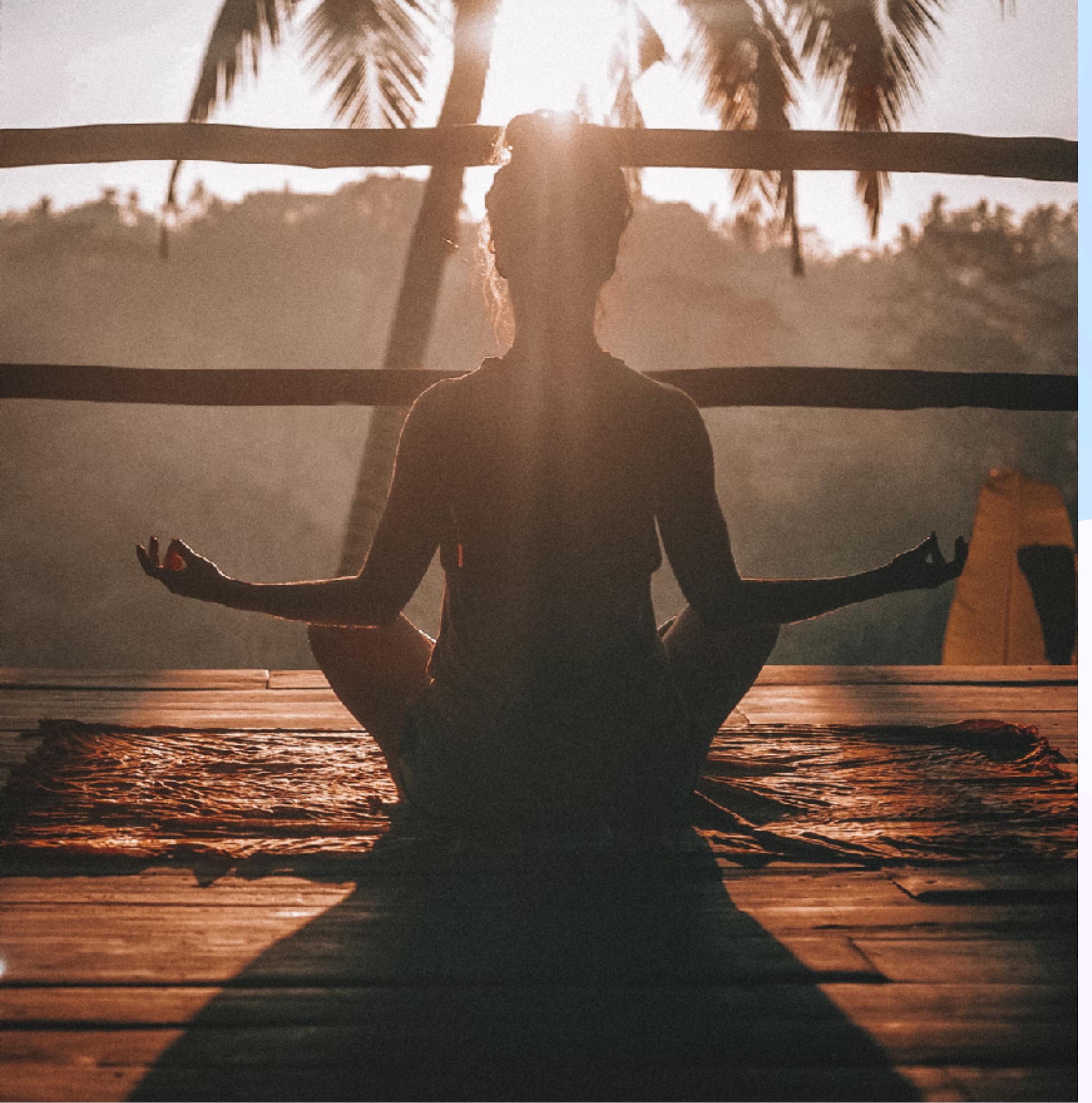


Software Engineering

Game time

ຝຶກສຕິ ເກມຄໍາ ໂຈມຕີ



Human Computer

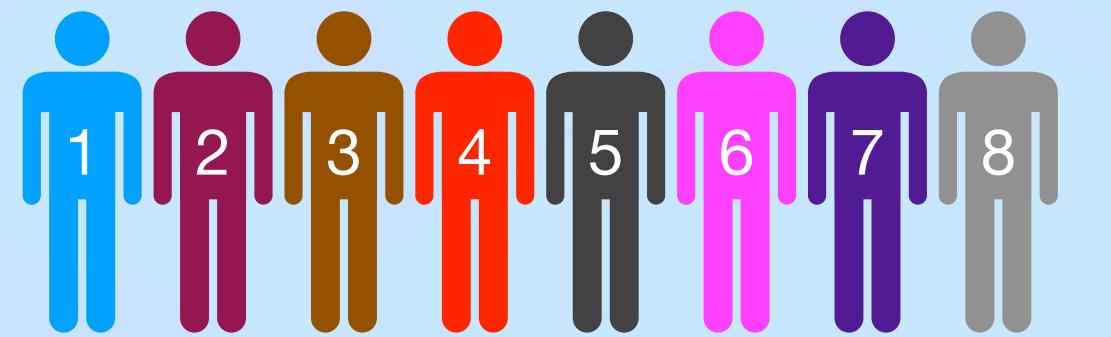
Move It!



Let's Split Into Groups

7–8 people per group

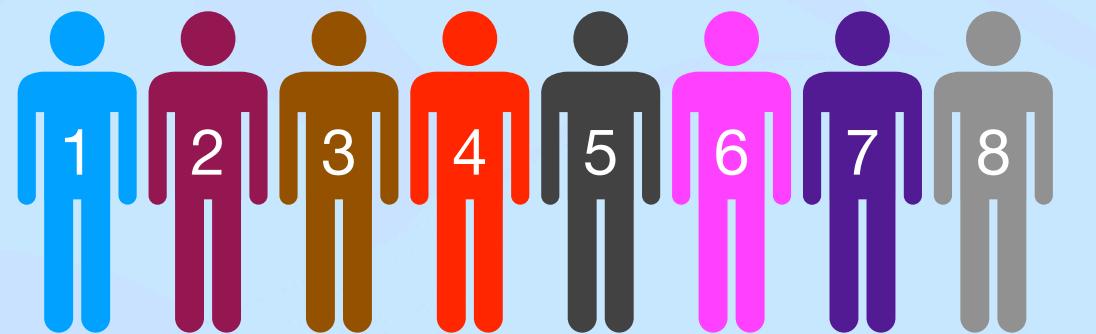
Group:



Let's Split Into Groups

Six objects

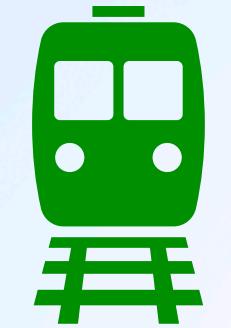
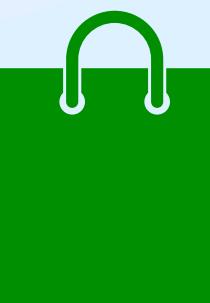
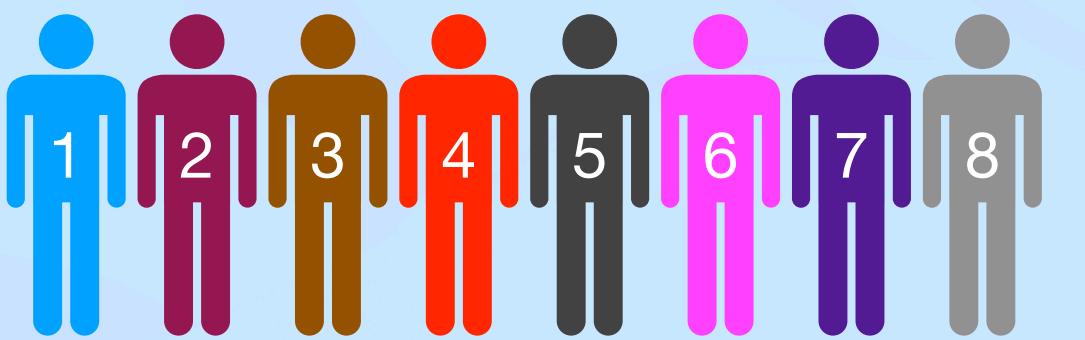
Group:



