Documenting Processing + Raspberry Pi

Elgin-Skye McLaren
emclaren@sfu.ca

Project Abstract

The goal of this project is to make existing Processing for Raspberry Pi documentation accessible through the development of a standalone website and a short series of introductory videos.
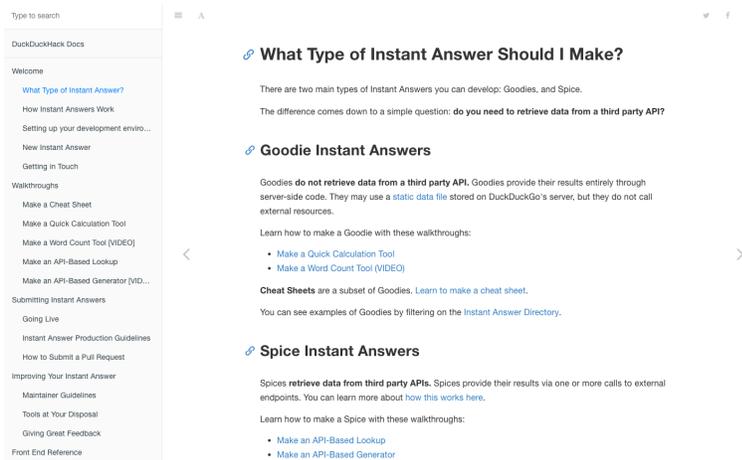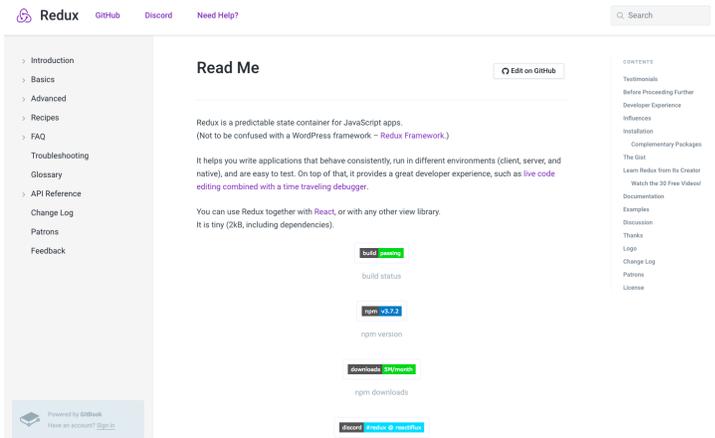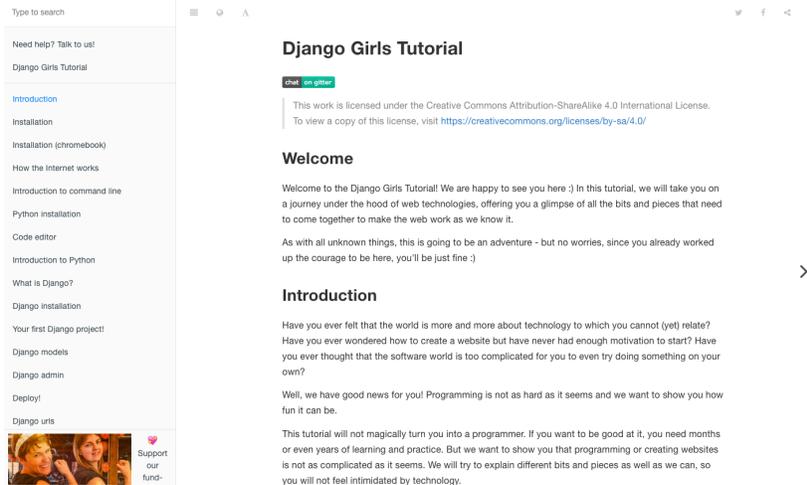

Project Description

Raspberry Pi computers are portable and inexpensive. When paired with Processing they can be particularly useful for creating installation art projects (and more). There is little information online about how to get started using Processing on the Raspberry Pi platform. I recognized this gap first-hand in my experience as a teaching assistant for a graduate-level tangible computing course.

To make Processing on Raspberry Pi more accessible, I propose developing a standalone website and a short series of three introductory videos. Gottfried Haider has written a significant amount of documentation which is not currently publicly distributed. In correspondence related to Google Summer of Code, he indicated an interest in publishing this work online. Based on this, I would like to create a new website to host this content. This project would involve building a website, porting over existing documentation, reviewing content, and adapting it for the web. The page would be formatted using a [GitBook](#) template (see fig 1).

