

# Introduction to Programming with Xojo

# TEACHER'S GUIDE

# RHINE, LEFEBVRE





# Introduction





# Introduction

Welcome to the Teacher Guide for [Introduction to Programming with Xojo](#)! This guide has been developed to help you teach your students how to program, even if you have limited experience yourself.

For each chapter in *Introduction to Programming with Xojo*, you'll find an accompanying chapter in this guide. The chapters in this guide are broken into four sections:

## CONCEPTS AND VOCABULARY

---

You won't necessarily find "dictionary definitions" here, but this section will introduce you to the basic ideas of the chapter and the terms you need to know. Wherever possible, the terms are explained in basic English.

## LINKS AND REFERENCES

---

When you're ready to move past terms and definitions, you'll find some links in this section that can provide you with more information on the chapter's topics.

## REVIEW QUESTIONS

---

These are not questions to which there is necessarily a right or wrong answer. These questions are intended to encourage discussion among your students.

## PROGRAMMING CHALLENGE

---

If you want to challenge your students beyond what's in the Xojo textbook, these programming challenges are designed to expand upon the exercises they've worked on in each chapter. Solutions have not been provided, because for any given programming problem, there are almost countless ways to solve it. More important than the code being "right" is the program's behavior and output. To encourage exploration, challenges may involve using controls or techniques that have not yet been introduced in the Xojo textbook.

You are encouraged to read each chapter of the Xojo textbook yourself, going through the exercises and referring to this guide as you go.





# Chapter 1: Hello, World!

# 1.1 Concepts & Vocabulary

## 1) IDE

IDE stands for Integrated Development Environment. An IDE is a computer program used to enter and edit code, and often includes such features as code completion and debugging (the Xojo IDE offers both of these). Simply put, it's the program that you use to create programs.

## 2) Bug

A bug is a defect in a computer program. When a program does not behave as expected, that's a bug. Sometimes a bug is due to a problem with the code that has been entered. Often, it's a problem with an assumption that the programmer has made.

## 3) Debug

Strictly speaking, to debug means to remove bugs or errors from a program. In more practical terms, it's also a term used to describe the process of analyzing a running program to check for defects and verify its behavior.

## 4) Code

Simply put, code is the “words” of the programming language that you enter into an IDE to build programs. As you'll see later, code can be made up of keyword, function names, variables, and more. But for now, think of code as the words you type into the IDE.

## 5) Language

If code is the words you type in, the language is the set of words you can choose from. Xojo is a programming language; so are Swift, C#, PHP, Objective-C, Java, and Python. While many languages share similarities in syntax and even keywords, each language is unique in its own way.

## 6) Run and Build

Whenever you launch an app on your computer, tablet, or smartphone, you are running a program. In order to run a program, an IDE (or sometimes a compiler, which is built into many IDEs) needs to turn your code into something the computer can understand and use as instructions on how to act. When the result is an app that you can tap or double click to launch, that process is called building. Xojo and some other IDEs can also take your code and allow it to run without creating an app that you can use later. With Xojo, you can also build a web application that can be run in your computer's web browser.

## 7) Exception

An exception occurs when something in your program happens that was never supposed to happen. This is not necessarily the same as a bug. A bug is an error in your code; it's something you've done wrong when creating your app. An exception happens when the final program is running, and while it often points to an error in your code, it could also be that you wrote perfectly good code with the wrong assumptions. Xojo's exception handling can tell you what kind of exception has occurred. Remember that the word “exception” means that something that's not supposed to happen has happened, such as running out of memory to hold a picture or trying to access a file that doesn't exist. The key to dealing with exceptions is to prepare your code to handle such situations.

# 1.2 Links & References



1. **Programming:**  
[http://en.wikipedia.org/wiki/Outline\\_of\\_software\\_development](http://en.wikipedia.org/wiki/Outline_of_software_development)
2. **Programming Q&A:**  
<http://programmers.stackexchange.com/>
3. **Debugging:**  
<http://en.wikipedia.org/wiki/Debugging>
4. **Exception:**  
<http://docs.xojo.com/index.php/Exception>

## 1.3 Review Questions

1. What is the difference between running and building an application?
2. One of the options you can set for your application's StageCode is beta. How important is it to beta test your applications? What do you think are some good ways to test your applications?
3. An exception is something that's not supposed to happen, but sometimes does. A good example is your car failing to start when you turn the ignition key. What are some things you could check for this particular exception?
4. What are some other real world "exceptional" situations you can think of? How would you prepare to deal with them?

