

RTF Girls Tech Bootcamp PYTHON Syllabus

Python Curriculum

Recommended Text Editor: *Sublime Text*

Project link:

https://docs.google.com/document/d/14_IXHkGrdaoY524W-wX26G9AKL3gVwzaGGCpPbi6o/si/edit?usp=sharing

Quiz:

Ask the kids to explain in their own words what they feel programming is and what they'd love to do with it.

Lesson 1 (Introduction):

- What is a computer?
- What is programming/coding
- What is a programming language(Relate to English as a language)
- What is Python programming language
- Why Python
- Uses of Python
- What is a Text Editor(IDLE)
- Show the kids the power of Python (Python Projects)
- Advantages of Python in the modern world
- Job opportunities for Python developers
- Inbuilt functions(`print()` and `type()`)
- `print("Hello World")`

Example: Create multiple illusive instances of each instance and ask the students to identify them. Reveal the answers using `print()` and `type()`

Exercise:

Assignment: The kids should print a short essay on themselves.

Lesson 2 (Data Types):

Revision Quiz: Create multiple illusive instances of each instance and ask the students to identify them. Reveal the answers using `print()` and `type()`

- Python Data Types (Strings, Integers, Floats, Booleans)
- Advantages of each data type
- Strings
- Integers
- Floats
- Booleans
- `type()` function

Example: Create multiple arithmetic and comparison operations and walk the kids through the thought process of solving each problem. Deliberately set up Error Traps and explain the errors to the kids.

Classwork:

Assignment:

Lesson 3 (Arithmetic and Logical Operations):

Revision Quiz: Create multiple illusive instances of each instance and ask the students to identify them. Reveal the answers using print() and type()

- Python Data Types (Strings, Integers, Doubles, Booleans)
- Advantages of each data type
- Logic Gates
- Truth Table
- Arithmetic Operators I (+, -, *, /, **)... Skip(%, +, -, /, =)
- Arithmetic Operations I
- Arithmetic Operations II(For each data type)
- Comparison Operators(<, >, >=, <=, ==, !=)
- Comparison Operations(For each data type)... (Deliberately get a TypeError)

Example: Create multiple arithmetic and comparison operations and walk the kids through the thought process of solving each problem. Deliberately set up Error Traps and explain the errors to the kids.

Classwork: The kids should represent the equation $(\frac{1}{4} * \frac{3}{2}) \div 4 + 5 \geq 15$ in Python programming.

Assignment: The kids should represent $5^2 * \frac{4}{8} - (3\frac{1}{2} + 2) \neq 7$ in python programming

Lesson 4 (String Formatting):

Revision:

Create multiple arithmetic and logical operations and walk the kids through the thought process of solving each problem. Deliberately set up Error Traps and explain the errors to the kids.

Let them attempt to solve the error(Guide the thought process and give positive reinforcement).

- Python Errors
- Python's Case sensitivity
- Python's order of execution(Top to Bottom)
- String Formatting I(.upper(), .lower(), .title())
- Logical operations with string formatting. (e.g "Henry".upper() == "Henry".lower())
- Concatenation I(using "+")
- String Formatting II("\n", "\t", "\\", "\'")

Example: Using all the string formatting learned in multiple instances(Explain the meaning of Instances), print() a short introduction of yourself.

Classwork: The kids should print

```
>>> list1 / 20 Traceback (most recent call last):
      File "", line 1, in <module>
        list1 / 20
TypeError: unsupported operand type(s) for /: 'list' and 'int'
```

utilising their knowledge of string formatting.

Assignment: The kids should print

[1, 2, 3]	[4, 5, 6]
[1]	[2, 3, 4, 5, 6]
[1, 2, 3, 4]	[5, 6]

Using their knowledge of string formatting

Lesson 5 (String Formatting):

Revision: Draw a square using String Formatting

- Python Errors
- Python's Case sensitivity
- Python's order of execution(Top to Bottom)
- String Formatting I(.upper(), .lower(), .title())
- Logical operations with string formatting. (e.g "Henry".upper() == "Henry".lower())
- Concatenation I(using "+")
- String Formatting II("\n", "\t", "\\", "\'")

Example: Open a random web page online with formatted. You and the kids should recreate the text alignment and arrangement using string formatting

Classwork: Print the multiplication table for 2 and 3 side by side

Assignment: Either draw two squares side by side or draw a triangle.

