University of Bergen
Department of Informatics

Title of your master thesis

Author: Your name

Supervisors: Name of supervisors

UNIVERSITETET I BERGEN

Det matematisk-naturvitenskapelige fakultet

April, 2024

Abstract

Lorem ipsum dolor sit amet, his veri singulis necessitatibus ad. Nec insolens periculis ex. Te pro purto eros error, nec alia graeci placerat cu. Hinc volutpat similique no qui, ad labitur mentitum democritum sea. Sale inimicus te eum.

No eros nemore impedit his, per at salutandi eloquentiam, ea semper euismod meliore sea. Mutat scaevola cotidieque cu mel. Eum an convenire tractatos, ei duo nulla molestie, quis hendrerit et vix. In aliquam intellegam philosophia sea. At quo bonorum adipisci. Eros labitur deleniti ius in, sonet congue ius at, pro suas meis habeo no.

Acknowledgements

Est suavitate gubergren referrentur an, ex mea dolor eloquentiam, novum ludus suscipit in nec. Ea mea essent prompta constituam, has ut novum prodesset vulputate. Ad noster electram pri, nec sint accusamus dissentias at. Est ad laoreet fierent invidunt, ut per assueverit conclusionemque. An electram efficiendi mea.

`Your name`
Friday 12th April, 2024

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

Natum mucius vim id. Tota detracto ei sed, id sumo sapientem sed. Vim in nostro latine gloriatur, cetero vocent vim id. Erat sanctus eam te, nec assueverit necessitatibus ex, id delectus fabellas has.

Lorem ipsum dolor sit amet, iisque feugait quo eu, sed vocent commodo aliquid an. Minim suavitate dissentiet te eos. Dicunt eirmod adolescens no sed. Esse nonumy melius an mel, mei ut maiorum luptatum. Eu eum iudico scripta, movet option assueverit mel ex, mea at odio noluisse efficiendi. Ad vidisse atomorum conceptam quo, saepe volumus philosophia eos eu, delenit conceptam no usu.

Vituperata sadipscing deterruisset ei mel, at qui nonumy blandit. Delectus dissentiet et sea, ut rebum regione numquam nam, cum ex augue constituto. Te per nihil semper. Posse voluptatum qui an, aliquando democritum disputando id quo, everti perpetua cu vim. Laudem fabellas mei an, eu reprimique quaerendum usu. Quidam prompta fabellas ne est.

1.1 Background

Lorem ipsum dolor sit amet, cu graecis propriae sea. Eam feugiat docendi an, ei scripta blandit pri. Nonumes delicata reprimique nam ut. Eu suas alterum concludaturque est, ferri mucius sensibus id sed [1].

We can do glossary for acronymes and abriviations also: Software as a Service (SaaS). As you see the first time it is used, the full version is used, but the second time we use SaaS the short form is used. It is also a link to the lookup.

1.1.1 Listings

You can do listings, like in Listing 1.1

```
$ java -jar myAwesomeCode.jar
```

Listing 1.1: Look at this cool listing. Find the rest in Appendix A.1

You can also do language highlighting for instance with Golang: And in line 6 of Listing 1.2 you can see that we can ref to lines in listings.

```go
1 package main
2
3 import "fmt"
4
5 func main() {
6     fmt.Println("hello world")
7 }
```

Listing 1.2: Hello world in Golang

1.1.2 Figures

Example of a centred figure

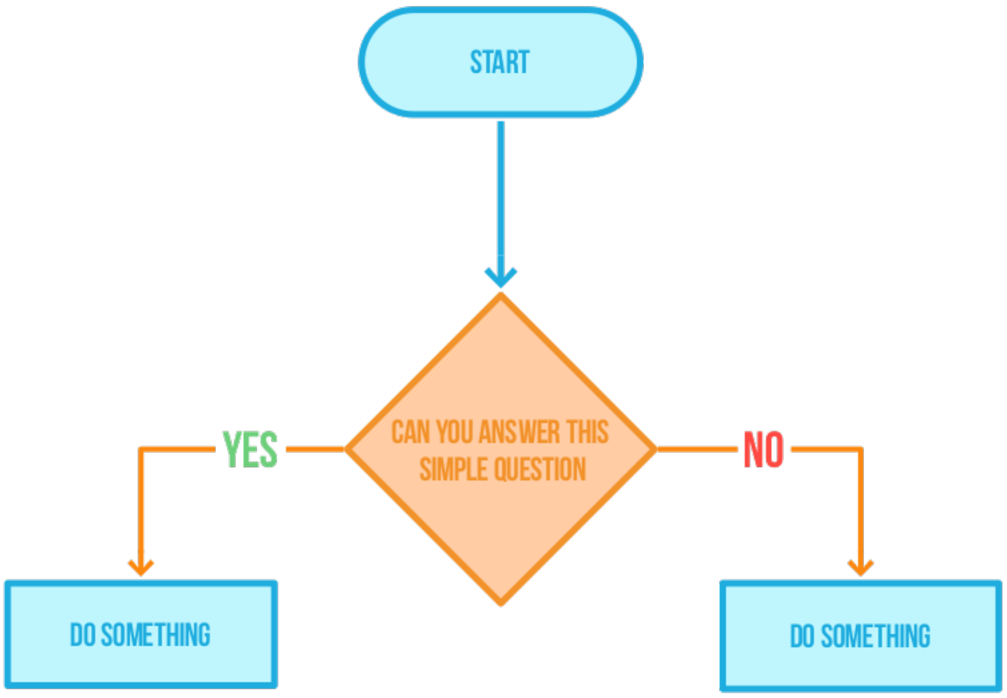

Figure 1.1: Caption for flowchart

Credit: Acme company makes everything `https://acme.com/`

1.1.3 Tables

We can also do tables. Protip: use `https://www.tablesgenerator.com/` for generating tables.