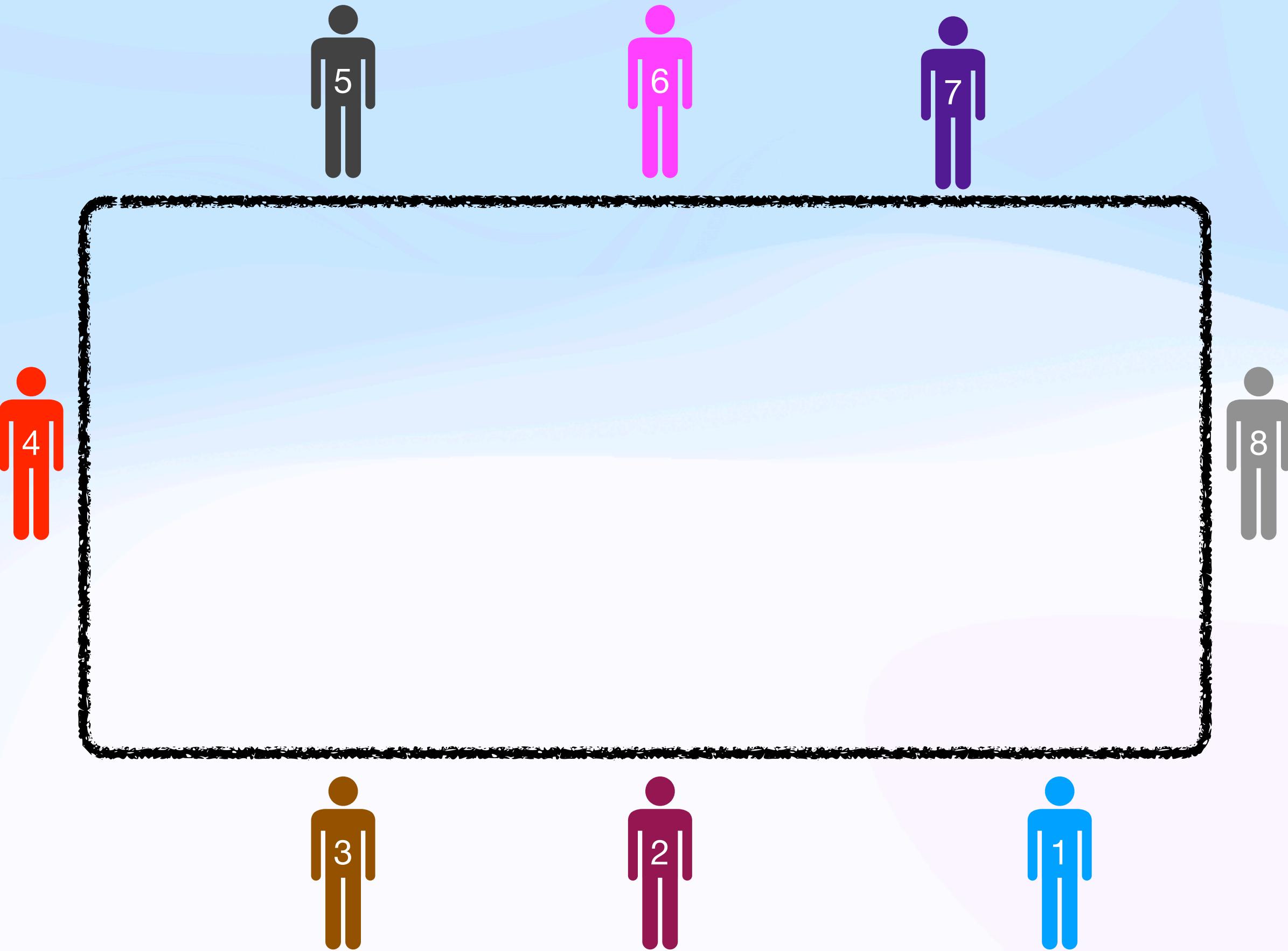
Object:



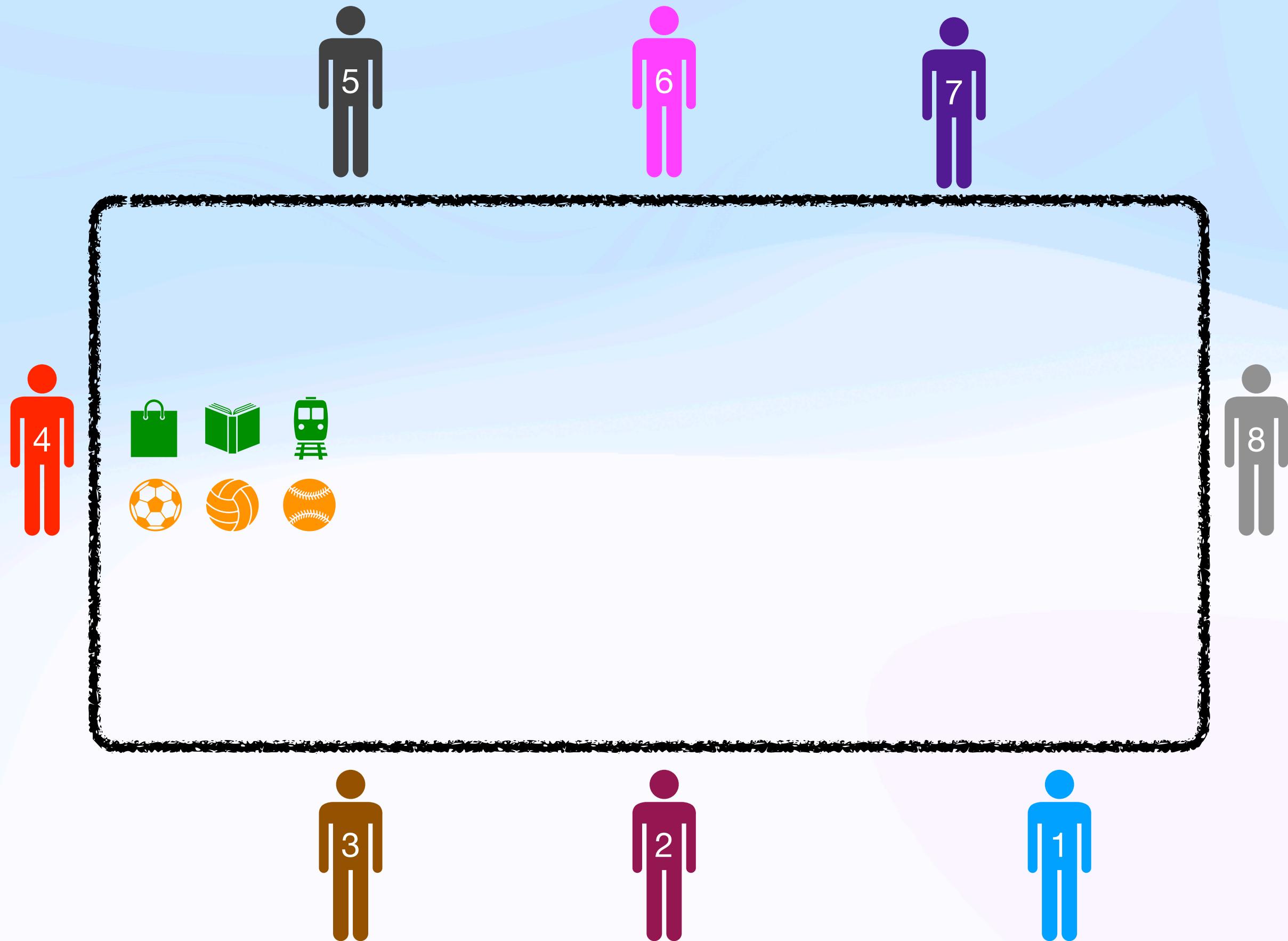
Allocate space



Allocate space



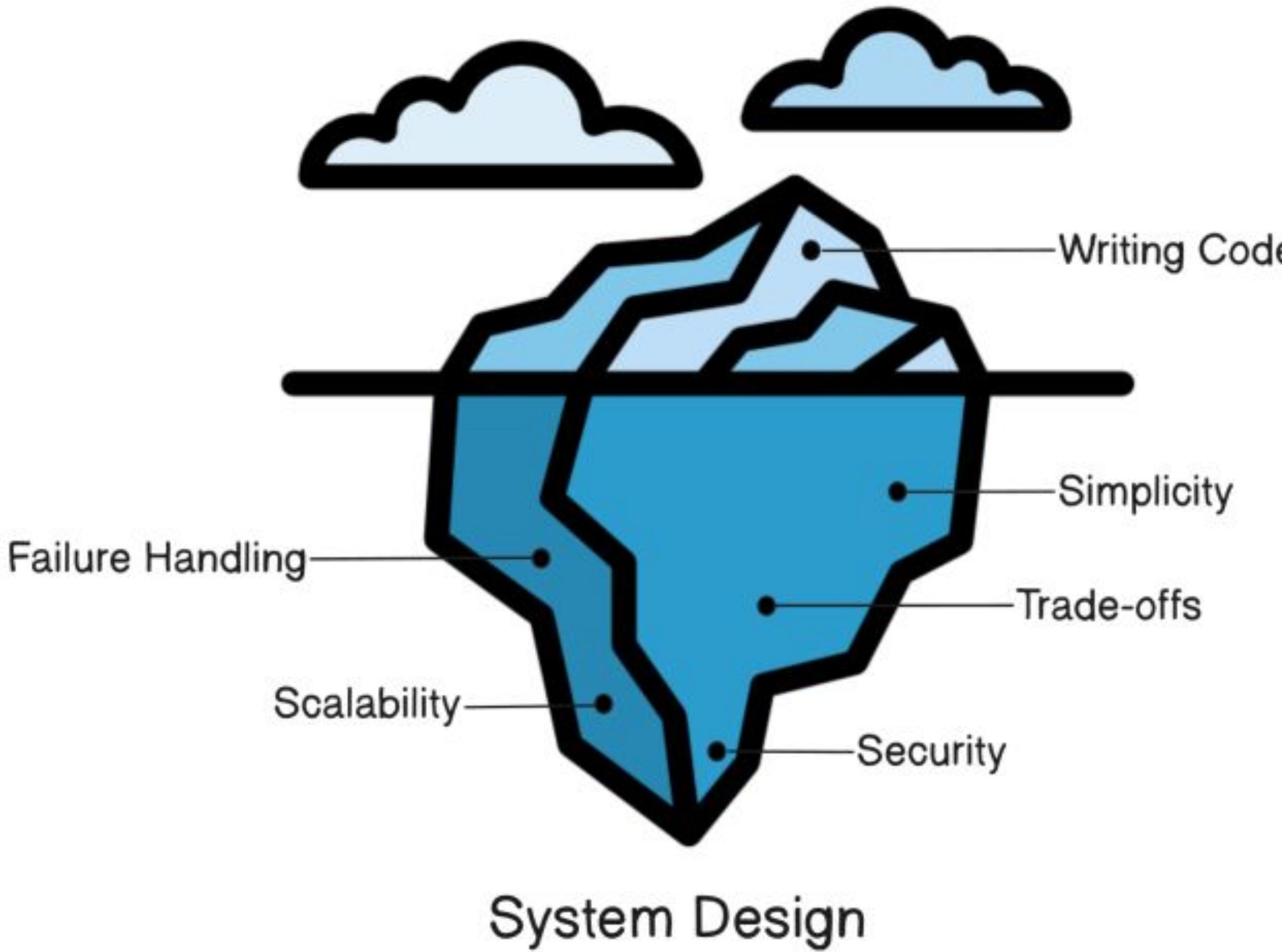
Allocate space



What is the software engineer?

Good engineers write code.
Great engineers make good decisions.

@rauljuncoV



**Software engineering is what
happens to **programming** when
you add **time** and other
programmers**

Russ Cox

Software engineering

= **programming + time + people**



Be a “**better**” engineer



Disclaimer

- I'm sharing my own personal thoughts
- I'm not here to give advice—just to share an ideas! 😊

Why an engineer?

Why need to be better?

Why an engineer?

Why need to be better?

Engineer is about
"building stuff"
(not talking, building!)



**Make things happen,
don't just talk about
them**





We need to refactoring

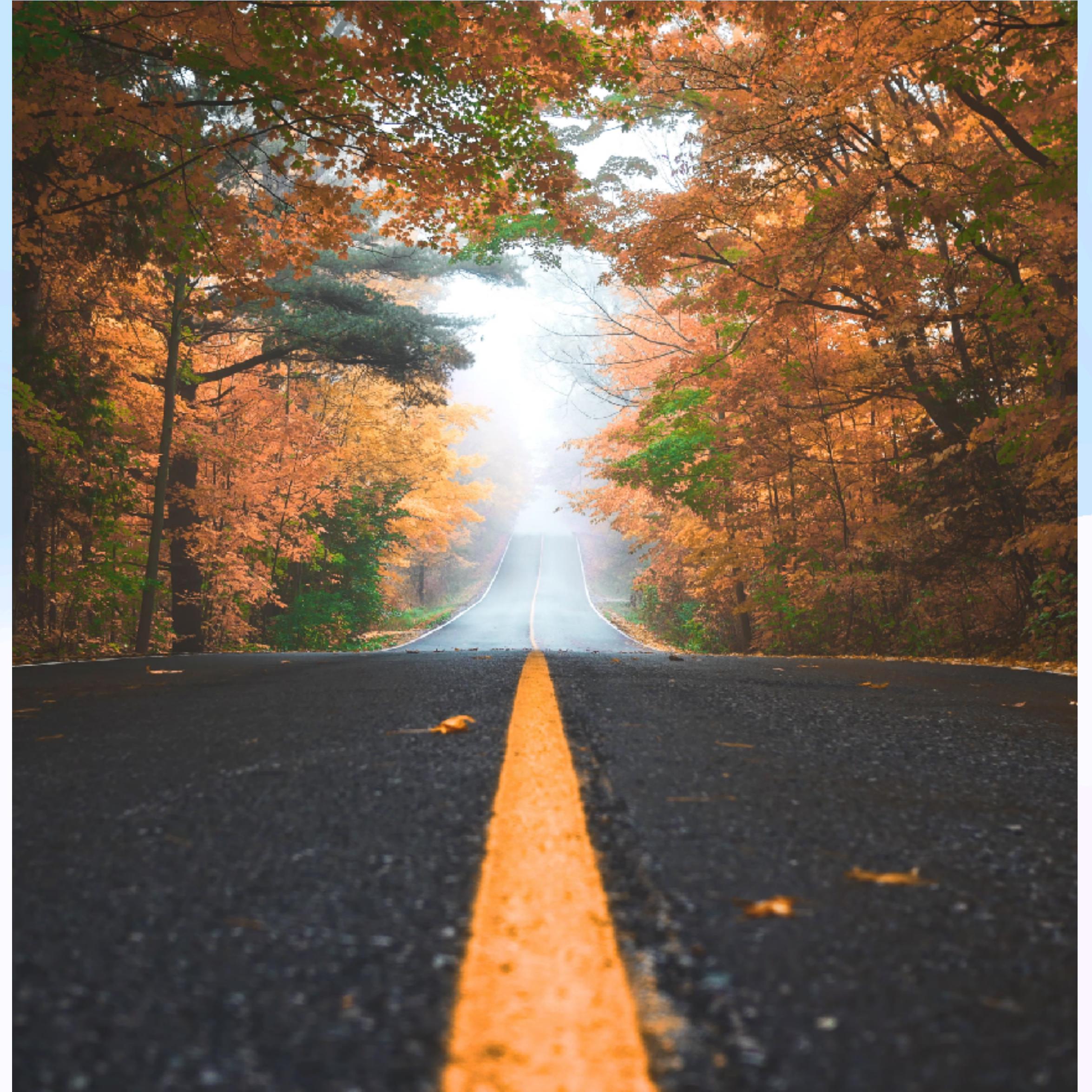


I already did some refactoring, here is the result

```
31     def __init__(self, path=None, debug=False):
32         self.file = None
33         self.fingerprints = set()
34         self.logdups = True
35         self.debug = debug
36         self.logger = logging.getLogger(__name__)
37         if path:
38             self.file = open(os.path.join(settings['job_dir'], path), 'a')
39             self.file.seek(0)
40             self.fingerprints.update(fp.strip() for fp in self.file)
41
42     @classmethod
43     def from_settings(cls, settings):
44         debug = settings.getbool('superfluous.log_dups', False)
45         return cls(job_dir(settings), debug)
46
47     def request_seen(self, request):
48         fp = self.request_fingerprint(request)
49         if fp in self.fingerprints:
50             return True
51         self.fingerprints.add(fp)
52         if self.file:
53             self.file.write(fp + os.linesep)
54
55     def request_fingerprint(self, request):
56         return request_fingerprint(request)
```

Why not "the best", just better?