## 1.4 Programming Challenge

For an easy challenge, change the message that appears when you click the button. For a more advanced challenge, use a Bevel Button instead of a Button to trigger the message.

# Chapter 2: Introduce Yourself



## 2.1 Concepts & Vocabulary

### 1) Variable

A variable is a named value in your code. The value can change, but the name remains the same. You might have a variable called `PageCount` that keeps track of how many pages your program has printed. Each time your program prints a new page, `PageCount`'s value should increase. Variables can be of different types. Most of these types are explained in *Introduction to Programming with Xojo*, while a few are detailed below.

### 2) Assign

The process of telling your program what a variable's value should be is called assignment. You're assigning a value whenever you type something like this:

```
x = 42
```

Assignment can also be self-referential, such as:

```
x = x + 1
```

### 3) String

One of the most common types of variables is the string, which is simply a piece of text. Any time your program tracks a name, a web address, or a paragraph of information, it's stored as a string. If the word "string" doesn't mean much to you, think of various characters (numbers, letters, punctuation, etc.) that we "string" together for display.

### 4) Integer

An integer, on the other hand, is a numeric value (an integer is always a whole number; Xojo also features doubles, which can store decimal information). It's important to distinguish between strings and integers: you could store "1" as a string or as an integer. The basic distinction is that you should store variables as integers (or doubles) if you're going to be doing any sort of mathematical manipulation, even something as simple as addition.

### 5) Boolean

A boolean is a variable type that can store one of two values: true or false. They are most helpful when you need to keep track of something's status or condition, such as whether a document has been saved or your program's preferences have been set. A boolean always defaults to false and doesn't change until your code tells it to.

### 6) Comment

Comments are a way to document what your code is doing. While they are part of your code, they are not part of your final compiled app. They are there as a reference for anyone who may have to read, debug, or modify this code at some later date.

## 2.2 Links & References

### 1. Variable:

[http://en.wikipedia.org/wiki/Variable\\_\(computer\\_science\)](http://en.wikipedia.org/wiki/Variable_(computer_science))



2. Boolean:  
[http://en.wikipedia.org/wiki/Boolean\\_data\\_type](http://en.wikipedia.org/wiki/Boolean_data_type)
3. String:  
<http://www.techterms.com/definition/string>
4. Integer:  
<http://www.mathsisfun.com/whole-numbers.html>

## 2.3 Review Questions

1. What are some reasons it's important to add comments to your code?
2. How would you explain the difference between "12" (the string) and 12 (the integer) to someone with no programming experience?
3. Why is it important to be consistent with your variable naming strategies?

## 2.4 Programming Challenge

Instead of calculating the user's age in years, calculate it in days and/or months.



# Chapter 3: Where Do We Go Now?





# 3.1 Concepts & Vocabulary

## 1) Flow Control and Boolean Logic

Flow control and boolean logic go hand in hand. Boolean logic is the technical term for a “if this, then that” scenario. It involves checking a variable’s value and then responding appropriately. Flow control is the name given to that entire process (think of a flowchart with various branches, and you’ll have the right idea). Check the Xojo textbook for examples of what if statements look like in code.

## 2) Operator

An operator is something that acts on a variable. Typical examples include mathematical operators like +, -, and \*. Xojo features other operators as well. You will learn more about these in future chapters.

## 3) Case

Sometimes “if this, then that” situations can get very complex in your code. If you have a situation where a variable could have more than two values and each value requires a different action, that’s a good place to use a case statement. A case statement is very similar to an if statement, but offers any number of choices. Again, the Xojo textbook offers examples.

## 4) Loop

A loop is a way of cycling through a list of items. In the code example in this chapter, the program loops through all of a user’s fonts on a computer. For each run through the loop, your program can perform any number of actions, including breaking out of the loop if necessary.

# 3.2 Links & References

1. Flow Control:  
[http://www.webopedia.com/TERM/F/flow\\_control.html](http://www.webopedia.com/TERM/F/flow_control.html)
2. Boolean Logic:  
<http://www.howstuffworks.com/boolean.htm>
3. Operator:  
<http://www.bfoit.org/itp/Operators.html>
4. Loop:  
[http://en.wikipedia.org/wiki/For\\_loop](http://en.wikipedia.org/wiki/For_loop)

# 3.3 Review Questions

1. How is a Select statement more efficient than an If statement?
2. What are the differences between a For...Next loop and a Do...Loop? When would you use one approach over the other?

