Hello and Welcome, my name is Alexander Smith and I’m a final year computer science student at University College Cork.

Throughout this period, I have worked with multiple programming languages including Python, Java, SQL and JavaScript to mention a few.

During this time I noticed a common theme while working on my various assignments, internships and public hackathons like Googles Hashcode - I noticed that I would often need to load, examine and transform source data before it could be used for the task at hand, no matter what the language or assignment.

It was often inconvenient and ad-hoc to effectively construct simple data processing workflows and could vary greatly depending on the programming language being used.

This is what motivated me to create Hoshie Lang

----------- NEXT -----------------

Hoshie Lang is a convenient and simple to use data processing language designed with the primary goal of making is easy to create dataflows in a matter of minutes.

Hoshi Lang is a Declarative and Functional programming language with immutable data types.

* Declarative: The programmer can concentrate on the logic needed without worrying about the flow of the program, no loops!
* Functional: Hoshi Lang supports pure functions, ensuring logic can be reused easily.
* Immutable Data: Programmer does not need to concern themselves with memory management or data structure life spans.

Hoshie Lang uses the following analogy to further simplify the declarative programming style:

* Pipeline: The data processing takes place inside a “pipeline”, the programmer can think of the data continuously “flowing” through the pipe.
* Activity: Inside the pipeline, there can be many “activities”, an activity is something that changes the data is some way, either by filtering out rows or by altering the row content.
* Sensors: These can also be placed anywhere in the pipeline and they simply “observe” the data rows as they flow by. They can be used to calculate things like “Row Count”, “Min / Max Values” or even “Statistical Summaries”.

It is worth noting that an entire “pipeline” is also an “activity” again encouraging code reuse.

Combining all these elements has resulted in a simple, intuitive language for processing data.

-------- NEXT ---------------

For any programming language to be truly easy to use, it will need a good Integrated Development Environment. As part of this project I also created an extension for the very popular opensource IDE “VS Code”, with the following features:

* Syntax colouring
* Syntax error highlighting and reporting as you type
* Outline summary
* Tooltip information
* Ability to compile and run the active Hoshie Lang file.

Thus, making Hoshi Lang programming even more enjoyable.

------ NEXT ---------

Now Enough of the talk let’s give you an example of implementing some data processing – in this example we are going to process a JSON file with 1000 (fake) people, containing name, age and job title and calculate the age distribution for all “Architects”:

""" nice, elegant example here with the typing for video. """

----- Next ---------

Now you may be thinking yeah Alex this language looks great but what if I want to target different platforms, like Spark or Hadoop?

Well don’t worry, The Hoshie Lang Code Generator has been completely encapsulated, so that targeting different runtime environments and languages is just a matter of adding a new code generator to the project.

It is worth mentioning that the default Code Generator targets JavaScript and uses modern “Generator” methodology to ensure that the data is genuinely streamed through the activities one row at a time.

Thanks for watching this quick overview of Hoshie Lang and I am excited to answer any questions you may have at my Virtual Booth.