HOW IT STARTED

- Got sick of **Dynamic** languages

- Looked for something: **CLEAN, FAST, SAFE, STABLE**

- The **Language Authors** are SICK

FIRST STEPS

- A **tour of Go**
- Couple **videos** & **courses**
- Get a **job**

FIRST IMPRESSIONS

🟩 UGLY

🟩 PRIMITIVE

🟩 TOO SIMPLE

EXPECTATIONS

🟩 Loads of **frameworks**

🟩 **Clever** ways of doing things
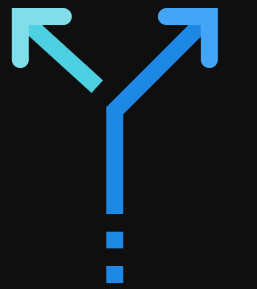
🟩 Many **gotchas** & magic

SIMILARITIES

- Easy to **prototype**

- Can get **ugly**

- Very **accessible** & **productive**

DIFFERENCES

- **Static**ally typed nature

- **Stable** without breaking changes

- **Little** to no **changes** in the ecosystem

- **Simple Design**

- Encourages **libraries** & **concepts** vs fully fledged frameworks

- No **centralised** package **repository**

THINGS I LIKE

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

THINGS I DON'T LIKE

- Too **simple**
- Can get **ugly**
- Small **adoption**
- Hard to find a **job**

DOS

- Change your **mindset**

- Focus on **concepts**

- Don't use the language for **everything**

- Get used to **magicless** code

DON'TS

- Don't focus on **frameworks**

- Don't violate **liberty**

- Don't use libs & frameworks for **everything**