

Lesson 10: Model Cards in App Lab

45 minutes

Overview

In this lesson, students practice importing their models into App Lab, this time including models that have numerical data and using model cards to help improve the user experience of filling out their form. They will then learn how to view the model card within App Lab and use this to add more descriptive elements to an app. Next, they focus on improving the user experience by adding informational text to help guide users through completing the form and adding a style to their app to improve the user experience.

Question of the Day: How can I use a Model Card to improve my app?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

► AP - Algorithms & Programming

Agenda

Warm Up (5 minutes)

Journal

Activity (35 minutes)

Model Cards in Apps (15 mins)

If-Statements in Apps

Wrap Up (5 minutes)

Journal

Objectives

Students will be able to:

- Use a model card to update the user interface of an app
- Use if-statements to change the behavior of an app

Preparation

- Review the Code Studio levels before the lesson
- Check the "**Teacher's Lounge**" forum for verified teachers to find additional strategies or resources shared by fellow teachers

Links

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

For the teachers

- **Model Cards in App Lab** - Slides
▼ Make a Copy
- **Printable Model Cards** - Resource
▼ Make a Copy

For the students

- **If-Statements in App Lab** - Video

Introduced Code

- `if (condition) { statement }`

Teaching Guide

Warm Up (5 minutes)

Journal

 **Display:** Show a screenshot of an app that has imported a model into App Lab with no changes to design mode.

 **Prompt:** *What changes could you make to this app to make it easier for people to use? Write a list of changes to the design or to the app flow that you would recommend for this app.*

Have students journal individually first, then share in small groups before bringing the full class together.

Discussion Goal: Students may comment on style elements, such as using colors to liven up the page. Students may comment on the descriptions to the user, like how the generated prompts aren't very descriptive. Students may comment on the flow of the app, such as suggesting that it be broken up into multiple screens or adding a "welcome" screen to the beginning. All of these are valuable brainstorm ideas that will carry into today's lesson.

 **Code Studio:** Have students log into Code Studio. The first level has students explore an enhanced version of this same app, with many of the improvements implemented. As the class explores, highlight any improvements that matched their suggestions. Emphasize that this app does one thing we haven't seen before: it changes screens based on the animal that is predicted.



Example App: Zoo Classification

Remarks

As we can see, there are a lot of ways to improve our app - and you all predicted many of the improvements we see here. Some improvements aren't only with the design - even adding additional information can make an app more helpful to its users. Today we'll learn more about how we can make our apps easier to use with the help of a Model Card.

 **Question of the Day:** How can I use a Model Card to improve my app?

Activity (35 minutes)

Model Cards in Apps (15 mins)

 **Code Studio:** Have students continue to the next level in Code Studio. In this first sequence of levels, students will see how to use a Model Card to update their app.



Pet Recommender

Teaching Tip

Purposefully Mysterious: This app is deliberately mysterious and requires a lot of guessing to understand. This is purposeful, since we will work to improve this app in the next few levels. If students ask questions or express uncertainty about how the app is working: validate their instincts, but avoid revealing too much about the app -

instead, encourage them to keep these thoughts in mind as the lesson continues.

 **Discuss:** *Some of the features use dropdowns and others use text boxes. What do you think is the difference between these two types of features?*

Discussion Goal: Guide students to notice that the text boxes require numbers, and the dropdown boxes use words. Once this point is made, transition to the next slide with the definitions for Categorical and Numerical data.

Vocabulary: Show students the vocabulary for the lesson, which includes the new term for Numerical data.

- **Categorical Data:** data that can be separated into groups
- **Numerical Data:** data that can be counted or measured

Teaching Tip

Numbers in Dropdowns: Students may notice that the NO feature has a dropdown that contains the numbers 1-4, and they may ask why this is represented as a dropdown instead of a text input. This is a valid question that will get cleared up in the next level - for now, confirm with students that this feels confusing and prompt them to consider: what extra information do you wish you had to help understand what's happening here?

 **Discuss:** *This app isn't very easy to understand. What do you think the features represent? How many different pets can it recommend?*

Let students share their guesses, but be careful not to validate as "right" or "wrong". Instead, let the class come to their own conclusions before continuing.

Remarks

If we're going to create this app, we'll need to add more information to make it easier for the user to understand how this app is making decisions. Luckily, we have a Model Card that's included with our app and has this information. Let's look at the model card and see what extra information it can tell us about our features and label.

 **Code Studio:** Have students continue to the next level in Code Studio, which displays the model card. Have students read the labels and features to better understand how the app is working. If students guessed any of these in the earlier discussion, circle back and validate those answers from before.

