Attendance:
tiny.cc/event-attendance

App Dev League

Day 1: Intro to Python

Ice Breakers

- ➔ Name
- ➔ Grade
- ➔ Fun Fact

Type in chat!

Agenda

1. Course Logistics
2. Python Basics
3. Kahoot
4. Hangman Project

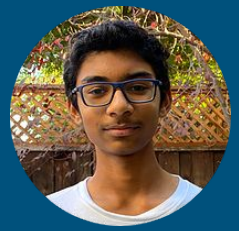
Course Logistics

Important Reminders

- ➔ June 21 to July 21
- ➔ 4:30pm -- 6:30pm (Pacific Time)
- ➔ Please come 2-5 minutes early to every session
- ➔ To get a certificate, you must attend at least 80% of sessions
- ➔ Attendance form [tiny.cc/event-attendance] at start of every class

Course Instructors

Athreya (Day 1, 5, 7)



Ishir (Day 2, 3, 4, 6, 8)

Rules and General Info

- ➔ No disruptful talking and sidechat (including Zoom chat)
- ➔ Unless you are asking a question, don’t interrupt
- ➔ Expected to attend all days unless excused (parent must confirm)
- ➔ No formal grade associated with the class!
- ➔ It is a learning experience for you (take notes!)
- ➔ Recordings will be provided 48 hours after each session

Brief Course Roadmap

Session 1-2

Intro to Python with Projects

Session 3-4

Intro to Pygame and Pygame Basics

Session 5-6

More advanced Pygame & Projects

Session 7-8

Final Projects

Python Basics

2

Python

- High-level programming language and easy to read
- Interpreted language so it usually requires an installation (compared to HTML)
- Used in traditional programming, website development, and game development (focus of camp)

Repl.it

https://replit.com/

Printing

- The print function allows for displaying text
- Can be used to write stories, draw shapes, ask for user input
- One of the most fundamental concepts of Python

```
print(“Some text”)
```

Variables

- Variables are a way for storing information
- It helps the program remember certain things
- Convention for naming variables is `x_y`

```
message = “hello”
```

Data Types

- Data Types are essentially types of data
- A variable can hold numbers, text, decimals, etc...
- There is no explicit keyword for typing, thus Python is a “dynamically typed language”

```
name = “David”

age = 15

money_left = 93.51

is_raining = True
```

Casting (Type Conversion)

- Casting (a.k.a. Type conversion) is a way to switch between types of data
- For example, a string can be converted to an integer
- A helpful tool and is used frequently in Python

```
age_text = “15”

age_num = int(age_text)
```

```
money_int = 13

money_decimal = float(money_int)
```

Formatted Strings (fstring)

- A formatted string is way of printing strings without any concatenation needed
- Condenses code
- Casting isn't needed at all for fstrings!

```
name = "Jack"

print("My name is " + name + "!")

print("My name is {name}")


age = "15"

print("My age is " + int(age))

print("My name is {age}")
```

Input

- Using the input function creates user interaction in your program
- User interaction is important for user engagement

```
name = input(“Name: ”)

Age = int(input(“Age: ”))
```

User Form Exercise

Conditionals

- Conditionals are a way to simulate conditions in real life
- There are three types of conditional statements
- “If” and “Else if” and “Else”

```
if([insert boolean]):
    [insert statement]
```

Lists

- Lists are a way of structuring data into a sequential format
- List that stores students in the class

```
names = [“jeff”, “bob”, “jack”]

ages = [15, 34, 23, 9, 12, 15]
```

- Lists can easily hold any types of data and they can be modified

Dictionaries

- A dictionary is a way to map a key to a value
- Dictionary that stores month abbreviations

```
month_abv = {
    "Jan": "January"
}
```

- Dictionaries can be thought of almost as lists except the indices are whatever you choose

For Loops

- For loops are a way to repeat a collection of statements a fixed amount of times
- Usually used when the programmer knows how many times the loop will be run
- Can sometimes have nested loops!

```
for i in range(5):
    print(“Hello”)
```

While Loops

- While loops are a way to repeat a collection of statements until a condition is met
- Used if the programmer is unsure how many times it will run

```
isGameOver = False

while not isGameOver:

    print(“Game is still on”)
```

Calculator Exercise

Functions

- A function is a collection of code that performs a specific task
- Function that prints “hi”

```
def sayHi():
    print(“hi”)
sayHi()
```

- Functions help keep organized in chunks

Classes

- A class is a way for building your own data type that simulates attributes of a real life object
- Class that displays the attributes of a video game

```
class VideoGame:

    def __init__(self, numberLevels, isHard, yearsOld):
```

- Used everywhere in python programs and is a very fundamental concept

Objects

- An object is an instantiation of a class
- Way of calling the data type

```
flappyBird = VideoGame(8, True, 1)
```

- Objects are used in every programming language and its a very common concept we see even in the real world

Kahoot

3

THANKS!

ANY QUESTIONS?

You can find more info @

- https://www.appdevleague.org
- https://linktr.ee/AppDevLeague