

# Lesson 3: Variables Practice

## Overview

In this lesson students spend most of their time practicing using the skills and processes they have learned about variables. At the conclusion of the lesson students discuss the main things they realized and still have questions about at the conclusion of this lesson.

## Purpose

This lesson is students primary opportunity to get hands on with variables in code prior to the Make activity in the following lesson. Give students as much class time as you can to work through these. For this lesson it's recommended that you place students in pairs as a support and to encourage discussion about the challenges or concepts they're seeing. In the following lesson students are encouraged to work independently.

## Standards

Full Course Alignment

### CSP Conceptual Framework

- **AAP-1** - To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.
- **AAP-2** - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.
- **AAP-3** - Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing programmers to write programs more quickly and with more confidence.
- **DAT-1** - The way that the computer represents data is different from the way that the data are interpreted and displayed for the user. Programs are used to translate data into a representation that is more easily understood by people.

**CSTA K-12 Computer Science Standards (2017)**

## Objectives

Students will be able to:

- Debug programs that use variables and expressions
- Write programs that use variables and expressions with the support of sample code.

## Preparation

- Review the video in the level progression that covers global vs. local variables. This is a tricky topic.
- Review other programming levels to be better prepared to support students

## Links

**Heads Up!** Please make a copy of any documents you plan to share with students.

For the teachers

- **CSP Debugging Guide**
- **CSP Unit 4 - Variables, Conditionals, and Functions** - Slides
- **EIPM: A Short Introduction** - Resource
- **Guide to EIPM Lessons** - Video
- **Guide to Practice Lessons** - Video

For the students

- **Debugging Global vs Local Variables** - Video

## Agenda

### Lesson Modifications

#### Set Up

#### Warm Up

##### Quick Warm Up

#### Activity (40 minutes)

##### Practice Time

##### Assigning Numbers and Strings

##### Variables and Operators

##### Debugging Scope Issues

##### Putting it Together

#### Wrap Up (5 minutes)

##### Synthesizing Discussion

##### Assessment: Check For Understanding: AP

##### Practice

## Teaching Guide

### Lesson Modifications



**Attention, teachers!** If you are teaching virtually or in a socially-distanced classroom, please read the full lesson plan below, then click **here** to access the modifications.

### Set Up

This is the first official "Practice" lesson in the EIPM model. Review the EIPM model in the **EIPM One Pager**.

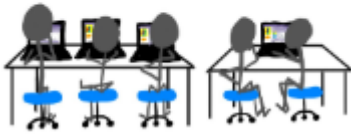
#### Practice Lessons:

**Overview:** Students practice using the new concept through a scaffolded series of programming activities.

- Students work independently or in pairs
- Teacher introduces debugging practices that the beginning of the Activity and circulates the room during the lesson to provide targeted support.

**Goal:** Students gain confidence in writing and debugging programs that use the new concept.

## Practice



Watch the following videos:

- **Guide to EIPM Lessons**
- **Guide to Practice Lessons**

## Warm Up

### Quick Warm Up

#### *Remarks*

Today we're going to have a chance to practice programming with a lot of the concepts and patterns we've explored over the last two lessons. I encourage you to go through these with a partner, but pay close attention to what each other is doing. In our next lesson you're going to have to use a lot of these on an independent project, and these activities are good practice for what you'll find there! Alright, let's get to it!

#### Teaching Tip

**Move Quickly to the Activity:** There's a lot in the main activity of today's lesson. You may optionally wish to do a quick vocabulary review or address any questions that came up in the last lesson. Otherwise, give students more time to get hands on with some code.

## Activity (40 minutes)

### Practice Time

**Group:** It is recommended that students work in pairs for this lesson and a number of the activities feature discussions. Consider using pair programming, having drivers and navigators switch every 3 minutes, not every level.

**Distribute:** Optionally pass out a plastic cup or other manipulative they can place on their computer when they are stuck as a signal that they need support.

#### *Remarks*

Today you're mostly going to practice what we've learned about programming with variables. As always you should be using the debugging process to help you as you work on issues.

