

# *Setanta* - An Irish programming language and learning environment

Interim Report  
CS460 Final Year Project  
2019 - 2020

Student: Eoin Davey - 16334926      Supervisor: Dr. Barak Pearlmutter

December 12, 2019

## Goals of the project

The primary goal of this project is to create a new, modern programming language built around the Irish language, *Setanta*, with an intuitive online learning environment, allowing the language to be used as an educational tool, with no installation required.

*Setanta* is a modern, dynamic, object oriented language, whose domain is education. It's easily accessible and intuitive, it will feel familiar to anyone with some programming experience, while being friendly enough to learn as a first language. *Setanta* takes inspiration from the Irish language and culture, for simple things like it's keywords, down to it's deeper semantics.

The *Setanta* learning environment provides an intuitive interface to program *Setanta* with. It provides a modern text editor, an I/O console, as well as a powerful API to interact with a custom graphics environment, all running in the browser.

The project also involves development of new tools to help in programming language implementation for the web, namely building an expressive, powerful parser generator for TypeScript. This parser generator is then used to develop the interpreter for *Setanta*.

## Background

English is well established as the lingua franca of the programming world, even languages developed outside the Anglosphere are designed to be written in English, for example *Ruby* is from Japan and *Lua* is from Brazil, but both are designed around the English language. Part of the aim of this project is to explore what syntactic and semantic constructs that a language not designed around English might have.

The Irish language is in daily use around Ireland, it is a strong living language in many areas, and has a strong community of enthusiastic speakers. It is the primary language in many schools, both primary and secondary, around the country. Ireland's education system is beginning to bring more and more Computer Science education into the school curriculum.

For this reason and many others, there will be more and more Irish speaking people looking to get started learning programming, and what better way than by learning on an accessible platform, in a modern language, and in their own native tongue.

As *Setanta* is to be run in the browser, it is written in a web friendly language, specifically TypeScript, a strongly typed JavaScript variant. However, there was no existing parser generator for TypeScript in existence. Part of this project therefore became to create a fully functional parser generator for TypeScript, and to then use it to build the *Setanta* interpreter.

## Progress to Date

There has been much progress made on the project so far, below I'll outline the progress made on the 3 core components.

### *tsPEG* - The TypeScript Parser Generator

The TypeScript parser generator, named *tsPEG* is effectively complete, It is [published on the NPM package repository](#), currently at version 1.1.6.

*tsPEG* is a fully featured parser generator, built around the PEG grammar formalisation (a variant of Context Free Grammars). *tsPEG* is capable of generating parsers for any context free language, and even some context sensitive ones. It's built on top of an innovative regex based implicit lexing system, meaning no separate lexical analyser is required.

*tsPEG* is self-hosting, the grammar for *tsPEG* grammars is defined itself in the *tsPEG* grammar. This serves as a strong verification of *tsPEG*'s power.

When I announced *tsPEG* on relevant channels online, it received strong positive attention, peaking at around 350 weekly downloads.

### *Setanta* - The Irish Programming Language - Teanga Ríomhchlárúchain as Gaeilge

At the time of writing *Setanta* is well under way to completion. It's code can be found on [Github](#). *Setanta* supports a whole list of features that you expect from a modern language. It has a REPL, supports all standard maths operators, dynamically typed, lexically scoped, first class functions, higher order functions, closures, a few types of loops and conditionals etc. It will eventually also support classes and inheritance, as well as asynchronous execution.

The interpreter for *Setanta* is being written in TypeScript, using a parser generated by *tsPEG*. It is an interpreted language that can be executed client-side in the browser, as well as locally through the node JavaScript engine. The syntax of *Setanta* will be familiar to those used to the C family of languages, but is an original syntax that borrows some of it's constructs from the grammatical structure of the Irish language itself.

### The *Setanta* learning environment

The online learning environment for *Setanta* has a [live demo online](#) and available now. The environment has a text editor, an IO console, a graphics display panel, and a high level OOP API for interacting with the drawing panel, allowing the user to program simple games and demos. The learning environment MVP was configured and developed before the *Setanta* interpreter was ready, so originally used JavaScript as the language you could run, now that the *Setanta* interpreter is operational, it is now executable. However as *Setanta* does not yet support the OOP operations required to interact with the graphics library API, the graphics are currently disabled.

## Problems Encountered

The biggest initial problem for *Setanta* was the lack of a sufficient parser generator for TypeScript. I had decided to do the project in TypeScript, as I wanted to run the project in the browser, but I didn't want to use JavaScript because of it's weak typing system. WebAssembly was considered, but I found it wasn't mature enough to be worth experimenting with for this project. After some discussion with my supervisor it was decided it would be a good addition to the project to create my own TypeScript parser generator, *tsPEG*. I took inspiration from some other parser generators, largely from textX, PEG.js and pgen, to create a fully featured one for TypeScript. This also served as a great way to learn the ins and outs of the TypeScript type system.

## Planned Steps to Completion