# Circle Docs Revamp Project Notes 2019-08

*Author: Jan Joost van Zon*

*Location: Oosterhout, The Netherlands*

*Date: August, 2019*

## Contents

[Contents 1](#_Toc15816861)

[Project Outline 1](#_Toc15816862)

[Introduction 1](#_Toc15816863)

[Goals 1](#_Toc15816864)

[Limitations 1](#_Toc15816865)

[Problems 1](#_Toc15816866)

[Steps 1](#_Toc15816867)

[Notes 1](#_Toc15816868)

[2019-08-04 Notes converting version folders to source control history 1](#_Toc15816869)

## Project Outline

### Introduction

Circle is an unfinished programming language.

I spent thousands of hours spread out over a few decades with this idea of how to better express the internals of computers and programming language constructs.

Circle Docs are an unfinished programming language design. The same folder of docs contain texts about all sorts of software applications. But this project 'Circle Docs Revamp' is really about the programming language design.

### Goals

#### Main goals

* Work on project that seems to matter.
* Keep my head occupied with something else.
* Get the documents in a state better accessible to others.
* Make the documents usable for implementation.

#### Side goals

* Make a technical design

To highlight the challenges of how one might implement this as a working piece of software.

* Open source it.

### Limitations

* Do not program.
* Limit it to the 'Language' part of the docs (not 'Framework', 'Operating System' or 'Applications & Media')

### Problems

* The read uses terminology in a very specific way, that is not shared with my peers, therefor not easing readers into the material. Maybe an introduction would do. The 'older' versions actually seemed to give that more gradual intro, that takes the reader by the hand, leading them to how and why this is actually useful, rather than just plainly stating the shapes that the diagrams can contain.
* That said, I do want to quickly come to the raw definition of how the diagrams are built-up. To introduce the raw basic specs quickly.
* The work might be modularized. I am not attacted to how CSS3 is modularized, each piece of specs with a different state of being finished up. It seems messy. But I can employ the same organization to accept certain concepts are just more crystalized out than others, making it easier to share, even in an unfinished state, and stimulating keeping things separated and separately usable even when other parts are just really still messy.

### Steps

* Convert version folders to source control history.
* Reorganize files and folders (e.g. lone files in folders is not handy).
* Start reading / writing / reformulating.

## Notes

### 2019-08-04 Notes converting version folders to source control history

This is a lot of work.