 Let's review the debugging process first. Debugging is the process of finding and fixing problems in code. We previously talked about these four steps: Describe, Hunt, Try, and Document.

 Today we're going to also focus on three particular debugging skills

1. Slowing down code with the speed slider
2. Using console.log to get output
3. Using the Watch area to watch your variables change values

 **Do This:** Direct students to Code Studio, Lesson 3 Level 2.

💡 Teaching Tip

**Providing Support:** Circulate around the room through the lesson encouraging students to use the strategies introduced at the beginning of the lesson. Students have a number of supports at their fingertips, so a big part of your role is helping build their independence in using those resources.

## Assigning Numbers and Strings

**Levels 1-2 Assigning Numbers and Strings:** These levels only use the `console.log()` command which prints commands in the debug console. Some key points in these levels.

- Students should practice using the "Watch" area to track how variables change over a program
- Students are asked to discuss what they see with a partner
- These levels heavily reinforce language around assigning and creating variables
- Level 2 is a debugging-heavy level which introduces more rules about syntax.



1-2

### Assigning Numbers and Strings

1

2

## Variables and Operators

**Levels 3-7 Variables and Operators:** These levels have students practice writing more complex expressions and using operators. A few tricky things to look out for

- These levels transition from using only `console.log` to using full apps with user interfaces.
- Level 5 introduces `\n`, the new line character
- Students continue to be encouraged to discuss what they are learning with a neighbor
- Levels 6 and 7 introduce the use of the counter pattern with numbers and strings. Students may need to consult the Help and Tips tab for support.



3-7

### Variables and Operators

3


4


5

6


7

**Regroup:** Bring the class back together to watch the Scope Practice Video.


 **Display:** Scope Practice video. This video is located on the slides and on Level 9. You may opt to have students watch it alone or as a class.

 **Do This:** Discuss debugging scope issues with the next three slides before directing students back to Code Studio.

💡 Teaching Tip

 **Reviewing a Map Level:** Level 9 is a Map Level, which contains a review of variable scope. It is highly recommended that you watch the video in Level 9 prior to the lesson. Slides are available for this lesson if you would like to optionally review as a class.

## Debugging Scope Issues

 **Levels 8-9 Debugging Scope Issues:** These levels identify a common bug that can come up when working with variables. While students should be aware of this bug, they don't quite have the background they'd need to fully understand it. For now support students in following the three main takeaways.

- Create variables once
- Create variables at the top of your program
- Don't create variables inside `onEvent()` or `function()` blocks




8-9

Debugging Scope Issues

8

9

## Putting it Together

 **Level 10 Putting It All Together:** This level asks students to put together many of the concepts they've seen so far. This is an opportunity to have a bit more of a "blank screen moment" while still being able to use some starter code as a guide.




10

Putting it Together

## Wrap Up (5 minutes)

### Synthesizing Discussion

 **Discuss:** *What aspects of working with variables do you feel like clicked today? What do you still feel like you have trouble with?*

**Discussion Goal:** Use this opportunity to address any lingering questions or misconceptions in the room. You can also use this as a source of discussion topics to kick off the following lesson. As you lead the discussion, call out the many resources students have access to help when they're getting stuck.

**Discuss:** Have students share with one another before sharing with the whole class.

### Remarks

Variables can be a little bit tricky, but I saw a lot of good progress today in nailing down this concept. We may have a few lingering questions, but you also have a lot of resources available. Next time you'll have a chance to put all this together by programming an app that starts with "the blank screen"!

## Assessment: Check For Understanding: AP Practice

*Check For Understanding Question(s) and solutions can be found in each lesson on Code Studio. These questions can be used for an exit ticket.*

**Question:** What will be displayed after this code segment is run?

```
myPoints ← 2
```

```
myPoints ← 5
```

```
myPoints ← myPoints + 1
```

```
DISPLAY myPoints
```

☰ 11-12

Assessment: Check For Understanding: AP Practice

11 ✓

12 ✓