

Programming For a Pitch

How to make your pitch stand out with a live demo



Introductions

Mortiz
ComSSA



Nathan
ComSSA



Beth
ComSSA



Wini
CASSA



Workshop Outline

11:00-11:10	Introduction	
11:10-12:10	Beginner workshop	Intermediate workshop
12:10-12:30	Other useful tools	

Introduction

- Workshop is intended for multiple skill levels - we want to help everyone learn to program for a pitch successfully!
- Why programming?
 - Teaches you to think
 - Great for demonstrating a concept
 - Can convince people you know how to implement an idea!

Things to keep in mind

- Make sure to do practise runs
- Have a backup!
- Focus on the novel parts of your pitch in your program - don't do something everyone has seen before
- Demo isn't a test on how good your programming skills are, instead it's a way to demonstrate an aspect of your idea

Case Study - 2021 Hackathon (First Contact)

- Problem statement - find a way to communicate with aliens
- Group pitched a solution using the rotations of atoms as a universal language, communicated by a flashing light
- To prove this pitch worked, programmed an Arduino with a flashing light that took in a string sequence and displays it in atom-language
- First place in the hackathon!



Intro to C/Python

An overview of Python

- Dynamically typed (don't mention data types)!
- Formatting based on indentation - no curly braces!
- Supports many different programming 'styles'
 - Object oriented, procedural, functional...
- Large library of modules
- User-friendly, perfect for junior developers

Fizzbuzz in Python

```
1 target_number = 50
2
3 # For the numbers 1 through to 50
4 for number in range(1, target_number + 1):
5
6     # Create the message as empty first
7     message = ""
8
9     if number % 3 == 0: # If divisible by 3
10         message += "fizz"
11     if number % 5 == 0: # If divisible by 5
12         message += "buzz"
13
14     if message == "": # If it was not divisible by 3 or 5 (the message is empty)
15         message = number # Say the number we are at
16
17     print(number, message)
```

An overview of C

- Very low-level language - no objects!
- Strongly typed
- Use structs for formatted data
 - Need to manually allocate memory to structs
- Define header files for functions
- Pointers! A big nightmare

Fizzbuzz in C

```
int main(void) {
    int target_number = 50;

    // For the numbers 1 through to 50
    for (int ii = 1; ii <= target_number; ii++)
    {
        if (ii % 3 == 0) // If divisible by 3
        {
            printf("fizz"); // Say 'fizz'
        }
        if (ii % 5 == 0) // If divisible by 5
        {
            printf("buzz"); // Say 'buzz'
        }
        if ((ii % 3 != 0) && (ii % 5 != 0)) // If not divisible by 3 or 5
        {
            printf("%d", ii); // Say the number we are at
        }

        //Go to the next line
        printf("\n"); //'\\n' is the newline character
    }
}
```

Beginners - Wini
Intermediate - Moritz

Figma

Free online design tool

- For designing prototypes, wireframes, UI/UX
- Collaborative tool
- Can create reusable UI objects



Canva

Free online graphic tool

- Easy to design captivating graphics
- Access to thousands of templates
- Collaborate and share presentations that engage and inspire audiences



Other Software Tools

Github - Cloud-based Git repository hosting service

Blender - 3D computer graphics software tool

Python tools

- Skikit-Learn, Keras, SciPy (Data Science/Machine Learning)
- Flask, Django (Web Framework)

Arduino code

- Arduino code is written in C++ with an addition of special methods and functions
- The Arduino Integrated Development Environment (IDE) is the main text editing program used for Arduino programming.
- Arduino code is referred to as **sketches**.