FIRIA LABS

LESSON: Pi-thon Day Project		Time: 45- 60 minutes
Overview: This lesson will celebrate pi day (March 14) with a fun project that displays the digits of pi and pi facts.		**Objectives:** ● I can add pi facts to a list. ● I can display a long string of digits across the screen. ● I can use a rectangle and circles to draw a snake head. ● I can control the CodeX with buttons.
Grades 6-8 CS Standards: **2-CS-01** Design projects that combine hardware and software components to collect and exchange data. **2-CS-03** Systematically identify and fix problems with computing devices and their components. **2-AP-11** Create clearly named variables that represent different data types and perform operations on their values. **2-AP-13** Decompose problems and subproblems into parts to facilitate the design, implementation and review of programs. **2-AP-14** Create procedures with parameters to organize code and make it easier to reuse. **2-AP-16** Incorporate existing code, media and libraries into original programs, and give attribution.	**Grades 9-10 CS Standards:** **3A-DA-11** Create interactive data visualizations using software tools to help others better understand real-world phenomena. **3A-AP-13** Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. **3A-AP-14** Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. **3A-AP-16** Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. **3A-AP-17** Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules and/or objects.	**Grades 11-12 CS Standards:** **3B-AP-10** Use and adapt classic algorithms to solve computational problems. **3B-AP-14** Construct solutions to problems using student-created components, such as procedures, modules and/or objects. **3B-AP-16** Demonstrate code reuse by creating programming solutions using libraries and APIs. **3B-AP-17** Plan and develop programs for broad audiences using a software life cycle process. **3B-AP-22** Modify an existing program to add additional functionality and discuss intended and unintended implications.
Preparation: ● Download slides ● Be familiar with the final code ● Read through the teaching guide	**In the folder:** ● Pi-thon Day project slides ● Pi-thon Day project code ○ Starter code ○ Code solutions for all steps and final	**Agenda:** ● Warm-up (5 minutes) ● Complete program using slides (40 minutes) ● Optional: Wrap-up (5 minutes)
Teacher Notes: ● This lesson is designed so that students can work independently by following the slides. However, you can also work together as a class. See the teaching guide for specific helps and hints. ● Almost all mistakes made by students are typing mistakes. If students get errors when they run their code, first look over the code for spelling, punctuation and indenting. ● The program is most interesting if all steps are completed. But if you are short on time, students can stop after any of the five steps and still have a working program.		

Teaching Guide

Warm-up (5 minutes)

This short warm-up is to research pi, get pi digits and pi facts that will be added to the program on the CodeX.

💡 **Teaching tip – warm-up**

- Show a picture of pi, or a cartoon of pi. Several are available with a quick google search.
- Have students research pi. (Slide 3)
- Students need to find the first 100 digits of pi. Copy and paste, or type, the digits in a document to refer to later.
- Students should also find some fun facts about pi. They can paste these facts into their document to use later.

Create/Run the Program (40 minutes)

💻 This project can be completed individually or with pair programming.

💡 **Teaching tip:**

This project is not included in CodeSpace. Download and follow the slides. They include step-by-step instructions as well as code snippets to guide students through the program code creation.

You can have students complete the project one of two ways:

- Show the slides on a large screen or monitor and have the class work on each step together.
- Give the slides to the students and let them work through the instructions at their own pace.

💻 **Slides 1-3 Warm-up**

Students research pi and pi facts.

💻 **Slide 4 New File**

Students get into CodeSpace and start a new file. Use the sandbox! If students do not already have an account, they can easily create one using any email address. Or they can log in as a guest, which means their code will not be saved.

💻 **Slides 5-6 Start Code**

This project has starter code for the students that they will add to. Give the starter code file to your students. The easiest way is probably to use your LMS. You can also upload the file to the cloud and insert a link to the file in the slides for students to access. Students need to open the text file and copy and paste the code into a new file in CodeSpace. They will not open the starter file in CodeSpace.

Students should run the code and make sure there are no errors. The menu will appear, but nothing else will happen ... yet!

💻 **Slides 7-8 Step #1 Pi digits and Pi facts**

Students modify the pi variable to include 100 digits (or more). Students also add their pi facts to the pi_facts list.

- What to watch for: When adding items to a list, remember to separate each item with a comma. The comma is not inside the quotation marks. The last item does not have a comma after it.
- An example of the solution code after the first step is included for download.

💻 **Slides 9-14 Step #2 Show the Digits**

Students add code to display the 100 digits of pi. The digits will scroll across the screen like the body of a snake.

- The location of the stream of digits will be randomly selected.
- Snippets of code are included on the slides to guide the students.

- What to watch for: Indenting! The new while loop is not indented, but the code inside is. Also, watch for the use of a colon (:) at the end of each while loop statement.
- The display.draw_text() line of code is a bit tricky. If you think students may type it incorrectly, you can give them the line of code.
- The solution code for this step is provided included for download.

💻 **Slides 15-17 Step #3 Stop the Program**

Students add an if statement to break the loop. A code snippet is included to guide students.

- What to watch for: Indenting! The if statement needs to be at the same level as the while loop above. It is inside (or nested) inside the infinite while loop. Also, be careful with type and punctuation.
- The solution code for this step is provided included for download.

💻 **Slides 18-25 Step #4 Pi Facts**

Students add code to display a random pi fact when BTN-B is pressed. There are three parts to this step.

- First students add a new Boolean variable, and then they add code to the while loop to break the loop when BTN-B is pressed. Again, be careful with indenting! Code snippets are included to guide students.
- Another if statement is added to the code. This will go just above the if statement for BTN-B, and needs to be indented at the same level.
- The solution code for this step is provided included for download.

💻 **Slides 26-31 Step #5 🐍Pi-thon Snake**

Students modify the code so a snake head appears in front of the stream of digits. There are two parts to this step.

- First students add two more variables to the main program. Then they add another while loop for the snake head.
- What to watch for: The new while loop needs to be before the other while loop, just after the variables.
- The display.draw_text() line of code is, once again, a little tricky for typing. You might consider giving it to students if you think they will have trouble with the typing.
- This loop will run until the head disappears off the screen. Then the next loop will take over and continue the stream of digits until they run out or until BTN-B is pressed and a fact appears.

Wrap-up / Optional (5 minutes)

💻 You can wrap-up this project in a variety of ways, depending on your students and your classroom procedures.

- Students can share their projects with other students, especially students not in class. Challenge them to do this!
- Students can fill out a journal entry about their experience or what they learned during the lesson.
- Have a pi day party!!