# 3.4 Programming Challenge

Add style buttons to the Font Previewer app that change the example to bold, italic, and/or underline. For a more advanced challenge, display the font names in the List Box in their own fonts.



# Chapter 4: Getting Things Done



# 4.1 Concepts & Vocabulary

## 1) Method/Function

A method or function, in the simplest terms possible, is a set of instructions for your program. If your program does a specific task more than once, it's a good idea to turn that code into a method rather than type the same code into multiple places in your program. Both methods and functions are sets of code. The key difference is that a function returns a value while a method doesn't.

## 2) Parameter

Some methods and functions can behave differently based on values that you give them. These values are called parameters. Not every method or function will take parameters, but many will.

## 3) Return

Some methods and functions will perform a task and exit without doing anything else. Others are designed to provide a value back to the code that calls the function. These values are called return values.

## 4) Scope

Variables defined within a function can't be used outside that function. That's because of something called scope, which limits which parts of your code can access other parts of your code. For example, you may have a method that needs to modify a string, but you don't want any other code to have access to it. That string's scope would be contained within the method.

## 5) Default

You can specify some of your function's parameters to have specific values if they're not otherwise provided. These are called default values.

# 4.2 Links & References

## 1. Scope:

[http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Scope\\_\(programming\).html](http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Scope_(programming).html)

## 2. Scope:

[http://en.wikipedia.org/wiki/Scope\\_\(computer\\_science\)](http://en.wikipedia.org/wiki/Scope_(computer_science))

## 3. Functions:

<http://www.webopedia.com/TERM/F/function.html>

## 4. Functions:

<http://www.cs.utah.edu/~germain/PPS/Topics/functions.html>

## 5. Return:

[http://en.wikipedia.org/wiki/Return\\_statement](http://en.wikipedia.org/wiki/Return_statement)



## 4.3 Review Questions

1. When should you turn a piece of code into a method or function?
2. Default parameters can be very convenient, but can you think of a situation in which they can introduce bugs into your application?
3. In what cases would it be smart to give a method a scope of `Private`? Why?

## 4.4 Programming Challenge

In the Font Previewer app, add the ability for the end user to select the font size in the preview. For a more advanced challenge, “double” the interface of the app, so that users can select two fonts to see how they look together on screen.

# Chapter 5: Making A List



2ECCCS  
1AB Chipset  
MP2 5002 AIN  
MP2 3100 Knapel  
MP2 3100 Gold  
MP2 2000 Knapel  
MP2 2000 AIN  
MP2 3100 Knapel

## KEYBOARDS

Logitech Access Keyboard  
Cherry G83 MF102 Keyboard  
Cherry G83 Keyboard  
Cherry G81-3000 Keyboard  
Cherry G81-3000 Keyboard  
Logitech Cordless Desktop  
Logitech Cordless Desktop

# 5.1 Concepts & Vocabulary

## 1) Array

At its most basic level, an array is a list of items. These items can be something simple like strings or integers, or something more complicated, such as custom data types that you define. You can even have an array of arrays. Each item in an array has something called an index, which indicates its position in the array (these positions are zero-based, so the first item is at index 0, the second item is at index 1, etc.).

## 2) Adding Items

Adding an item to an array is similar to adding something to the end of a list. You can also add an item into an array at a specific position by telling Xojo the index where you want the new item to be.

## 3) Sort

By default, an array's items remain in the order in which they were created. You can sort them, however. The most common sorting methods are alphabetically or numerically.

## 4) Shuffle

Just like with a deck of cards, shuffling puts an array's items into random order.

## 5) Delimiter

A delimiter is a character that's used to "mark off" sections of data. For example, you may have worked with a .csv (comma separated values) file in the past. If so, you may have noticed that each value was separated, or delimited, by a comma.

# 5.2 Links & References

## 1. Arrays:

<http://www.bfoit.org/itp/Arrays.html>

## 2. Delimiters:

<http://en.wikipedia.org/wiki/Delimiter>

## 3. Sorting:

[http://en.wikipedia.org/wiki/Sorting\\_algorithm](http://en.wikipedia.org/wiki/Sorting_algorithm)

## 4. Shuffling:

<http://www.codinghorror.com/blog/2007/12/shuffling.html>

# 5.3 Review Questions

1. Aside from the card game example, what are some situations in which you may want to shuffle an array?
2. Explain your class schedule in terms of an array.