Lesson 6 (Variables):

Revision Exercise:

Using all the string formatting learned in multiple instances(Explain the meaning of Instances), print() a short introduction of yourself.

- Variables... (Make reference to variables in mathematics)
- Variable Naming Conventions
- Invalid Variables
- Arithmetic and Logical Operations using variables
- Errors (Undefined error)

Example:

Create variables to store student data, and print a short introduction.

Classwork: The kids should create variables to store their basic details such as first name, middle name, last name, age, class, school, etc. and then print sentences using all the details they collected

Assignment: Store the output of $\frac{5}{7} * \frac{3}{4}$ in a variable(a) and then store the output of $15 - 32 * (4/5)$ in another variable(b) then create a third variable(c) to store the output of $\frac{a}{b} \geq 5$.

Lesson 7 (Variables):

Revision Exercise: The kids should create variables to store their basic details such as first name, middle name, last name, age, class, school, etc. and then print sentences using all the details they collected

- Variable Reassignment I
- Pythonic variable declaration (e,g name, age = "Henry", 72)
- String formatting III(Immutability of variables to string formatting)
 - `"""Sample code"""`
 - `name = "Henry"`
 - `print(name.lower())`
 - `print(name)`
- Parsing (Data Type conversion)
- Variable Concatenation (+)
- Variable Reassignment II (Assign an existing variable to a new one)
 - `"""Sample Code"""`
 - `first_name = "Henry"`
 - `last_name = "Smith"`
 - `full_name = first_name + last_name`

Classwork:

Assignment: Using Pythonic Variable declaration store all your details in variables on a single line. After which re-assign your first name and age to that of your best friend after which you'll form sentences with them

Lesson 8 (Inputs):

- Python Input()
- Storing inputs in variables
- Input direct and indirect parsing
- Python Comments

Example: Create a Python form using inputs to gather user data and then print() a report of the input entered

Classwork :

Assignment:

Revision: Create a Python form using inputs to gather user data and then print() a report of the input entered

- Input formatting
- Python imports/ external libraries (time, random)
- `time.sleep()`, `time.time`
- `random.randint()`, `random.choice()`

Classwork: Use Inputs to gather details from a user and then store them in variables, then create a form with those details

Assignment (Automated Intelligence): The kids should create a Python form to gather specified user data using inputs. After this, they create print() statements

utilizing the gathered student data. The students then prompt users to answer questions and create fixed creative contextual comments on each user's input. **Note:** Award extra points for students who can creatively use `time.sleep()` and `random.choice()` to give comments on user inputs.

Lesson 9 (Tuples):

- Concatenation II(,)
- Advantages of the two methods of concatenations
- Python Iterables I (tuple)
- Tuple Immutability
- Tuple addition
- Indexing (Strings and Tuples)
- Advantages of Tuples

Example: Create a variable, store a tuple of names inside, and then reference each name using indexing.

Exercise: The kids should gather multiple user data using inputs and then create a story from the inputs by using the indexes of each data as well as applying string formatting(`.title()`, `.upper()`, `.lower()`) where applicable.