**Greatness is a
journey of growth,
not a point of arrival**





**Anyone can grow and
become better**

Growth mindset vs Fixed mindset

Carol Dweck — psychologist

GROWTH MINDSET VS FIXED MINDSET



Failure is an
opportunity to
grow

I can **learn** to
do anything I
want

Challenges
help me to
grow

I'm **inspired** by
the success of
others

I like to try
new things

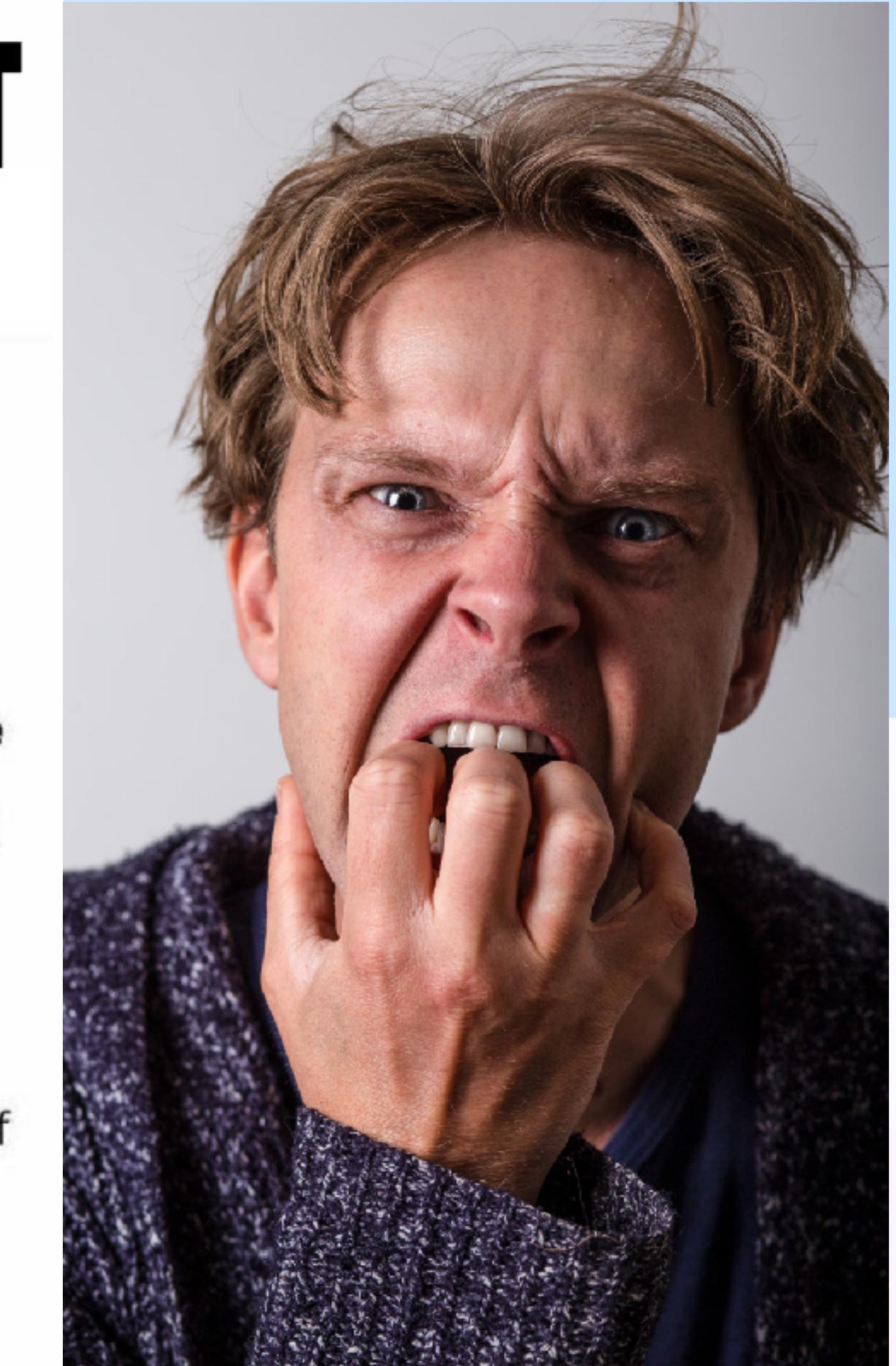
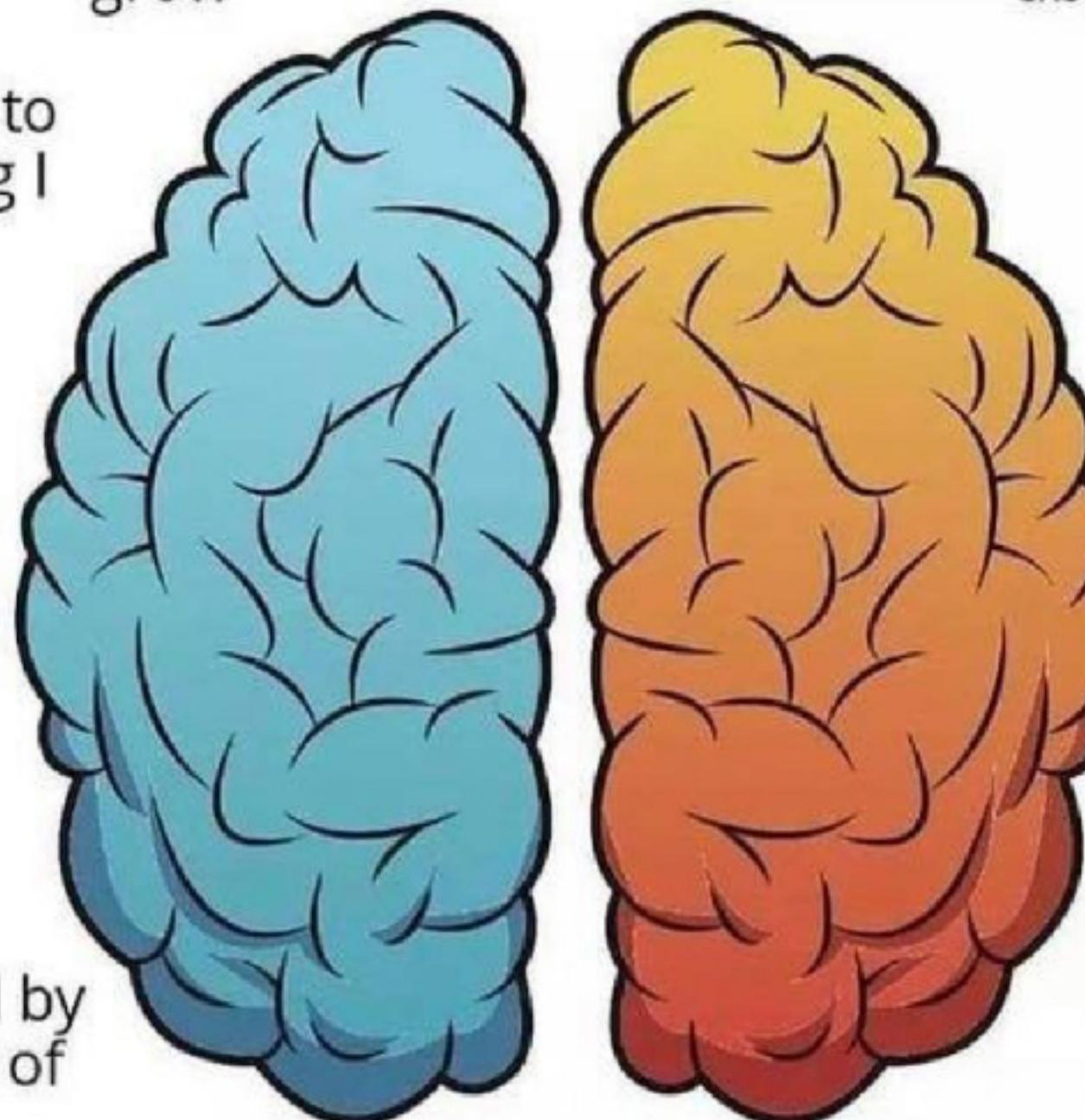
Failure is the
limit of my
abilities