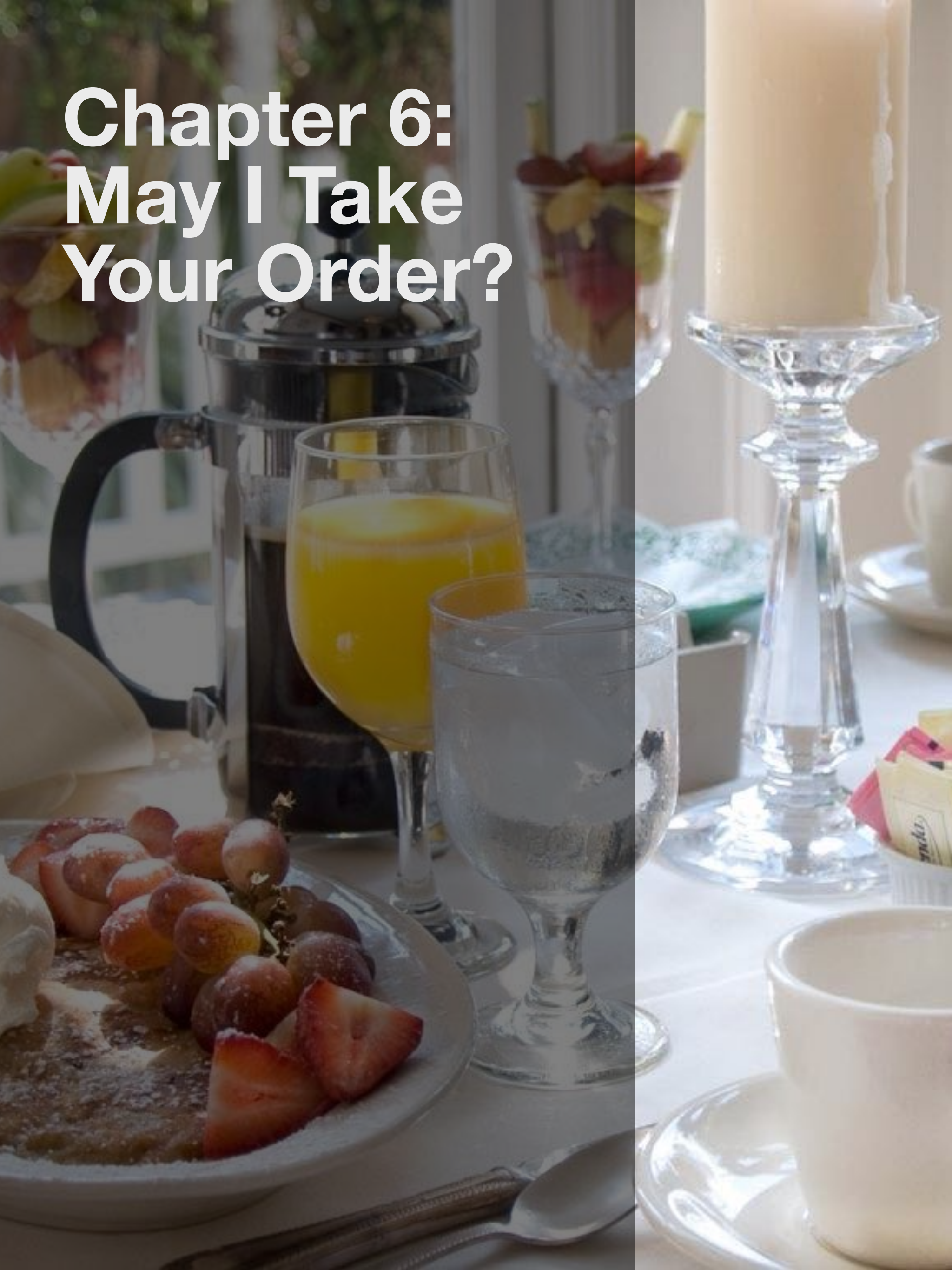


3. The to-do list example is a web application. What makes the web an especially good platform for a to-do app?
4. What are some other apps that might lend themselves to web versions?

## 5.4 Programming Challenge

Add the ability to assign each task a priority and enable the user to sort by it.

# Chapter 6: May I Take Your Order?



# 6.1 Concepts & Vocabulary

## 1) Event

When your program does certain tasks, it necessarily has to do some others. For example, if you need to show a button on the screen, your program has to create the button and it has to draw the button. Your program also has to be able to respond to the user. When the user clicks on a button, you may have code you want to run in response to that. Each action that the user takes is called an Event. You can add code to different events in your program to change the way it behaves and interacts with users.

