**Julia Style:**

* Use type stable functions within Julia.
* Use functions for everything. Julia automatically tries to optimize functions.
* Julia style documentation:
* <https://docs.julialang.org/en/v1/manual/style-guide/>
* Use blue style:
* <https://github.com/JuliaDiff/BlueStyle>

**Unit Tests:**

* Use: Using Tests
* You can test all sorts of things ie (data compared to real world data, code to see if it works, etc).

**Documenting Code:**

* Document.jl should be used.
* Make sure you document every single function.

**Benchmarking:**

* Tells you efficient the code is.
* Benchmarktools.jl
* Try to reduce allocations by pre-allocating things. Don’t change the size of your vectors if you don’t need to.
* @code\_warntype tells you about type stability of a function. It tells you which variables of type any.
* Try to avoid using global variables and type stabilities.
* @profview tells you what is on each line and which lines take the most amount of time.

**Inplace Functions:**

* Add exclamation at the end of the name to show its an in-place function.
* Use a dot in front of equals sign to modify in place.
* It allows you to modify things without having to re-create the vectors.

**Static Arrays**

* Static arrays don’t allocate more memory than they need to. The issue is you can’t update any of the entries. Mvector allows you to update those entries. Can’t change the size.