# Swift, an Introduction

Setup, Demo, Variables, Types, print()

Donald Pinckney and Ash Dreyer
djpinckney@ucdavis.edu, dreyershelby@gmail.com

# Table of Contents


- Administration
- What is Swift, and Why?
- Environment Setup
- Demo
- Constants
- Types
- print()

# Administration

- Introductions: who is here?
- Fill out team registration if you haven’t!
- Sign up for GitHub: https://github.com/join?source=header-home
- All questions, announcements, resources, etc. for programming training on Piazza
  - Invite email will be sent out to everyone tonight
- Email Donald or Ash for personal questions, or if you don’t get the invite
  - djpinckney@ucdavis.edu, dreyershelby@gmail.com

# What is Swift, and Why?

“Swift 3 - The powerful programming language that is also easy to learn”

*–Apple*

- Transitions well to App Programming (uses Swift + Java)
- Transitions well to Robot Programming (uses C++)

# Environment Setup

# Follow **ALL** the Directions Here:

https://github.com/frc1678/vagrant-box

# Demo “vagrant up”

# Constants

# What is a Constant?

- Name Attached to a Value
- The Value Can Not Be Changed After Creation

# Examples

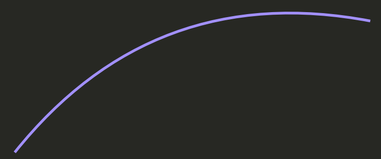
```
let a = 5

let b = 5 + 9

let c = 8.72

let name = “Donald”
```

# Parts of a Constant

```
let b = 5 + 9
```

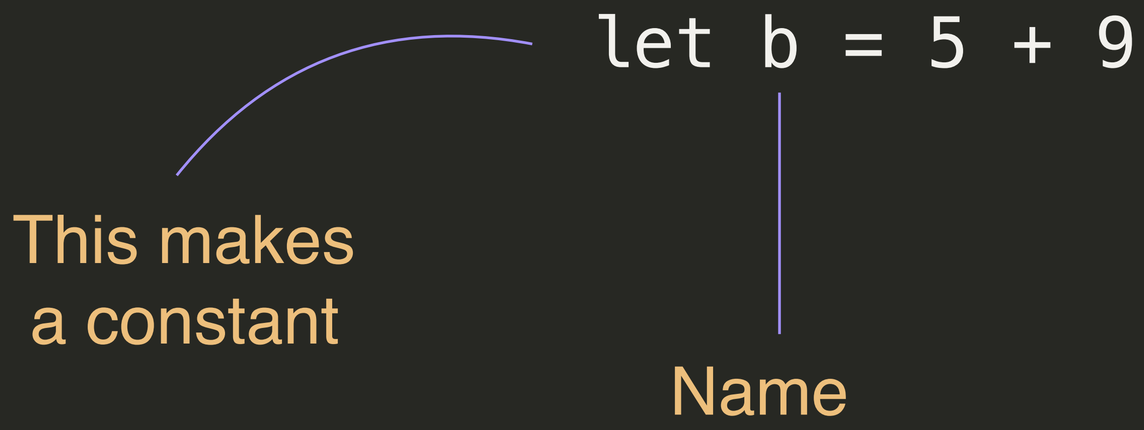
# Parts of a Constant

```
let b = 5 + 9
```