## 2) Control

A control is sometimes called a widget. It's a user interface object that displays information or reacts to the user, such as a button, a label, a listbox, or a scrollbar.

## 3) Label

A label is a (usually) non-interactive control that indicates another control's purpose or name. It is often used as a caption.

## 4) Button

A button is an interactive control that performs an action (that you specify) when the user clicks on it. Buttons can take on various sizes, shapes, and forms.

## 5) Window

A window is a physical grouping of controls on your screen. Most often we think of each window as a screen, sheet, or view.

# 6.2 Links & References

1. Events:  
[http://en.wikipedia.org/wiki/Event-driven\\_programming](http://en.wikipedia.org/wiki/Event-driven_programming)
2. Controls:  
[http://en.wikipedia.org/wiki/GUI\\_widget](http://en.wikipedia.org/wiki/GUI_widget)
3. Controls:  
<http://www.apl.jhu.edu/~hall/CWP-Chapter13/>

# 6.3 Review Questions

1. In general, it's better to use system-native controls (like Buttons) rather than design your own. What are some reasons this might be so? What are some situations in which you may want to create your own buttons from scratch?
2. What are some advantages to using pickers, such as a Slider or Radio Group, instead of allowing the end user to enter text?



# 6.4 Programming Challenge

In the food ordering app, add a radio button for pepperoni, but make sure it's only enabled when pizza is selected as the main dish.



# Chapter 7: Just Browsing





# 7.1 Concepts & Vocabulary

## 1) Table

A table is visually similar to a spreadsheet, featuring rows and columns of data. In Xojo, a table is called a listbox. A listbox can have headings, handle sorting, and even allow inline editing. Whenever your app needs to display tabular data of some sort, the listbox will likely be the control that you need.

## 2) Control Group

A control group is exactly what its name implies: a logical grouping of controls. Controls in a control group can be acted on together.

## 3) Progress Bar

A progress bar is a control that indicates how much of something has happened or has been processed. Typical examples are a file download or a movie player. The progress bar is used when the end value is known. When it's unknown, the progress wheel should be used instead (note here that you can also use a progress bar with an indeterminate value to get a “barber pole” that will tell the user that the length of the task is unknown).

## 4) Tab Panel

Tabs are another way to group controls together. You've probably encountered tabs in your web browser. Using the Tab Panel control, you can organize your interface into more logical groupings since each tab has its own set of controls.

# 7.2 Links & References

## 1. Progress Bar:

[http://en.wikipedia.org/wiki/Progress\\_bar](http://en.wikipedia.org/wiki/Progress_bar)

## 2. Tabs:

[http://en.wikipedia.org/wiki/Tab\\_\(GUI\)](http://en.wikipedia.org/wiki/Tab_(GUI))

# 7.3 Review Questions

1. What are some situations in which it would be good to use a Page Panel or Tab Panel? When would you use one versus the other?
2. A web browser is obviously much more complicated than the example project in this chapter. What are some fundamental browser features that are missing? What some non-standard features you would add to a web browser?
3. Explain how you could use Page Panels to create a “wizard” interface.

## 7.4 Programming Challenge

Add a list of preselected bookmarks that your end user can navigate to with one click. For a more advanced challenge, give the end user the ability to add his or her own bookmarks (don't worry about saving them when the app quits).



# Chapter 8: Do It Yourself





# 8.1 Concepts & Vocabulary

## 1) Class

A class is a custom type of variable that you design yourself. As noted in the Xojo textbook, it usually represents a real-world object (like a person or a car) or an abstract idea (like a hotel reservation or a class enrollment). A class can include its own data, properties, and methods. A person class may includes properties like name and age, and methods like jump, run, and speak.

## 2) Object

Think of a class as the definition of your custom data, and an object as an instance of that data in your code. The class is the idea and the object is what your code acts upon. For example, you specifically would be an object of the “person” class, with your own properties.

## 3) Property

A class can have properties that you define. These are attributes of the object or idea that it represents. For example, a class that represents a person might have properties like LastName as string, FirstName as string, and AgeInYear as integer.

