Lesson: 1.3 Exploring Data Using Python

In this lesson, students will learn the basics of Python programming in the context of data science. This includes how to define and use variables and lists, how to use comparison and logical operators, and the importance of knowing the different data types used in Python.

Objective

Students will be able to:

Use the basics of Python in the context of data science

Define and use variables and lists

Use comparison and logical operators

Understand the importance of the different data types used in Python

Activities

These are all the activities included in the lesson.

Activity

1.3.1 Exploring Data Using Python

1.3.2 Exploring Data Using Python

1.3.3 Live Code - Lists

1.3.4 Using Lists

1.3.5 Practice with Lists

1.3.6 Mini-Project: Lists

1.3.7 Practice with Operators

1.3.8 Finding the Average

1.3.9 Mini-Project: How Can We Use This?

Solution References

Refer to the solution reference for a more detailed look at exercise solutions.

Solution Reference

1.3.2 Exploring Data Using Python

1.3.5 Practice with Lists

1.3.7 Practice with Operators

1.3.8 Finding the Average

Problem Guides

Refer to the problem guides for a more in-depth look at this lesson's problems.

Problem Guide

1.3.5 Practice with Lists

1.3.6 Mini-Project: Lists

1.3.7 Practice with Operators

1.3.8 Finding the Average

1.3.9 Mini-Project: How Can We Use This?

Handouts

Use handouts to supplement your class. Please note that there are handouts for teachers and for students.

Pair Programming Guide

Pair Programming Reflection

Planning Notes

This is an “equalizer” lesson and can be used in a variety of ways depending on the programming strength of your students.

If students already have Python programming experience: This lesson can be used as a review and should take one class period.

If students have experience programming in a different language: This lesson should take 1-2 class periods and the emphasis should be placed on proper syntax.

If students do not have programming experience: This lesson could take 2-3 class periods. There are Python tutorials within each activity for further instruction and practice.

Consider utilizing pair programming throughout this lesson. There is a pair programming guide and reflection document included in the resources section of this lesson plan.

Teaching and Learning Strategies

Lesson Opener:

Have students brainstorm and write down answers to the discussion questions listed below. Students can work individually or in groups/pairs. Have them share their responses. [5 mins]

Activities:

Watch the lesson video and complete the corresponding quiz to check for understanding. [10 mins]

If students already have Python programming experiences, this quiz can be used as a pretest.

Watch the Live Code - Lists video. [5 mins]

Have students take notes on important functions used and what questions they might still have.

Students can also work along with the recording by pausing it and typing out what they see in a sandbox item (or the example) to try it themselves.

Explore the Using Lists example. [5 mins]

Have students answer and discuss the questions in the description before running their code.

Complete the Practice with Lists exercise. [15 mins]

Students can be paired up or put into small groups to complete this activity.

Python tutorials are provided in the description for scaffolding.

Complete the Mini-Project: Lists exercise. [10 min]

Students should copy their data from the past Mini-Project activity into the data.csv file in this activity. While they don’t necessarily need to do this for this specific activity, they will use this procedure continually throughout the course and should practice how to do it.

Complete the Practice with Operators exercise. [15 mins]

Python tutorials are provided in the description for scaffolding.

Complete the Finding the Average exercise. [5-10 mins]

Complete the Mini-Project: How Can We Use This free-response activity. [5 mins]

Lesson Closer:

Have students reflect and discuss their responses to the end of class discussion questions. [5 mins]

Prior Knowledge

No prior knowledge is needed. See Planning Notes for more information on dealing with different levels of programming experience.

Video Slides

Discussion Questions

Beginning of Class:

How is a variable used in math?

A number or value can be substituted into a variable in a math problem.

What programming languages are you familiar with?

Answers will vary.

What is meant by programming?

Answers will vary.

End of Class:

What are some basic guidelines for naming a variable?

Use descriptive names, underscores take the place of spaces, use only lowercase letters, and don’t start with a number.

What are the five data types learned in this lesson? Give an example for each one.

Integer: 9; Float: 9.5; String: “nine”; Char: 9; Boolean: True

Determine whether the following Boolean statements are true or false:

(3 <= 4) and ("a" == "b")

(3 <= 4) or ("a" == "b")

not( (3 <= 4) or ("a" == "b") )

Answers: False; True; False

Modification: Advanced

Remove the examples given in the starter code to further challenge advanced students.

Have students complete the Challenge in the Finding the Average exercise.

Modification: Special Education

Have students reference and complete the tutorials provided in the activities.

Utilize pair programming throughout this lesson. There is a pair programming guide and reflection document included in the resources section of this lesson plan.

Modification: English Language Learners

Have students reference and complete the tutorials provided in the activities.

Utilize pair programming throughout this lesson. There is a pair programming guide and reflection document included in the resources section of this lesson plan.