Assignment:

- String Formatting IV(Slicing and Splitting)
- `random.choice(tuple)`
- Tuple Inputs (User data)
- Tuple Parsing(`list()`, `tuple()`)

Example:

Exercise:

Assignment: Using the `random.choice()` function from the `random` module, the kids should create a tuple of responses to messages and then use `input()` to prompt the users. They should use `random.choice()` to select a randomized response from the tuple. The kids could also apply `time.sleep()` at will

- Multidimensional Tuples

Lesson 10 (Lists):

- Concatenation III("Hello {}".format(name))
- Python Iterables II (List)
- Multidimensional lists
- List mutability
- List `insert()`, `append()`, `pop()`
- List inputs (user data)
- List indexes
- List index re-assignment
- Uses of lists
- `random.choice(list)`, `random.sample(list)`
- Python Iterables (Sets)
- Set functions such as unions, intersections, etc
- `Len()`
- String formatting V (`.strip()`)
- List Parsing

- Advantages of lists
- Introduction to Dictionary

Example: Create three lists, the first is a list of student names, the second is a list of the students' ages, and the third is a list of their favourite food. Create an input that allows the user to enter a student's name and then check using if statements if the student's name is in the list, if yes, get the index of the name and use that index to access the student's details in the other lists. Assign the student's details to a variable as a tuple then use indexing and string formatting to give a detailed report on the student's details

Classwork:

Assignment:

Lesson 11 (Conditionals):

- Concatenation IV(f"Hello {name}")
- Python conditionals I(Match Case)
- Uses of Match Case
- Nested Match Cases

Class Exercise: Cutting across all the kids taught, the tutor and the kids should work together to create a matching case simulating the purchase of various MTN data plans, with an initial live demonstration of MTN's data purchase process.

- Python Conditionals II(If, elif and else statements)
- Concept of If statements
- Uses of If statements
- Nested Ifs

Classwork: Similar to the previous Project Classwork, the kids should create a program using if statements to simulate the MTN data purchase process. Pointing out the differences between If statements and Match Cases.

Project / Assignment: The kids should develop creative programs utilizing either Match Case or IF statements and then explain their choice of conditionals.

Lesson 12 (Loops):

- Project Presentation
- Python Loops I(For Loops)
- Uses of For Loops
- range()
- for loops using range()
- for loops on iterables
- for loops utilizing len()
- Python Loops II(While Loops)
- While Loop conditionals
- Break and Continue
- Nested Loops

- While Loops and Booleans
- Patterns using For loops and while loops
- **Sample Code**

```
for i in range(8):
    print("x" * i)
```
- If statements in Loops
- Difference between For Loops and While Loops
- Use cases For loops and While Loops

Lesson 13 (Functions):

- Python Functions
- Non - Parameterized Functions
- Parameterized Functions
- Return Statements

Exercise: Create a function to concatenate two parameters of different data types. **Must Use:** Parsing, If statements and type()

Lesson 14 (Dictionaries):

- Abstraction (Importing other python files)
- Introduction to OOP
- Python dictionaries

Example: Convert Words to emojis

Lesson 15 (Class):

- Python Classes
- Class Attributes
- Class Functions
- Inheritance
- Class Instantiation
- `__init__`
- class values (self)
- `__super__`

Lesson 16(SQL):

- What is a Database
- Uses of a Database
- Types of Database
- Introduction to Structured Query Language
- Importing sqlite3
- Connecting Python codes to a Database
- Querying a database
- What is a cursor?
- What are Tables
- Create Table
- Committing a database

- Populating a table (INSERT)
- UPDATE queries
- DELETE queries

Week 17 (GUI)

- What is a graphical user interface
- Creating a window
- Tkinter Labels
- Tkinter Entries
- Tkinter Text Box
- Tkinter Buttons
- Adding Functions to Buttons
- Tkinter Option Menu
- Tkinter ComboBox

Week 18 (Tkinter and SQL):

- Collecting data from the Front End
- Storing Front End data in database
- Creating multiple windows in Tkinter
- Tkinter Message Box
- Passing data from backend to front end
- Dynamic Labels
- Passing Data From Database to Front end

Robotics Curriculum

Quiz:

Ask the kids to explain what they feel robotics is all about

LESSON 1: INTRODUCTION

Topic 1: Introduction to Robotics and Robots

- What is Robotics?
- What is a Robot?
- Examples of Robots
- Relationship between human and Robots
 - Environment sensing
 - Decision making
 - Movement
- Show the kids some robots and their application
- Basic features of a robot
- Ask the kids to guess what hardware components are responsible for different features of a robot (Movement, talking, sight, object recognition, etc.)