 3

Pet Recommender

Teaching Tip

Printing Model Cards: You may decide to print the model card as well so students can read it on paper. If possible, the model card is designed to be printed two-sided so it appears as a single sheet of paper that you can flip back and forth.

 **Discuss:** *How can you tell the difference between categorical features and numerical features in the model card?*

Discussion Goal: Students should notice that numerical features have a min and a max on their card, while categorical features have a bulleted list of the possible values.

Remarks

Let's use the information from this model card to improve our app, especially the questions for each feature. We'll also start improving the look and design of our app to be more user friendly in the next few levels.

 **Code Studio:** Have students continue to the next level, where they can still access the model card for this app and use it to update the user interface.

 4

Pet Recommender

Circulate: Monitor students as they update their app, ensuring that they can access the Model Card to determine the correct labels

 Teaching Tip

Updating Labels Only: Students should only be updating the design of their app - they don't need to update the code, which still uses the shorter one-or-two letter abbreviations from the original app. This is because it's common to use much shorter abbreviations with data, and use longer explanations in places like the model card. If a student updates the `addPair()` section of their code, it may cause errors in their app.

If-Statements in Apps (20 minutes)

Remarks

You may have noticed that in this app, the prediction appears on the home screen. But in the app we tested, we went to a whole new screen based on the result! In order to have our app change screens, we'll need to check the result of our prediction. This means our code will need to use a new type of block called an **if-statement**.

 **Video:** Show the Conditionals video in the slide deck.

 Teaching Tip

Videos for Students: Videos are intended to be watched and discussed as a class, and so they are not provided as individual levels in Code Studio. If a student needs to re-watch a video, they can be found in the Help and Tips section of levels or by visiting the Student Resources page of each lesson.

To encourage active engagement and reflection, use one or more of the strategies discussed in the [Guide to Curriculum Videos](#).

 **Code Studio:** Have students go to the next level in Code Studio, where they will add if-statements to their code to change the screen. They will continue to work through the remaining levels as well, adding new elements to their app along the way.

 5-7

Adding New Screens

5

6

7

 Teaching Tip

Normalizing Mistakes and Supporting Debugging: As programming levels become more complex, students may find themselves with bugs in their code that they need to untangle. If this happens frequently, this can be a demoralizing experience for students and can affect their self-perception of how capable they are in class.

To counter this, we recommend normalizing bugs and mistakes as something that happens to everyone - it's just part of the process. You can show students our [Debugging Video](#), which includes several students normalizing mistakes and discussing debugging strategies that students can use. Additionally, consider displaying the [Student Guide to Debugging](#) for students to reference throughout the unit and having [Bug Report Quarter-Sheets](#) available for students to use.

Practice Levels: Have students choose one or two practice levels to complete in order to deepen their understanding of model cards and if-statements. Students don't need to complete all practice levels - once they feel comfortable, they can continue to the next level to see how well they've mastered the concepts in this lesson.



8

Practice

Teaching Tip

Student Choice and Practice Levels: Having students choose their own practice levels are one way to differentiate instruction with your students. To learn more about how these levels support differentiation, as well as other strategies, you can review the [CSD Guide to Differentiation](#)



9

Assessment

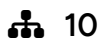
Assessment Opportunity

Formative Assessment: This level can be used as a formative assessment. A rubric is provided in the level, and written feedback can be given to students. [Click here to learn more about giving feedback to students.](#)

Choice Levels: Once students have created their Pet Recommender app, they can choose from several example activities to continue improving their app. Each level uses the same Pet Recommender app they've already built as a template to add to. Students can choose to complete as many of these levels as they would like - each one lets them continue to customize and improve their app.

Teaching Tip

Deja Vu? Students may notice that the choice levels look familiar in this lesson - they're the same levels that students could choose from in Lesson 7. This is intentional, since it gives students a chance to learn a different piece of code than they learned previously. For students who have explored each option already, encourage them to apply their knowledge to this new app.



10

Challenge

Wrap Up (5 minutes)

Journal

 **Prompt:** *What is one example of how a model card can be useful when making an app?*

Discussion Goal: Look for responses that highlight how a model card can help communicate the purpose of an app to a user. Some examples from today may include:

- Using label or feature descriptions to help clarify choices in an app
- Using the “Intended Uses” section to help create a welcome screen for an app
- Including the possible values for categorical features in an app



This work is available under a **Creative Commons License (CC BY-NC-SA 4.0)**.

If you are interested in licensing Code.org materials for commercial purposes **contact us**.