## 4) Variant

A variant is an intrinsic data type in Xojo that can hold any kind of data, whether string, integer, boolean, or a custom object that you designed. Variants should be used with great caution, as they make it very easy to introduce subtle bugs into your code. In general a variant should only be used when you need to store different kinds of data (strings, integers, or even custom objects) in the same place, such as in a CellTag of a ListBox.

## 5) Module

In Xojo, a module is similar to a class, but only one instance of it can exist in your app at one time. It’s easiest to think of a module as a single use class. Unlike a class, a module always exists, which makes it useful for tasks such as storing and loading preferences, handling application-wide data storage, and other global functions.

# 8.2 Links & References

## 1. Classes:

[http://en.wikipedia.org/wiki/Class\\_\(computer\\_programming\)](http://en.wikipedia.org/wiki/Class_(computer_programming))

## 2. Classes:

<http://www.bfoit.org/itp/JavaClass.html>

## 3. Properties:

[http://en.wikipedia.org/wiki/Property\\_\(programming\)](http://en.wikipedia.org/wiki/Property_(programming))

## 4. Variants:

[http://en.wikipedia.org/wiki/Variant\\_type](http://en.wikipedia.org/wiki/Variant_type)

## 8.3 Review Questions

1. This chapter used the example of a class being like a blueprint and an object being like a building. What are some other analogies to help explain the difference?
2. Variants should be used with caution. What would be some good uses of variants?
3. When you should build your own classes instead of using Xojo's built-in data types?

## 8.4 Programming Challenge

Add an “EmailAddress” property to the Student class. Provide a way for a teacher to email a student (hint: check out the ShowURL method).



# Chapter 9: In And Out





# 9.1 Concepts & Vocabulary

## 1) File

A file is a physical piece of data that resides on a disk. A file can be anything from a Microsoft Word document to a song in MP3 format to a driver that controls your printer.

## 2) Folder

Sometimes referred to as a directory, a folder is a logical container that stores files and other folders. In Xojo, both files and folders are represented by the FolderItem class.

## 3) Stream

Streaming is a method of writing data to a file or reading data from a file. Information is “streamed” from your program to a disk (saving a file) or from a disk to your program (opening a file).

## 4) Dialog

A dialog is a specialized window that either presents the user with specific information (such as an error message or other warning) or requests information from the user (such as which file to open).

## 5) Text Styles

You are probably already familiar with text styles from using a word processor. Text styles include embellishments like bold, italic, and underline. They also include information like the selected font and font size.

# 9.2 Links & References

## 1. Files and Folders:

[http://en.wikipedia.org/wiki/File\\_system](http://en.wikipedia.org/wiki/File_system)

## 2. Files:

[http://en.wikipedia.org/wiki/Computer\\_file](http://en.wikipedia.org/wiki/Computer_file)

## 3. Folders:

[http://en.wikipedia.org/wiki/Directory\\_\(file\\_systems\)](http://en.wikipedia.org/wiki/Directory_(file_systems))

## 4. Text Styles:

[http://en.wikipedia.org/wiki/Formatted\\_text](http://en.wikipedia.org/wiki/Formatted_text)

# 9.3 Review Questions

1. How would you explain the difference between a file and a folder to someone with little computer experience?
2. When would you have your application use text files as opposed to binary files?

# 9.4 Programming Challenge

You’ve learned about styled text in this chapter. Create a new app whose sole purpose is to take styled text and return a plain text version to the end user (the same words but stripped of font and style data).





# Chapter 10: Picture This (Then Print It Out)

# 10.1 Concepts & Vocabulary

## 1) Image

In Xojo, there are differences among images, pictures, and graphics. An image is what you might normally think of as a picture. It's a capture of a scene or item. If you have a photograph of a dog, that would be an image.

## 2) Picture

A picture, in Xojo terminology, is an object that represents an image. It has properties that reflect the details of the image, such as height and width.

## 3) Graphics

Graphics is the Xojo object that does most of the heavy lifting when it comes to working with images. The graphics class contains methods for drawing, printing, and nearly anything related to displaying graphics.

## 4) Canvas

The canvas is a special control in Xojo that provides you with many options for displaying pictures or text. It has a graphics object that makes it extremely flexible.

## 5) Print

Printing is the act of sending information, whether text or pictures, from your program to a piece of paper.

# 10.2 Links & References

## 1. Canvas:

<https://docs.xojo.com/Canvas>

## 2. Printing:

<https://docs.xojo.com/UserGuide:Printing>

# 10.3 Review Questions

1. What are the differences between a picture, an image, and graphics? Why is there a difference at all?
2. Why is it unwise to assume that you'll know the dimensions of the Graphics object being sent to a printer?