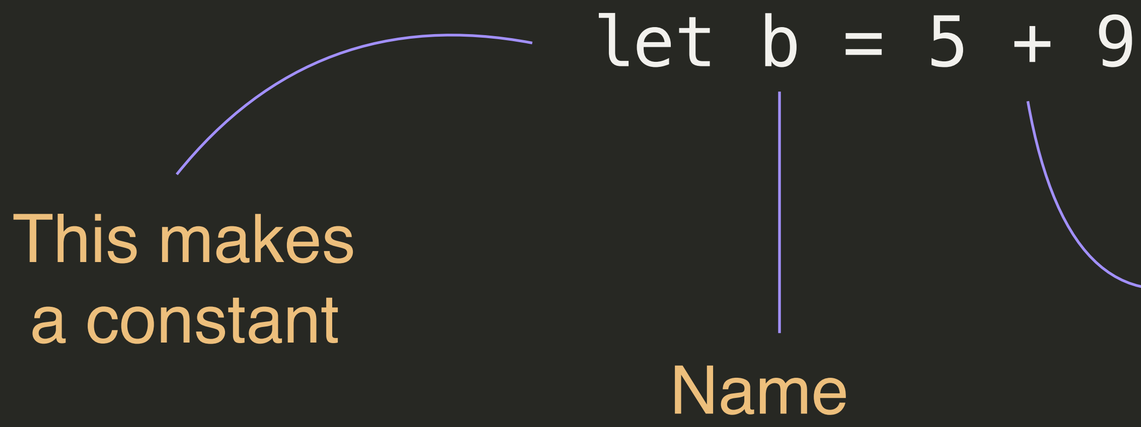


This makes
a constant

# Parts of a Constant



# Parts of a Constant



# Types

# What Are Types?

- Categories for different kinds of data
  - Int: 5, 9, 2, -8, etc.
    - Whole numbers only!
  - Double: 1.3, 8, -98.3, etc.
    - Any decimal number!
  - String: “Ash”, “🐱”, “Donald”
    - Text only!

# **EVERY** Constant Has a **Type**

- Swift guesses the **type** of your constant
  - This makes your code clearer and more concise
- You do need to understand types, since this doesn't always work
- Example:

```
  1> let a = 9
a: Int = 9
```

# Demo Constants and Types

# Sometimes Swift Guesses Wrong

```
  1> let angle = 45
angle: Int = 45
  2> let delta = 10.5
delta: Double = 10.5
  3> let angle2 = angle + delta
error: repl.swift:3:20: error: binary
operator '+' cannot be applied to operands of
type 'Int' and 'Double'
```

# Tell Swift the Type Directly

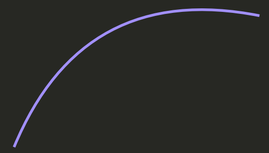
```
   1> let angle: Double = 45
angle: Double = 45
   2> let delta = 10.5
delta: Double = 10.5
   3> let angle2 = angle + delta
angle2: Double = 55.5
```

# Parts of a Constant Revisited

```
let b: Int = 5 + 9
```

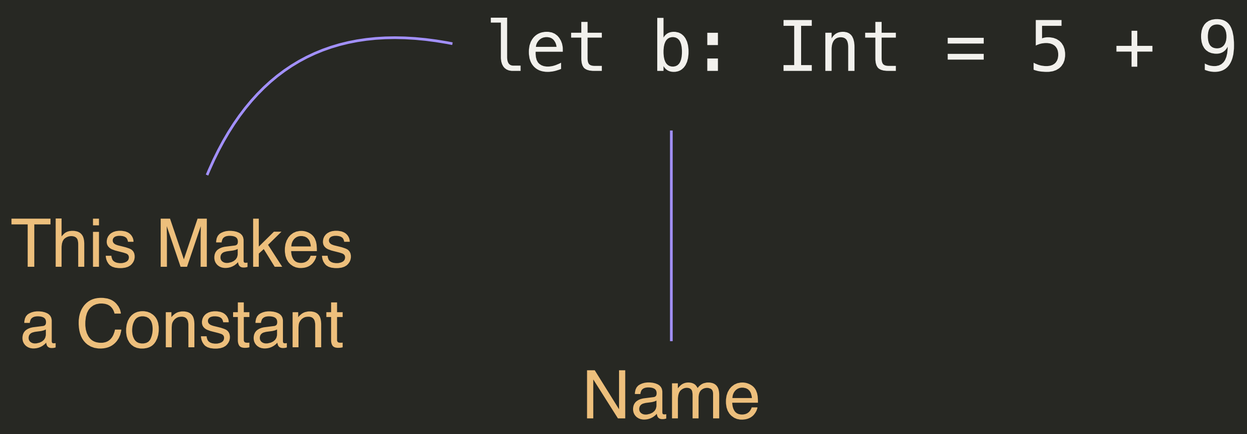
# Parts of a Constant Revisited

```
let b: Int = 5 + 9
```