Topic 2: Introduction to Microcontrollers and Arduino

- What is a microcontroller
- Similarities Between a microcontroller & a regular computer (Input, processing, output, CPU, Memory, RAM)
- Differences between a microcontroller and a microprocessor (Speed, Data processing power etc)
- What is Arduino?
- Why Arduino?
- What can the Arduino do? (Make the student understand the unlimited possibilities)
- Introduction to the arduino starter kit

Topic 3: Identification of Hardware components in the kit

- The Arduino microcontroller
- Breadboard
- Jumper wires
- LEDs
- sensors (Temperature & Humidity, Ultrasonic, PIR, Tilt etc)
- Resistors, Transistors, Potentiometer
- Displays (LCD, 7 segment display, etc)
- Output devices (motors, servos, buzzer etc)

Example: Give multiple examples of machines and ask the students to identify the ones that are robots and ones that are not a robot

Exercise: Walk the students through how to blink the built-in LED on the arduino by uploading the blink sketch

Assignment: Write a short essay on what you will love to build using the Arduino

LESSON 2: INTRODUCTION TO ARDUINO AND ARDUINO IDE

Revision quiz:

Topic 1: Setting Up the Arduino IDE and Drivers

- Downloading the Arduino IDE & Drivers
- Installing the Arduino IDE and drivers
- Walk-around the in the Arduino IDE interface
- Identification and selection of COM port

Topic 2: Upload your first program to the Arduino Board

- Walk-around in the Arduino IDE (Selecting the correct board type and COM port)
- What is a sketch?
- The Arduino built-in LED
- Compiling and Uploading the blink sketch to the Arduino board
- Adjust the code to make the LED blink faster

Topic 3: Practice Arduino at home

- Introduction to TinkerCad
- Opening an account with tinkerCad and Joining the classroom
- Implement a circuit and write a program to blink an Arduino built-in LED

Example: Implement a Circuit on the breadboard to blink and external LED

Exercise: The kids should experiment with the delay() function to make the LED blink faster or slower

Assignment: The kids should implement the blink circuit on TinkerCard

LESSON 3:

LESSON 4: ARDUINO PROGRAMMING BASICS & BASIC ELECTRONICS

Revision Quiz:

Topic 1: Arduino Programming basics (Digital Input and Output)

- setup() and loop() functions
- Comments in Arduino sketches
- Digital and Analog inputs and Outputs pin identification
- Some basic Arduino functions
 - pinMode()
 - digitalWrite()
 - digitalWrite()
 - delay()
- Explain Time functions delay(), millis(), when to use and how to use them

Topic 2: Basic Electronics

- Explain Digital inputs & Outputs pins
- Show the Digital I/O on the Arduino board
- The breadboard and how to use it
- Teach the kids the basics of building a circuit (by identifying Input devices, output devices, power supplying)
- Demonstrate how to implement a circuit on the breadboard
- Explain Voltage, Current, Resistance & Ohms Law(know when to use resistors e.g for LEDs)
- Understanding Logic levels (HIGH and LOW, 0 and 1, 0V and 5V)Understanding Levels (HIGH and LOW, 0 and 1, 0V and 5V)
- Understanding the LED +ve and -ve pins
- How to wire up the LED with the Arduino show the circuit diagram

Example 1: Implement a circuit and Write a program to create a traffic light system using Red, Green and Blue LEDs

Example 2: From Example 1 modify the circuit and the sketch to change the brightness of the RED LEDs using a potentiometer

Classwork: Students should continue with Example 2 and simultaneously adjust the brightness of the three LEDs

Assignment: Students should implement the classwork on TinkerCad

LESSON 4: ANALOG INPUT & OUTPUTS & MORE ARDUINO PROGRAMMING BASICS

Revision Quiz:

Topic 1: Analog Input & Output

- Introduction to Analog-to-Digital Converter (ADC)
- Understanding analog Output and Pulse Width Modulation
- Show the Analog Input and PWM pin on the Arduino board
- The potentiometer as voltage divider and an analog device
- Identifying the potentiometer pins
- Wiring the potentiometer with Arduino on the breadboard (show the circuit diagram)

Topic 2: Introduction to Serial communication

- The Arduino serial Monitor
- Taking analog readings from the Serial monitor using Serial.println() function
- Reading potentiometer wiper values in the Serial monitor

TOPIC 3: VARIABLES & CONTROL STRUCTURES

- Some arduino Analog functions
 - analogRead()
 - analogWrite()
 - The range of analogRead() and analogWrite() Functions
- The map() function
- Variables and datatypes
 - int
 - char
 - double
- Variable declaration , assignment and usage
- Control Structures
 - if statement
 - for loops
 - while loops
- Arithmetic, Comparison and logical operators
 - Arithmetic operators (+, -, *, /)
 - Comparison operator(<, >, <=, >=, ==, !=)
 - Logical operators (&&, ||, !)

Example 1: Design a circuit to Turn ON and OFF an LED using a push button switch

Example 2: (Light dimmer) Design a circuit and a program that adjusts the brightness of an LED using the potentiometer

Classwork: The students should implement a circuit and write a code that controls two LEDs with a push button switch

Assignment: The students should implement and experiment with the classwork on TinkerCad

Lesson 5: INTRODUCTION TO SENSORS

Revision:

Topic 1: Introduction to Senors

- Tupes of sensors
 - Light sensor
 - Temperature sensor
 - Tilt sensor
 - Ultrasonic sensor
- Sensors as input devices
- Understanding sensor pinouts (power pins, and output pin)

- Interfacing sensors with the Arduino
- Interfacing Light sensor with the Arduino
- Interfacing Temperature sensor with the Arduino

Topic 2: Arduino functions and Libraries

- What are functions
- Functions return type and argument
- How to declare a function
- How to call a function
- What are Arduino libraries
- How to install Arduino libraries
- How to use libraries

Example 1: Design a darkness detector using the light sensor (LDR)

Example 2: Design a temperature sensor to read temperature on the serial Monitor, a buzzer should start beeping if the temperature goes beyond a certain value

Classwork: The students should design a code a circuit that beeps a buzzer if a tilt sensor is touched

Assignment: The kids should get creative with their knowledge of sensors and implement their ideas on TinkerCad

LESSON 6: DISPLAY DEVICES & MORE ON SENSORS

Revision quiz:

Topic 1: Introduction to seven segment displays

- Displays as Output devices
- Types of display
 - Seven segments display
 - LCD displays
- Understanding the seven segments display
- How the 7 segment displays digits by turning LEDs ON & OFF
- Interfacing seven segments display with arduino
- Displaying digital 1 to 10 on the seven segments display

Topic 2: Introduction to Liquid Crystals Display (LCD)

- Understanding the LCD
- How the LCD display characters by controlling a 5x8 block of rectangle (pixels)
- The LCD pinout
- Wiring the LCD with the Arduino
- Interfacing Arduino with LCD
 - Installing the LCD library
 - Creating the LCD object
 - Understanding the LCD library functions

- LCD.init(), LCD.backlit(), LCD.setCursor(), LCD.print(), LCD.clear()
- Displaying text and numbers on the LCD
- Scrolling characters on the display
- Controlling display brightness
-

Example 1: Experiment with the different LCD display functions

Example 2: Build a temperature and humidity display using DHT sensor and LCD display

Exercise: The kids should display analogRead() values of potentiometer and Light sensor on the LCD

Assignment: The kids should implement the Example 2 on TinkerCad

LESSON 7: MORE ON SENSORS

Revision Quiz:

Topic 1: ULTRASONIC SENSOR

- What is Ultrasonic?
- What is ultrasonic sensor
- How The ultrasonic sensor works
 - The Ultrasonic transmitter and receiver
 - The relationship between sound speed, time and distance
- Interfacing the ultrasonic sensor with Arduino to measure distance
- Taking distance reading from the Serial monitor