I'm either
good at it or
I'm not

I **don't like**
to be
challenged

Jealous by
the success of
the others

I stick with the
old methods

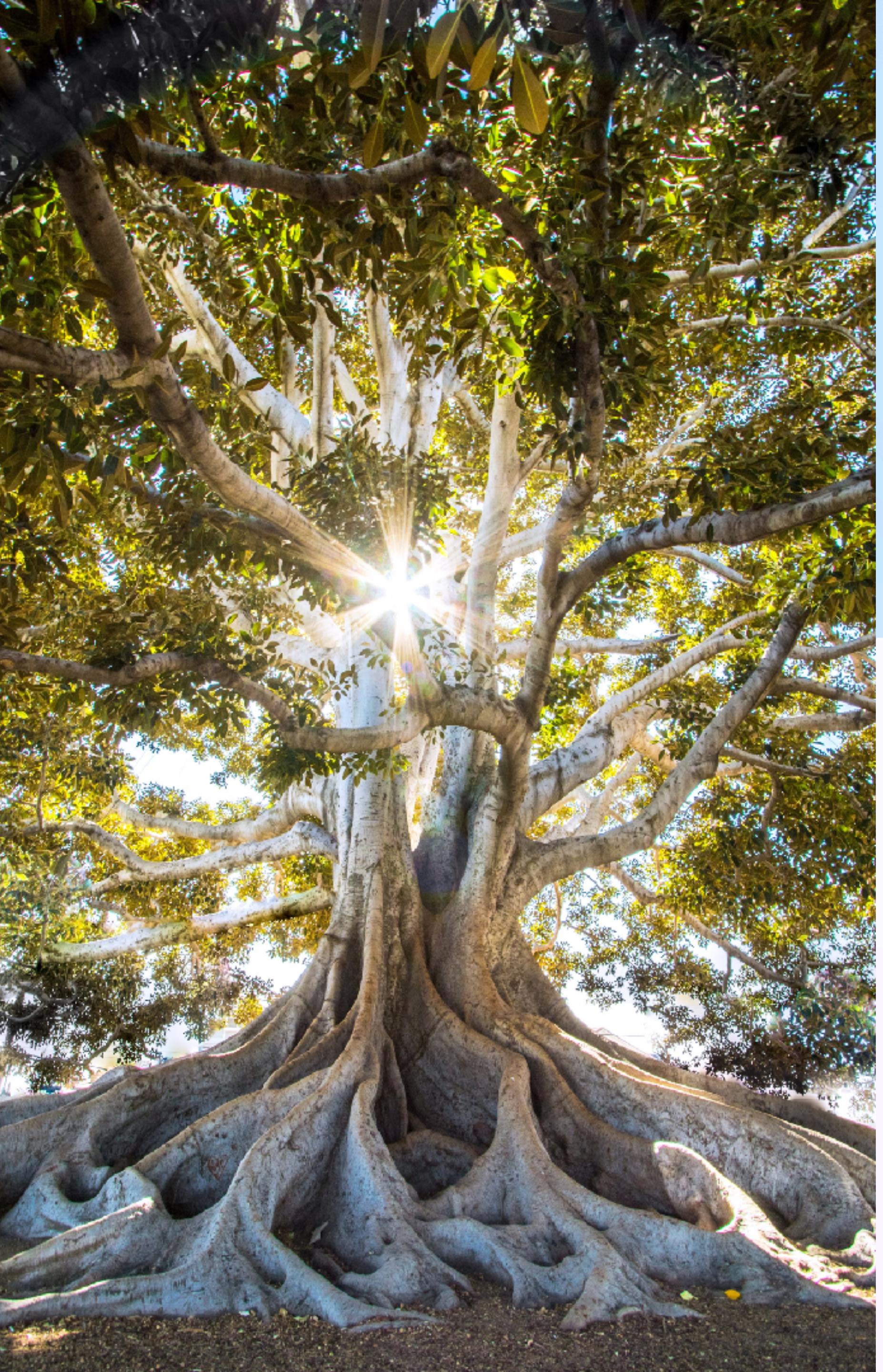


I can't do XYZ



If I practice for a few months,

I'll get better at XYZ





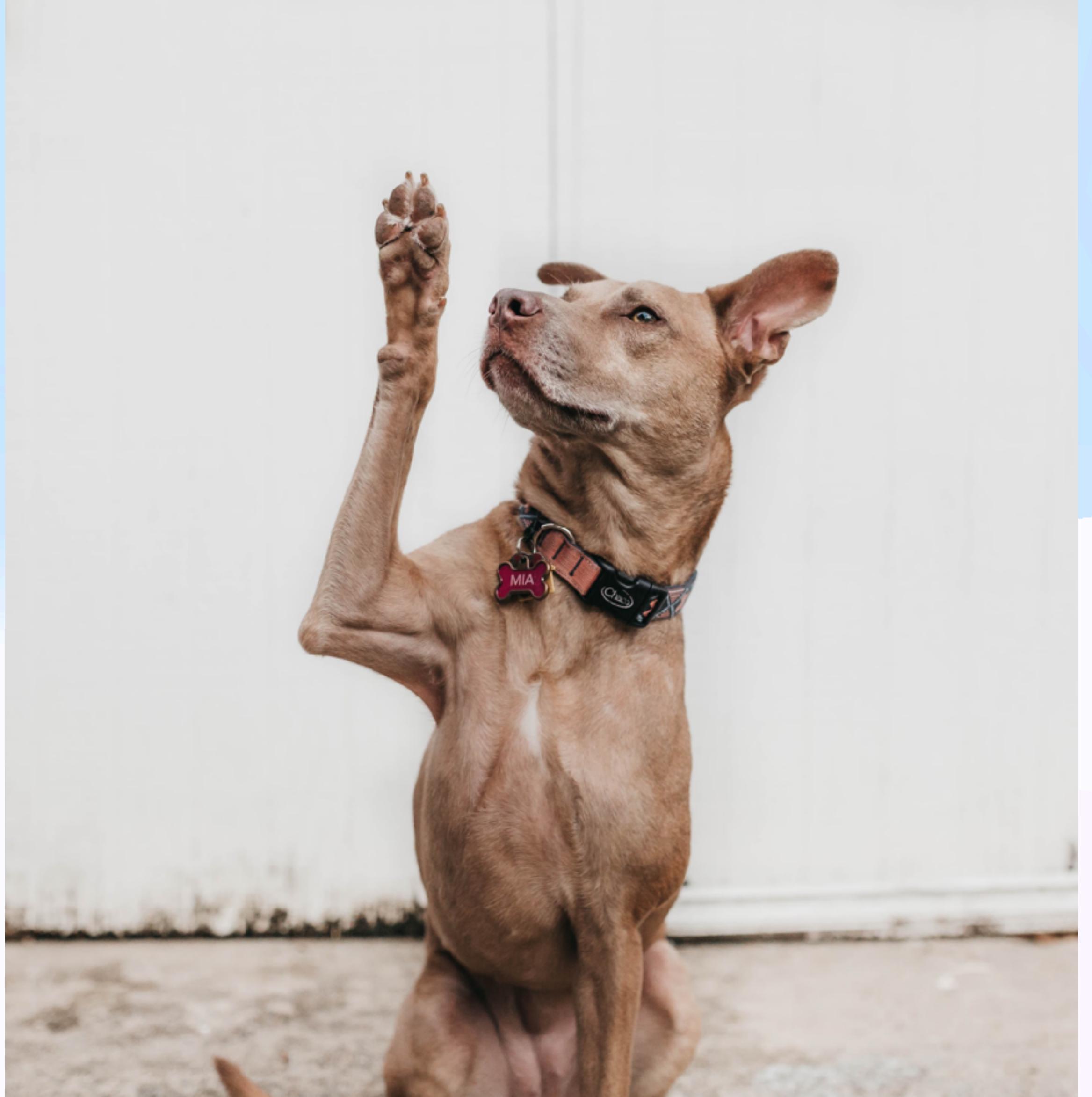
Learn how to ask question

**Great learning starts with honest
question**

- Start by sharing what you already know
- Ask—you don't need to be an expert to “Ask”
- Do some research before seeking help



I won't ask because
everyone else
probably know
the answer



**The only dumb question
is the one never asked**



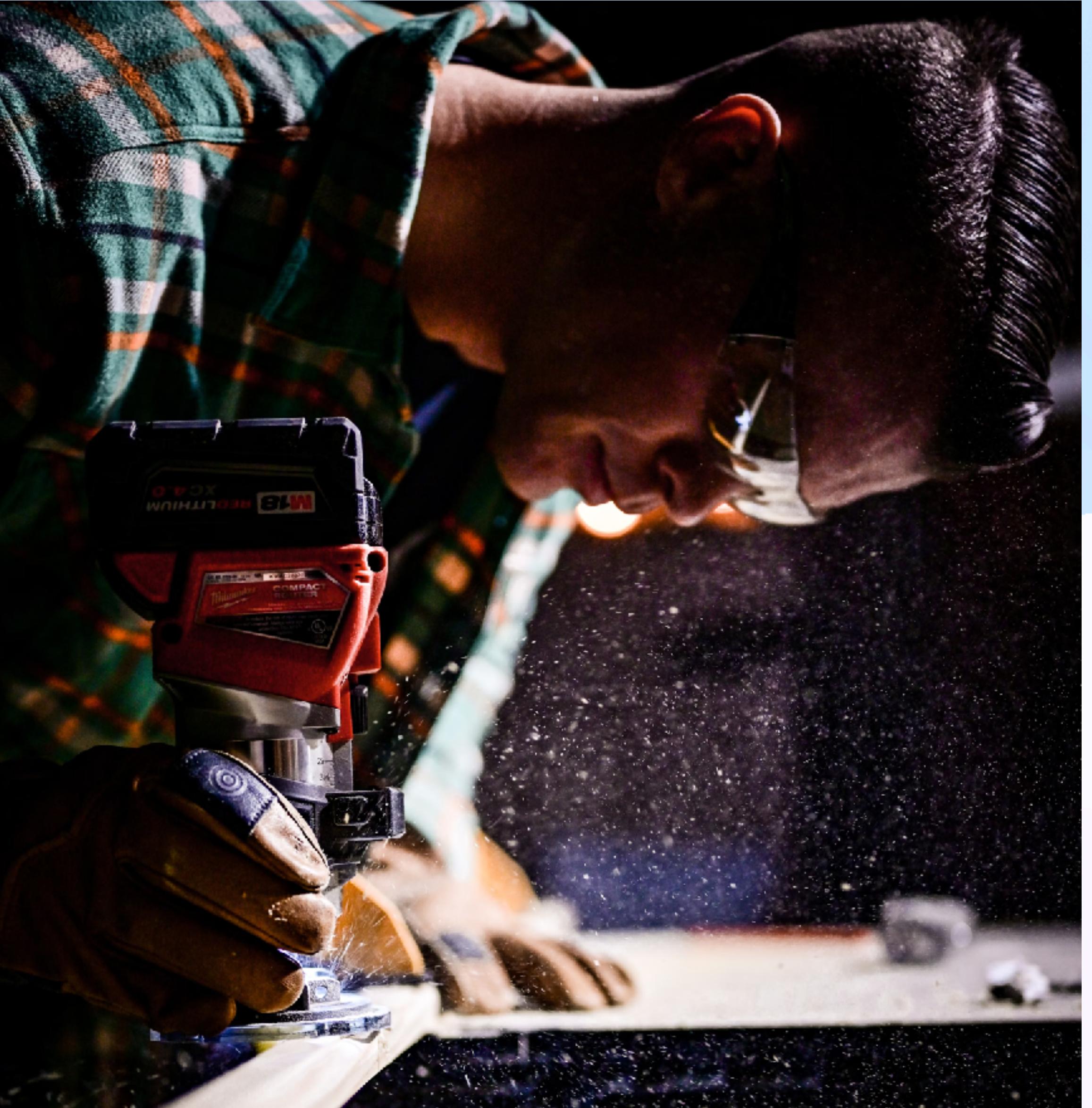
Dive deeper than the surface

understand the foundation behind the interface

- What under docker?
- How Garbage Collector work?
- How Postgres store data?
- Why Python slower than Go/Rust?



- Try to **build** your own container
- Try to **build** your own HTTP server
- Try to **build** your own database
- (or at least, try to google: “how to xxx from scratch”)