As this documentation will be very text-oriented (and perhaps a bit intimidating for new users) I will also record a series of youtube videos to introduce Processing for Raspberry Pi. These will be stylistically similar to Daniel Shiffman's Coding Train videos. The first video will explain the benefits of using Processing on a Raspberry Pi; the second will show users how to set up Processing on their Raspberry Pi; and the third will showcase some project ideas and direct users to where to find out more (ie: the GitBook, and Processing's docs).

Fig 1. Examples of GitBooks templates from https://www.gitbook.com/

Django Girls Tutorial

Type to search

Need help? Talk to us!

Django Girls Tutorial

Introduction
Installation
Installation (chromebook)
How the Internet works
Introduction to command line
Python installation
Code editor
Introduction to Python
What is Django?
Django installation
Your first Django project!
Django models
Django admin
Deploy!
Django urls

Support our fund-

chat on gitter

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License.
To view a copy of this license, visit https://creativecommons.org/licenses/by-sa/4.0/

Welcome

Welcome to the Django Girls Tutorial! We are happy to see you here :) In this tutorial, we will take you on a journey under the hood of web technologies, offering you a glimpse of all the bits and pieces that need to come together to make the web work as we know it.

As with all unknown things, this is going to be an adventure - but no worries, since you already worked up the courage to be here, you'll be just fine :)

Introduction

Have you ever felt that the world is more and more about technology to which you cannot (yet) relate? Have you ever wondered how to create a website but have never had enough motivation to start? Have you ever thought that the software world is too complicated for you to even try doing something on your own?

Well, we have good news for you! Programming is not as hard as it seems and we want to show you how fun it can be.

This tutorial will not magically turn you into a programmer. If you want to be good at it, you need months or even years of learning and practice. But we want to show you that programming or creating websites is not as complicated as it seems. We will try to explain different bits and pieces as well as we can, so you will not feel intimidated by technology.

Redux

GitHub   Discord   Need Help?   Search

Introduction
Basics
Advanced
Recipes
FAQ
Troubleshooting
Glossary
API Reference
Change Log
Patrons
Feedback

Powered by GitBook
Have an account? Sign in

Read Me

Edit on GitHub

Redux is a predictable state container for JavaScript apps.
(Not to be confused with a WordPress framework – Redux Framework.)

It helps you write applications that behave consistently, run in different environments (client, server, and native), and are easy to test. On top of that, it provides a great developer experience, such as live code editing combined with a time traveling debugger.

You can use Redux together with React, or with any other view library.
It is tiny (2kB, including dependencies).

build passing
build status

npm v3.7.2
npm version

downloads 5M/month
npm downloads

discord #redux @ reactiflux

CONTENTS
Testimonials
Before Proceeding Further
Developer Experience
Influences
Installation
  Complementary Packages
The Gist
Learn Redux from Its Creator
  Watch the 30 Free Videos!
Documentation
Examples
Discussion
Thanks
Logo
Change Log
Patrons
License

Type to search

DuckDuckHack Docs

Welcome
  What Type of Instant Answer?
  How Instant Answers Work
  Setting up your development enviro...
  New Instant Answer
  Getting in Touch
Walkthroughs
  Make a Cheat Sheet
  Make a Quick Calculation Tool
  Make a Word Count Tool [VIDEO]
  Make an API-Based Lookup
  Make an API-Based Generator [VID...
Submitting Instant Answers
  Going Live
  Instant Answer Production Guidelines
  How to Submit a Pull Request
Improving Your Instant Answer
  Maintainer Guidelines
  Tools at Your Disposal
  Giving Great Feedback
Front End Reference

What Type of Instant Answer Should I Make?

There are two main types of Instant Answers you can develop: Goodies, and Spice.

The difference comes down to a simple question: **do you need to retrieve data from a third party API?**

Goodie Instant Answers

Goodies **do not retrieve data from a third party API.** Goodies provide their results entirely through server-side code. They may use a static data file stored on DuckDuckGo's server, but they do not call external resources.

Learn how to make a Goodie with these walkthroughs:

- Make a Quick Calculation Tool
- Make a Word Count Tool (VIDEO)

Cheat Sheets are a subset of Goodies. Learn to make a cheat sheet.

You can see examples of Goodies by filtering on the Instant Answer Directory.

Spice Instant Answers

Spices **retrieve data from third party APIs.** Spices provide their results via one or more calls to external endpoints. You can learn more about how this works here.