Table 1.1: Caption of table

Title1	Title2	Title3
data1	data2	data3

1.1.4 Git

Git is fun, use it!

1.2 Collecting User Feedback for Syntactic Proposals

The goal for this project is to utilize users familiarity with their own code to gain early and worthwhile user feedback on new syntactic proposals for EcmaScript.

1.2.1 The core idea

Users of EcmaScript have a familiarity with code they themselves have written. This means they have knowledge of how their own code works and why they might have written it a certain way. This project aims to utilize this pre-exisiting knowledge to showcase new proposals for EcmaScript. Showcasing proposals this way will allow users to focus on what the proposal actually entails, instead of focusing on the examples written by the proposal author.

A proposal for EcmaScript is a suggestion for a change to the language. These changes come with a set of problems that if the proposal is included as a part of EcmaScript, those problems should be solved by utilizing the additions contained within the proposal.

Further in this chapter, we will be discussing the *old* and *new* way of solving a problem. What we are referring to in this case is whatever set of problems a proposal is trying to solve, if that proposal is allowed into EcmaScript as part of the language, there will be a *new* way of solving said problems. The *old* way is the current status quo when the proposal is not part of EcmaScript, and the *new* way is when the proposal is part of EcmaScript and we are utilizing the new features of said proposal.

1.2.2 Applying a proposal

The way this project will use the pre-existing knowledge the user harbors of their own code is to use it as a base, and *apply* the proposal to that base.

When a proposal is *applied* to a piece of code there are some main steps. A code snippet where the proposal might be used has to be identified. That same code snippet has to be transformed to use the new features the proposal implements. Then both the original and the transformed code snippet has to be showcased to the user.

Identifying a code snippet where the proposal might be utilized, the problem a proposal is solving allows us to iden

1.2.3 Syntactic proposal

A proposal for EcmaScript is a suggested change for the language, in the case of EcmaScript this comes in the form of an addition to the language, as EcmaScript does not allow for breaking changes. There are many different kinds of proposals, one such example is a Syntactic proposal

A syntactic proposal, is a proposal that contains only changes to the syntax of a language. This means, the proposal contains either no, or very limited change to functionality, and no changes to semantics.

This is an example of a very simple syntactic proposal.

Consider a imaginary proposal **optional let to int for declaring numerical literal variables**. This proposal describes adding an optional keyword for declaring numerical variables if the expression of the declaration is a numerical literal.

This proposal will look something like this:

```
// Original code
let x = 100;
let b = "Some String";
let c = 200;

// Code after application of proposal
int x = 100;
let b = "Some String";
let c = 200;
```

Listing 1.3: Example of imaginary proposal **optional let to int for declaring numerical literal variables**

See that in 1.3 the change is optional, and is not applied to the declaration of c, but it is applied to the declaration of x. Since the change is optional to use, and essentially is just *syntax sugar*, this proposal does not make any changes to functionality or semantics, and can therefore be categorized as a syntactic proposal.

1.2.4 Showcasing syntactic proposals

The application of a proposal on a program will then be utilized as a way of presenting the proposal to developers using EcmaScript(hereby referred to as users). If the code used for the application is already familiar to the user, we can severely limit the amount of time spent to understand the functionality of the initial code used in the application of the proposal. This allows the users more focus on the actual change presented by the proposal, which in turn might present itself in better feedback on the proposal, as well as presenting issues about the proposal earlier in the process.

This showcase is done by, The user being presented with two pieces of code. One being the original version with no changes, and the other being the transformed code after, this will allow the user to see how it was written before, and see how the proposal could have been applied here. This will give a low barrier of entry to understand a proposal and give meaningful feedback to the committee.

This paper does not focus on showcasing and gathering user feedback, however focusing more on the process of applying a proposal to some piece of code, while maintaining its functionality.

1.2.5 DSL NAME HERE?

In order to match ......

1.2.6 Using the DSL NAME HERE?with an actual syntactic proposal

Glossary

Git Git is a Version Control System (VCS) for tracking changes in computer files and coordinating work on those files among multiple people.

List of Acronyms and Abbreviations

SaaS	Software as a Service.
VCS	Version Control System.

Bibliography

[1] Diego Ongaro and John Ousterhout. In search of an understandable consensus algorithm. In *Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference*, USENIX ATC'14, pages 305–320, Berkeley, CA, USA, 2014. USENIX Association. ISBN 978-1-931971-10-2.

Appendix A

Generated code from Protocol buffers

```
System.out.println("Hello Mars");
```

Listing A.1: Source code of something