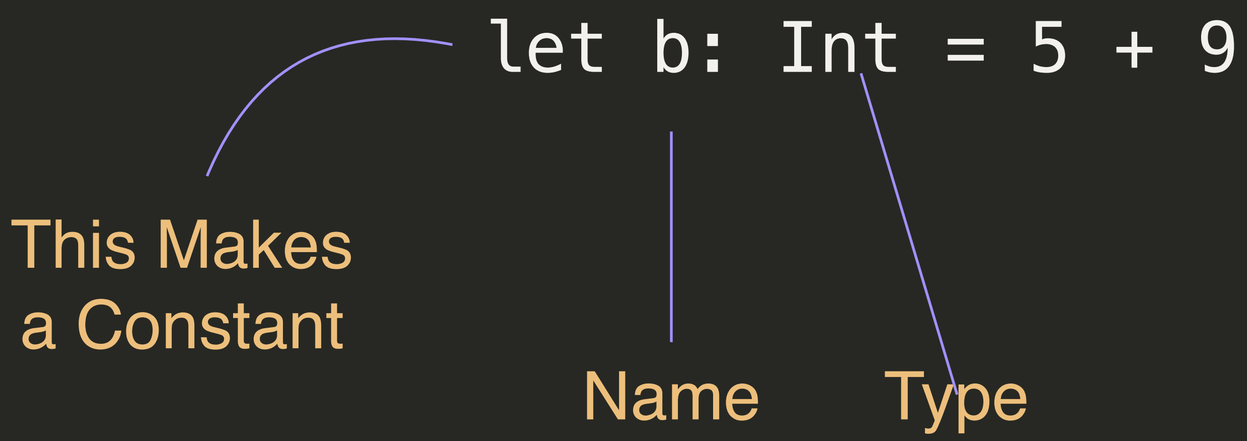


This Makes
a Constant

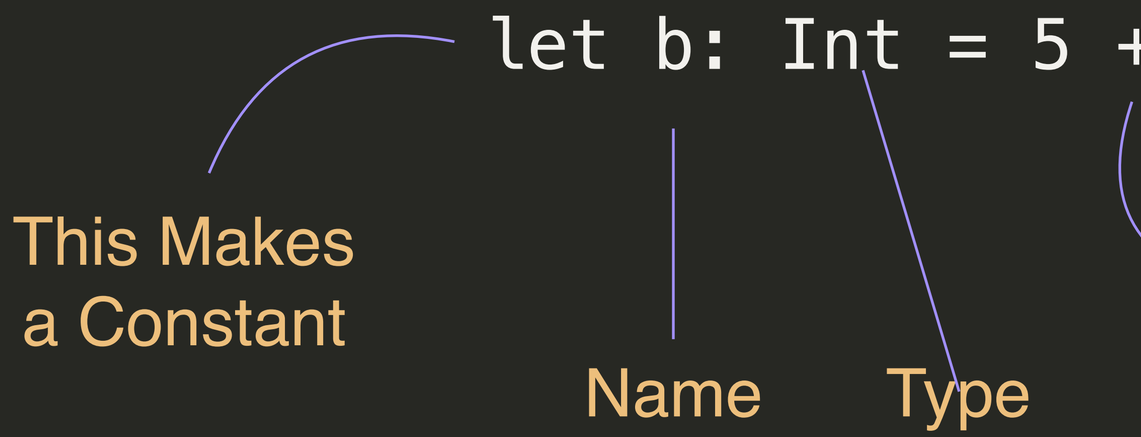
# Parts of a Constant Revisited



# Parts of a Constant Revisited



# Parts of a Constant Revisited



# What Types Are There?

- MANY MANY MANY TYPES!!!
- For Now, Only Know:
  - `let a: Int = 6`
  - `let b: Double = 4.2`
  - `let c: String = “Donald”`

# Program Output

# print() Command for Output

```
   1> print("Donald")
Donald
   2> print(5*98.4)
492.0
   3> print("Donald is \(21 * 12) months old")
Donald is 252 months old
   4> let team = 1678
team: Int = 1678
   5> print("Donald is \(21 * 12) months old, and mentors
team \(team)")
Donald is 252 months old, and mentors team 1678
```

# How to Actually Run Code

# 2 Ways to Run Code

- For experimenting, use the REPL
  - Run the command “swift”, and then type code line by line to experiment
  - Type “:q” and hit enter to quit
- For final programs use a file
  - Write your code in a file, for example, “hw1.swift”
  - Then run the command “swift hw1.swift”, or whatever your file is named.
  - If you use ~/Documents in vagrant, it will be synced to your computer.
  - See demo screenshot walkthrough, which will be emailed out.