# 10.4 Programming Challenge

Create an application that can be used to open, resize, crop, and scale images according to the end user's specifications.



# Chapter 11: Connections





# 11.1 Concepts & Vocabulary

## 1) Network

A network is a connected group of computers or other devices. Different types of networks have different connections and speeds, but typically each device can communicate with each other device.

## 2) Protocol

A protocol is a detailed description of how two or more programs or devices can communicate over a network. You may have noticed that many websites start with “http” - that’s an indication that the site uses the hypertext transmission protocol. There are protocols for printing, sending emails, and any other task you can perform across a network.

## 3) Port

With all of this data flowing into a computer, how does the computer know which data is going to which application? Well, every protocol communicates on a port, which is represented by a number. In and of themselves, ports don’t really mean much; they’re simply an agreed upon way for devices to communicate. The web usually uses port 80, while sending email uses port 25. On most computer systems, port numbers below 1024 are reserved for system use. Only one application can communicate on any particular port at one time.

## 4) Address

Each device on a network has an address. On most modern networks, this is an IP address, which is represented by four numbers from zero to 255, separated by dots. For the most part, you don’t need to understand how these addresses work; just remember that each device needs a unique address.

## 5) Socket

In Xojo, network tasks are accomplished using the socket class. The socket has functions and properties that take a lot of the “grunt work” out of doing network communications.

# 11.2 Links & References

## 1. Networks:

[http://www.wildpackets.com/resources/compendium/glossary\\_of\\_networking\\_terms](http://www.wildpackets.com/resources/compendium/glossary_of_networking_terms)

## 2. Protocols:

[http://en.wikipedia.org/wiki/Communications\\_protocol](http://en.wikipedia.org/wiki/Communications_protocol)

## 3. Protocols:

[http://en.wikipedia.org/wiki/Lists\\_of\\_network\\_protocols](http://en.wikipedia.org/wiki/Lists_of_network_protocols)

## 4. Ports:

[http://en.wikipedia.org/wiki/List\\_of\\_TCP\\_and\\_UDP\\_port\\_numbers](http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers)

## 11.3 Review Questions

1. Aside from retrieving the contents of a web page, what are some ways that you could use `URLConnection`?
2. Do you see more value in using (or extending) an existing protocol or in developing your own? Why?

## 11.4 Programming Challenge

In your email app, add the ability to send email to multiple recipients. For a more advanced challenge, add the ability to send file attachments.



# Chapter 12: Rows And Columns

STANDARD FORM / "STRING" SOUND



# 12.1 Concepts & Vocabulary

## 1) Database

A database is an organized collection of data. Information in a database is usually organized into tables, which are explained below.

## 2) Table

The table is the basic building block of a database. A table, like a class in Xojo, represents a real-world object (like a person or a car) or an abstract idea (like a hotel reservation or a class enrollment).

## 3) Column

Every table in a database has at least one column. A column is a piece of information related to the table. For example, a table that represents cars might have columns that store information like manufacturer, model number, and dimensions.

## 4) Row

A row is an instance of the “thing” that your database table represents. For example, if your table is called People, each row would represent one person.

## 5) Database Server

Some databases are housed in files that reside on your own computer, while others require database servers

## 6) SQL

SQL stands for Structured Query Language, and it’s the language that’s used to extract data from your database or insert data into it.

## 7) CRUD

CRUD is an acronym that stands for Create, Read, Update, and Delete, which are the four basic functions of any database or database application. If your app uses a database, most of what it does will boil down to one of these four things.

## 8) ISO Date

Most databases store dates in a format called ISO 8601, often referred to as an ISO Date. The format is YYYY-MM-DD: four digit year, two digit month, and two digit date. Note that both the month and the date should include leading zeroes for any value below 10.

# 12.2 Links & References

## 1. Learning SQL:

<http://www.sqlcourse.com>

## 2. Description of SQL:

<http://en.wikipedia.org/wiki/SQL>

## 3. CRUD (Create, Read, Update, Delete):

[http://en.wikipedia.org/wiki/Create,\\_read,\\_update\\_and\\_delete](http://en.wikipedia.org/wiki/Create,_read,_update_and_delete)

4. ISO Dates:  
[http://en.wikipedia.org/wiki/ISO\\_8601](http://en.wikipedia.org/wiki/ISO_8601)
5. Database Servers:  
[http://www.webopedia.com/TERM/D/database\\_server.html](http://www.webopedia.com/TERM/D/database_server.html)

## 12.3 Review Questions

1. List some real world examples of databases you've seen in action, whether computerized or not.
2. Why do you think dates are usually stored in ISO format? What are some advantages and disadvantages of this format?