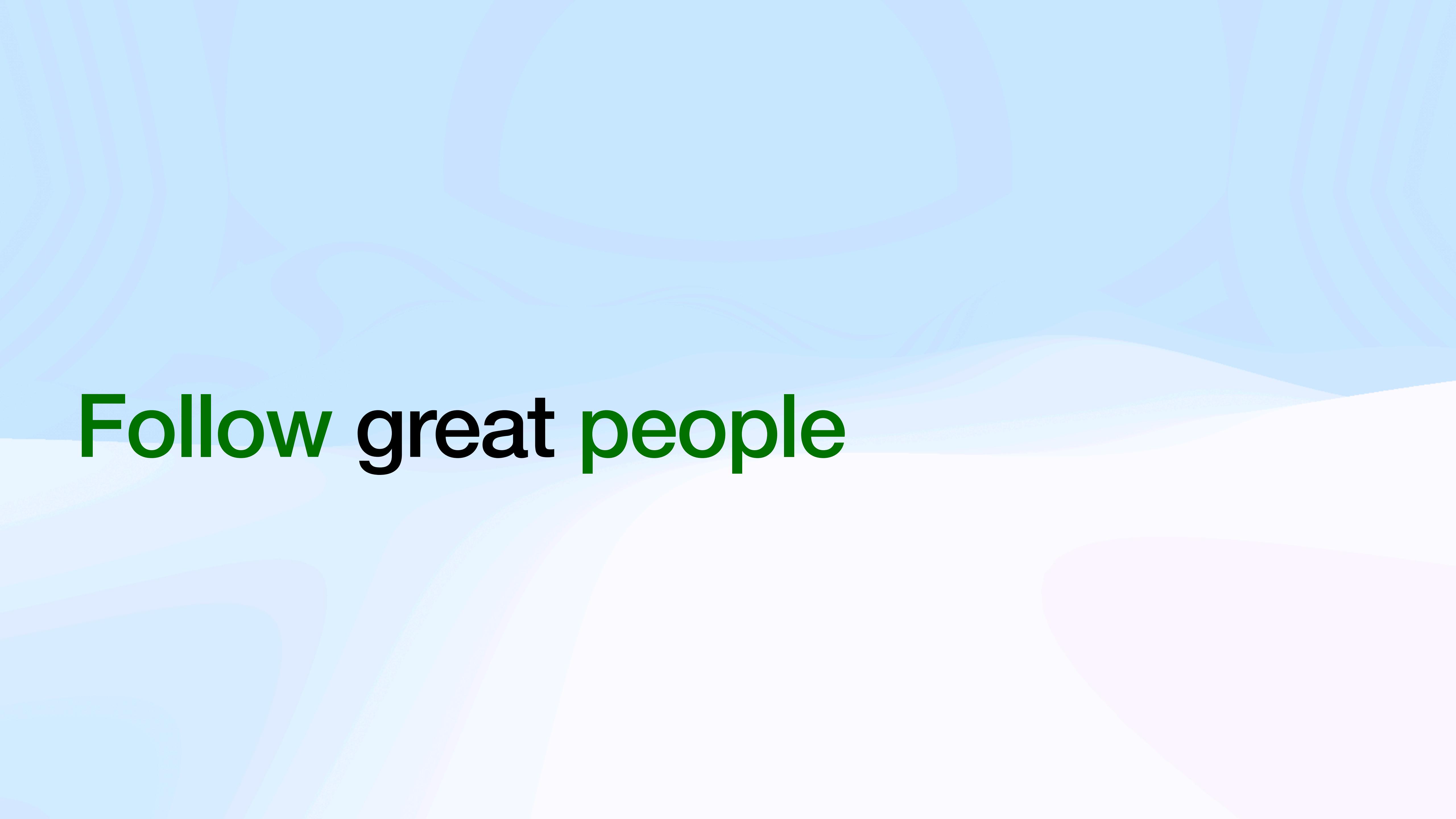




Join meetups Involve with Community

- Join as audience
- Join as speaker
- Join as organizer (easy and fun)

