HOW IT STARTED

- Got sick of **Dynamic** languages
- Looked for something: **CLEAN, FAST, SAFE, STABLE**
- The **Language Authors** are SICK

FIRST IMPRESSIONS

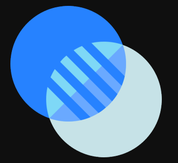
- UGLY
- PRIMITIVE
- TOO SIMPLE

EXPECTATIONS

- Loads of **frameworks**
- **Clever** ways of doing things
- Many **gotchas** & magic

SIMILARITIES

- Easy to **prototype**
- Can get **ugly**
- Very **accessible** & **productive**



DIFFERENCES

- **Statically typed** nature
- Mostly used for **Backend/Systems** only
- **Stable** without breaking changes
- **Simple Design**
- Encourages **libraries** & **concepts** vs fully fledged frameworks
- No **centralised** package **repository**
- Powerful toolchain & standard library
- Workhorse Performance

THINGS I LIKE

- **Simplicity**
- **Robustness** & Strictness
- Safety & **Stability**
- **Productive**
- Can get **hired without Go experience**

THINGS I DON'T LIKE

- Too **simple**
- Can get **ugly**
- Small **adoption**

FIRST STEPS

- A **tour of Go**
- **Videos/courses** & **Articles** & **blogposts**
- Get a **job**

RECOMMENDATIONS

- Change your **mindset**
- Focus on **concepts** not **frameworks**
- Get used to **magicless** code