Learn how to make a Spice with these walkthroughs:

- Make an API-Based Lookup
- Make an API-Based Generator

Development Process

Community Bonding: April 23 – May 13
I will spend the community bonding period reviewing the *Prototyping with Raspberry Pi and Processing* manuscript, exploring Gitbooks' features, and engaging with the Processing community via the forum.

Coding Period: May 14 – Aug 06
The project's coding period has three phases. The goal of phase 1 is to build the website and add existing content. The goal of phase 2 is to review the content and adapt it as needed for the web. The goal of phase 3 is to create video content to help first-time users get started.

Week 1: May 14 – 20
- Work with mentor to identify any important customizations required for the GitBook Manual template
- Set up website codebase
- Plan the information architecture for the content

Week 2: May 21 – 27
- Add and format content from chapters 1 - 3 of existing documentation
- Create any unique styling required for the website
- Test on several browsers to ensure the content loads properly

Week 3: May 28 – June 3
- Add and format content from chapters 4 - 7 of existing documentation

Week 4: June 4 – June 10
- Add and format content from chapters 8 - 10 of existing documentation

Week 5: June 11 – June 17
- End of Phase 1: all content from the existing *Prototyping with Raspberry Pi and Processing* manuscript will exist in GitBook format
- Begin reviewing the content to find areas where the text/images/examples should be adapted to the medium of the web, or where additional examples may be beneficial

Week 6: June 18 – June 24:
- Continue reviewing the content and identify any specific additional information that should be created or adapted
- Submit any changes or updates to mentor for feedback

Week 7: June 24 – July 1
- Continue reviewing and updating the content
- Test the page and its content on various devices to ensure it is responsive

Week 8: July 2 – July 8
- End of phase 2 - The GitBook contents should be fully reviewed, and have content that has been adapted for the web.
- Write scripts for videos on the following topics:
  - Intro to Raspberry Pi & Processing (what can it be useful for, how is it different than processing on a PC) ~ 3 minutes
  - Installing Processing and running your first sketch ~10 minutes
  - Demo what you can do with Processing, referring users to the Processing documentation, and the new GitBook ~3 minutes.
- Create or find interesting Raspberry Pi projects to showcase in the final video

Week 9: July 9 – July 15
- Record video content
- Begin editing video content

Week 10: July 16 – July 22
- Finish editing video content and upload files to youtube

Week 11: July 23 – July 29
- Add reference to the youtube videos into the the GitBook page
- Disseminate the new GitBook page and videos via the Processing (and related) forums

Week 12: July 30 – Aug 06
- Buffer week – in case something during an earlier week takes an especially long time

More about you

I'm an MSc student at the School of Interactive Arts + Technology at Simon Fraser University in Vancouver, Canada. I research tangible computing and child computer interaction. I love getting people excited about coding. I teach programming workshops* at my university, at NodeSchool, and for Canada Learning Code and am a founder of Women Who Code's Vancouver chapter. Open source tools that lower the barriers to entry, like Processing, are invaluable to this work. I have led workshops on Processing, and often use it and p5.js in my personal projects.

My own background is not in computer science, but arts management. I learned to code through online tutorials (go Shiffman!), and a programming bootcamp. I have worked as a web developer for a major Canadian Telecom, completed programming courses in my MSc, and contributed to the development of a website for ACM SIGCHI as an executive committee volunteer. Further, I have experience creating videos to showcase my lab's research. I think I could make some valuable contributions to the development and documentation of Processing for ARM. To get up to speed I have begun reviewing and contributing to Processing's Github repos.

This is my first experience with open source contributing. My goal is to use the skills and knowledge gained through GSOC to encourage more women in my local network to get involved with open source.

My portfolio: elginskye.com
My github: github.com/emclaren
Linkedin: linkedin.com/in/elgin-skye/
Processing Forum ID: emclaren

* Specific workshops/classes I've led
Simon Fraser University
        - IAT 235 Information Design (UX Design/HTML5/CSS3)
        - IAT 339 Web Design & Development (UX Design/Pug/SASS/JS/Preprocessing/Git)
        - IAT 882 Tangible Computing (Arduino/Processing)
RED Academy Tech Bootcamp
        - Web Development Foundation Course (HTML5/SASS/JS/Git)
Nodeschool
        - Nodebots workshop (Johnny-Five JS Robotics platform)
Canada Learning Code (Girls Learning Code + Ladies Learning Code)
        - Introduction to Generative Art workshop (Processing)
        - Introduction to web development workshop (HTML5/CSS3)
Code and Coffee Meetup Group
        -Intro to MaxMSP Presentation
Conf and Coffee (scheduled April 2018)
        - Nodebots workshop (Johnny-Five JS Robotics platform)