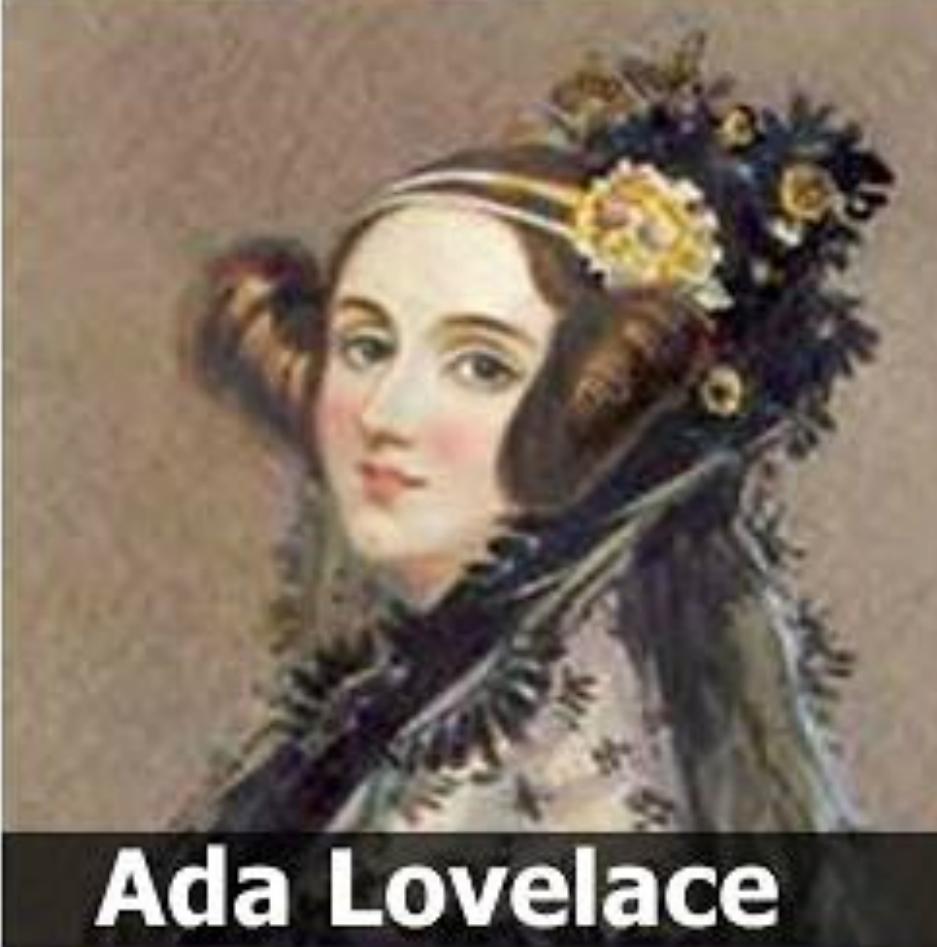




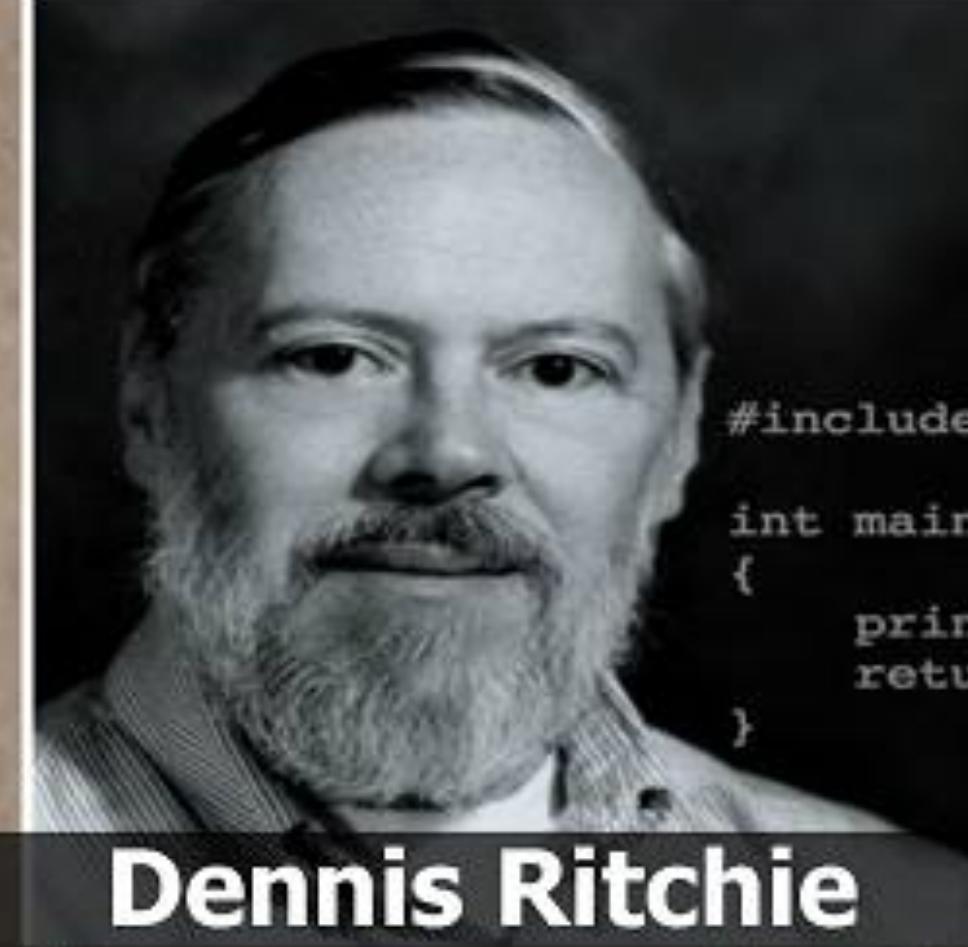
Follow great people

- เน้นคาดสมอง

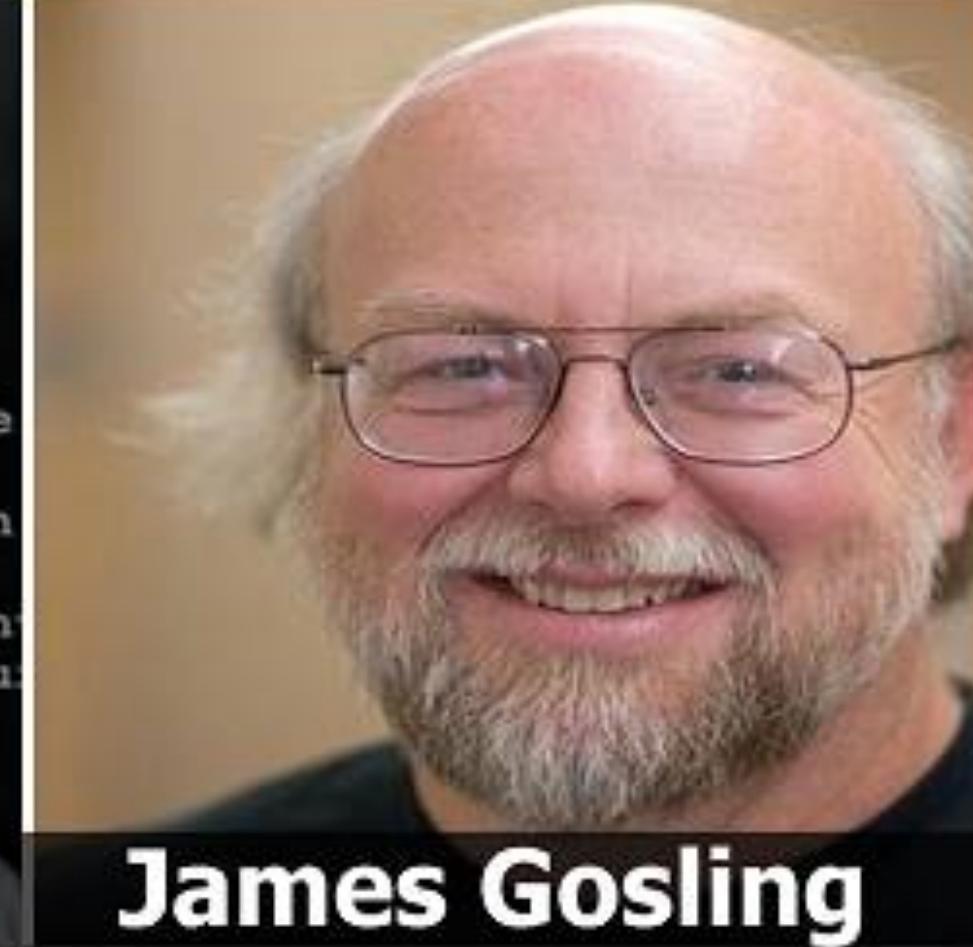




Ada Lovelace



