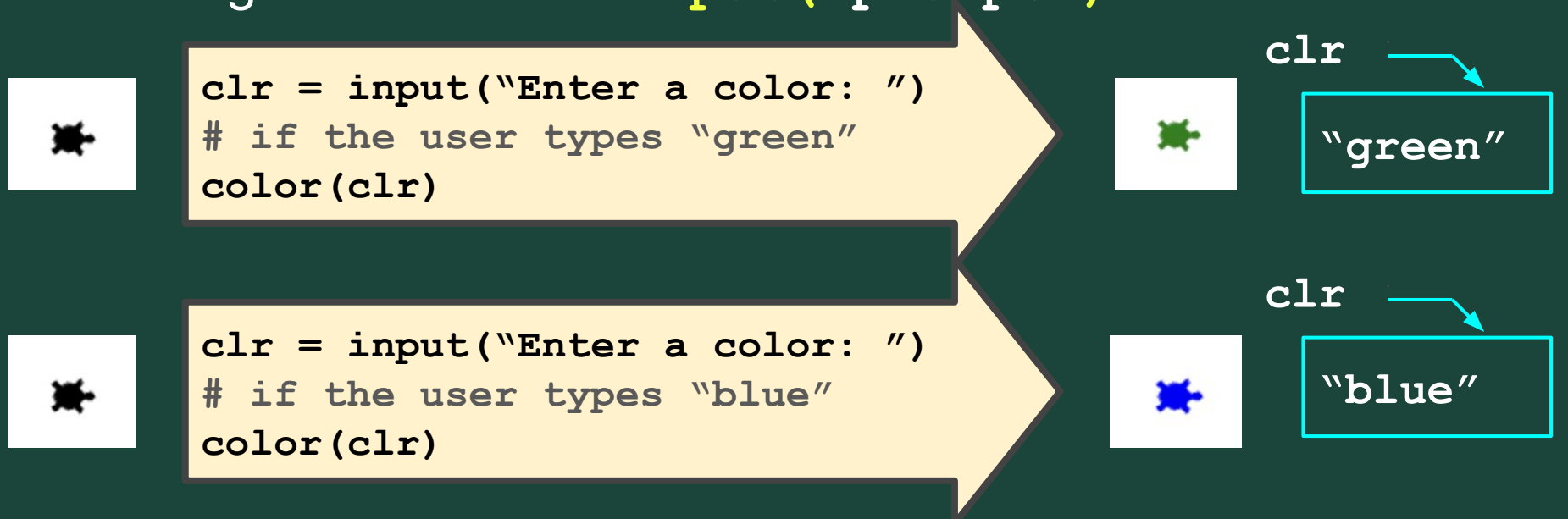
- Should start with a letter
- Consists of letters, numbers, and underscores (‘_’)
- Cannot be a Python “reserved word”
  (e.g., `int`, `float`, `input`, `print`, …)
- Should be meaningful – for example:

```
radius          speed          hair_color
```

User Input (Text)

Variables can also be used to save *user input*, letting the user indicate values on-the-fly while running programs!

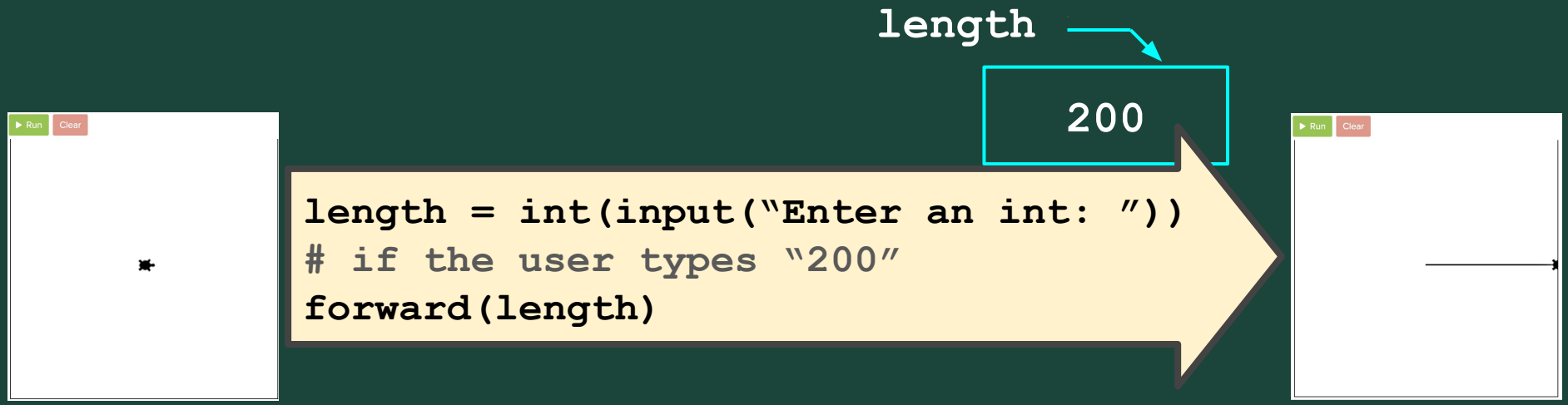
Use assignment with the `input("prompt")` function!



User Input (Numbers)

The user inputs a “string”, or sequence of characters.

To use a user input as a number, you have to tell Python to convert it to an “int” or a “float” – use `int(...)` or `float(...)`



Command	What does it do?
`color("color name")`	Changes Tracy's trail color
`pensize(number)`	Changes Tracy's trail thickness
`begin_fill()`	Starts tracking closed shapes
`end_fill()`	Fills & stops tracking closed shapes
`setposition(x, y)`	Moves Tracy to the input coordinates
`speed(number)`	Sets how fast Tracy executes commands
`name = value`	Saves the value in the variable
`input("prompt")`	Prints prompt and waits for user input
`int(...), float(...)`	Converts a value to a number (`int` or `float`)

Standup

- What is a challenge you faced today?
- What is something you were successful with?
- What do you want to improve on for next week?
- What are you most excited to learn?

Attendance and Temperature Check

Temperature Check