Topic 2: PROXIMITY SENSOR

- What is proximity sensor
- What is Infrared?
- Explain the relationship between Infrared and visible light
- How the proximity sensor works
 - Infrared Light Emitting Diode as a transmitter
 - IR receiver
- Interfacing proximity sensor with the Arduino

Topic 2: PIR SENSOR SENSOR

- What is PIR sensor?
- What is In
- The kids should talk about where they had seen a PIR sensor in action
- How the PIR sensor works
 - The relationship between Heat, Temperature, and Infrared Radiation
 - Heat signatures
- Interfacing the PIR sensor with Arduino

Example: Build a distance measuring device using device that displays distance on an LCD

Exercise: The kids should build a motion detector that turns ON an LED and a buzzer

Assignment: The kids should get creative with Ultrasonic sensor and PIR sensor, Their ideas should be implemented on TinkerCad

LESSON 8: MOTORS AND ACTUATORS

Revision quiz

Topic 1: INTRODUCTION TO MOTORS

- Types of Motors
 - DC Motors
 - Servos
 - Stepper Motors
- Motors as a crucial parts of robots
- Understanding Pulse-Width-Modulation (PWM)
 - How PWM controls the servo angle position
- Interfacing servo with the Arduino
 - Understanding the servo pinout
 - Wiring the servo with the Arduino
 - Installing the Servo library
 - Creating the servo object
 - Understanding the Servo library functions
 - `servo.attach()`, `servo.write()`, etc

Topic 2: DC Motors

- Power a DC motor with a battery
- Understanding how power and control motors with Arduino
 - Using Motor drivers
 - Using a Transistor
- Understanding Pulse-Width-Modulation (PWM)
 - How PWM is used to controls Motor Speed (Demonstrate this by asking the students power ON and OFF a DC motor and short time intervals and explain the effect of such)
-

Example: Design a circuit that controls the servo position using a potentiometer

Exercise: The student should combine their previous knowledge of PIR sensor to design an automatic door opening and closing system

Assignment: The student should combine their previous knowledge of Proximity sensor to design an automatic waste bin system on TinkerCad

LESSON 9: COMMUNICATION PROTOCOLS AND MODULES

Revision quiz:

Topic 1: Introduction to Serial communication

- What is Serial communication
- Types of Serial communication
 - UART
 - I2C
 - SPI
- Understanding UART communication
 - communication line (Tx and Rx pins)
 - Baud rate matching
- Serial communication with the Arduino
 - Talking to Arduino over the Serial Monitor
 - Using Serial.available(), Serial.read(), Serial.parseInt() etc

Topic 2: IR communication

- The IR transmitter
- The IR Receiver
- How communication works between the IR transmitter and receiver
- Interfacing the IR receiver with Arduino
 - Wiring the receiver with Arduino
 - Installing the IR receiver library
 - Understanding the IR receiver library functions
 - Decoding the individual codes of keys of the IR transmitter over the serial Monitor

Example: Design a circuit that controls three different LEDs by using IR transmitter

Exercise: The student should get creative with the IR remote and implement their ideas

Assignment: The students should Implement their idea on TinkerCad

LESSON 10: MORE ON COMMUNICATION AND MODULES

Revision quiz:

Topic 1: Bluetooth Module

- Introduction to the HC-05 (or HC-06) bluetooth module
- Understand the HC-05 pinouts
- Interfacing Bluetooth module with Arduino
 - Wiring the Bluetooth module to Arduino
 - How to use the Software serial library
 - Establishing communication between Arduino and the bluetooth module
 - Sending AT commands over the serial monitor
 - Installing the serial Bluetooth app on android device
 - Connecting the android device to the Bluetooth module

Example: Design a circuit to control a DC motor via Bluetooth on an android device

Exercise: The students should modify the code and circuit to control the speed of the Motor

Assignment: Project research

WEEK 11 - 15 : PROJECT RESEARCH AND DEVELOPMENT**WEEK 16: PROJECT PRESENTATION**

Ro