## 12.4 Programming Challenge

In your Address Book app, add the ability to edit existing contacts (hint: check out the UPDATE command in SQL).



# Chapter 13: All In The Family - Subclasses



# 13.1 Concepts & Vocabulary

## 1) Subclass

A subclass is a “child” of an existing class. As its child, it takes on all of its parent’s attributes and methods, but can also include additional attributes and methods.

## 2) Superclass

A subclass’s parent is also called its superclass.

## 3) Inheritance

When a subclass takes on its superclass’s attributes and methods, that is known as inheritance.

## 4) Cursor

In desktop operating systems, the cursor is the object on the screen that moves with the mouse or other pointing devices, showing the user’s position on the screen.

# 13.2 Links & References

## 1. Inheritance:

[http://en.wikipedia.org/wiki/Inheritance\\_\(object-oriented\\_programming\)](http://en.wikipedia.org/wiki/Inheritance_(object-oriented_programming))

## 2. Cursors:

[http://en.wikipedia.org/wiki/Cursor\\_\(computers\)](http://en.wikipedia.org/wiki/Cursor_(computers))

# 13.3 Review Questions

1. How would you explain subclasses to someone brand new to programming?
2. In this chapter, you learned about cursors. Why is it important to have different cursors in your app?

# 13.4 Programming Challenge

Add other types of people to your Subclasses app: teachers’ aides, principals, and secretaries. Think about the attributes they’ll need that might be different from the existing student and teacher classes.



# Chapter 14: Spit And Polish





# 14.1 Concepts & Vocabulary

## 1) Thread

Threads are a very complex subject, but for now, think of a thread as a worker in your computer. Each worker can work on a job or part of a job, and by using several workers at once, your app can be faster or more responsive.

## 2) Timer

The timer is a Xojo control that triggers an event when you specify. The events can be single or recurring, and timers can be turned on and off when you need.

## 3) Random

The random class in Xojo is used to generate a number without knowing what that number will be (although you can specify an upper and lower limit).

# 14.2 Links & References

## 1. Exception Handling:

<https://docs.xojo.com/index.php/Exception>

## 2. Threads:

[http://en.wikipedia.org/wiki/Thread\\_\(computing\)](http://en.wikipedia.org/wiki/Thread_(computing))

## 3. Randomness:

<http://en.wikipedia.org/wiki/Randomness>

# 14.3 Review Questions

1. Which is more important: computational speed or interface responsiveness? Would you rather have a fast program that seemed to be locked up or a slower program that remained usable and informative? Are there situations where your answer might change?
2. What does a sloppy interface imply about the programmer? What does an elegant, modern interface imply?

# 14.4 Programming Challenge

Create a new app that uses a timer to track changes to the user's clipboard. Store new clipboard entries in a List Box, being sure not to duplicate any existing entries.

# Afterword



---

## CONTENTS

1. Thank You
2. About The Authors

# Thank You

Thanks for taking the time to read *Introduction to Programming with Xojo - Teacher Guide*.

If you have any questions, comments, or suggestions about anything in this book, please feel free to send an email to [docs@xojo.com](mailto:docs@xojo.com)



# About The Authors

## **BRAD RHINE**

---

Brad is a self-professed computer geek who has worked as a Computer Programmer, Web Developer, Technical Writer, Database Administrator, Assistant Director of Technology, and, briefly, Christmas Tree Salesman.

He is also a former columnist for XDev Magazine and has presented at the Xojo Developer Conference on many different topics.

Brad has spent most of his professional career working in the public school system.

When he's not writing code or writing about code, you'll find Brad playing his guitar, hanging out with his family, or running.

He lives in rural Pennsylvania with his wife and their two children, as well as a dog and two maladjusted cats.

## **PAUL LEFEBVRE**

---

Paul is a Xojo Engineer and in addition to programming, he helps update the documentation and examples. He has been working with computers since first using an Atari 400 back in 1983.

# Copyright & License

This work is copyright © 2012-2021 by Xojo, Inc.

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](#). This means that you are free to share the material in this book as long as you are not doing so for profit and you retain the attribution to Xojo, Inc.