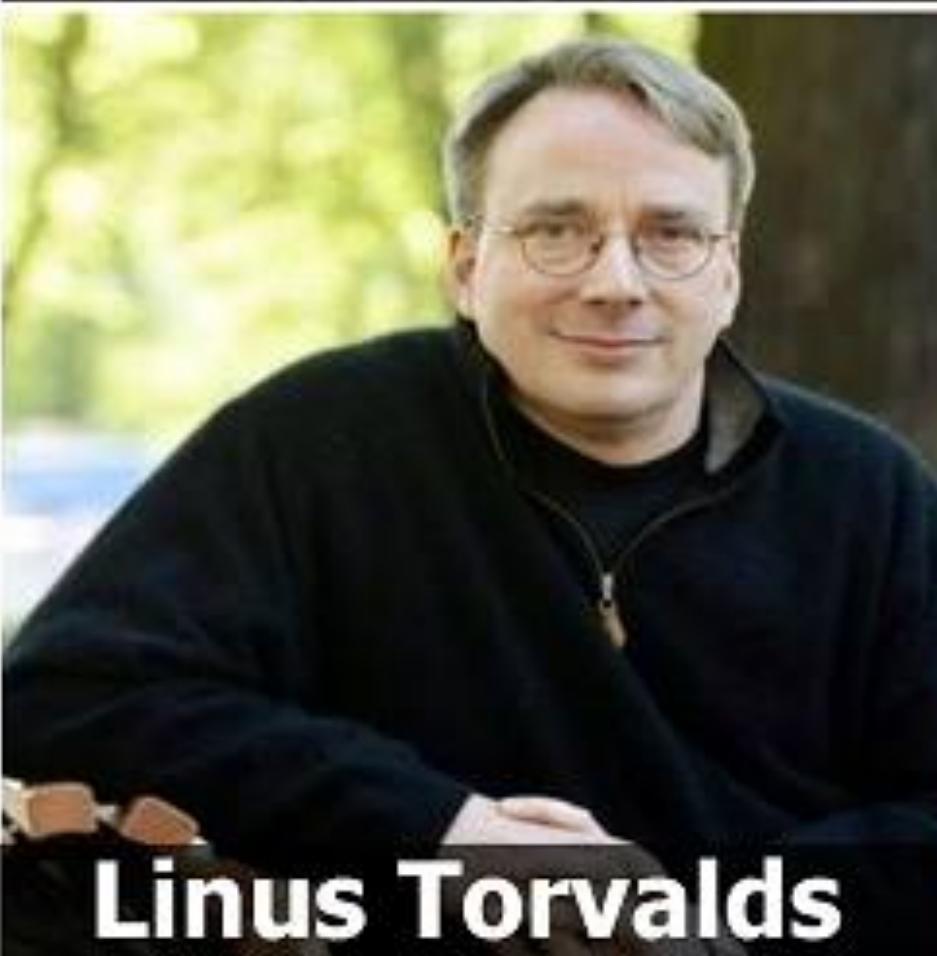
Dennis Ritchie



James Gosling



Guido van Rossum



Linus Torvalds



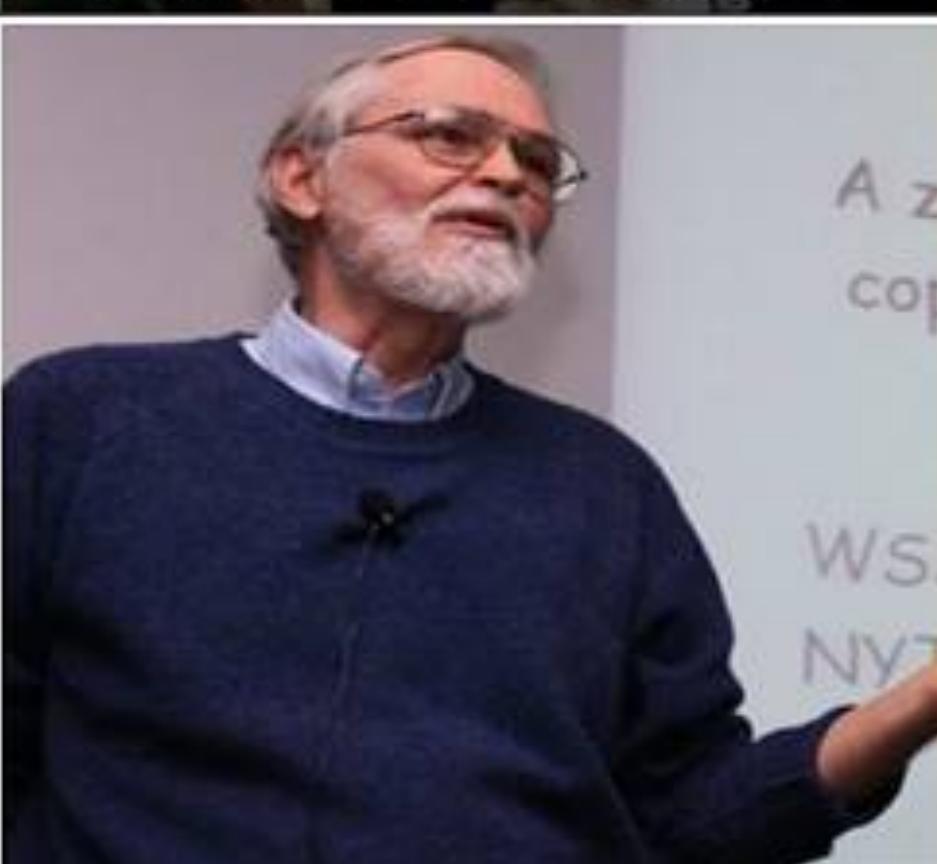
Bjarne Stroustrup



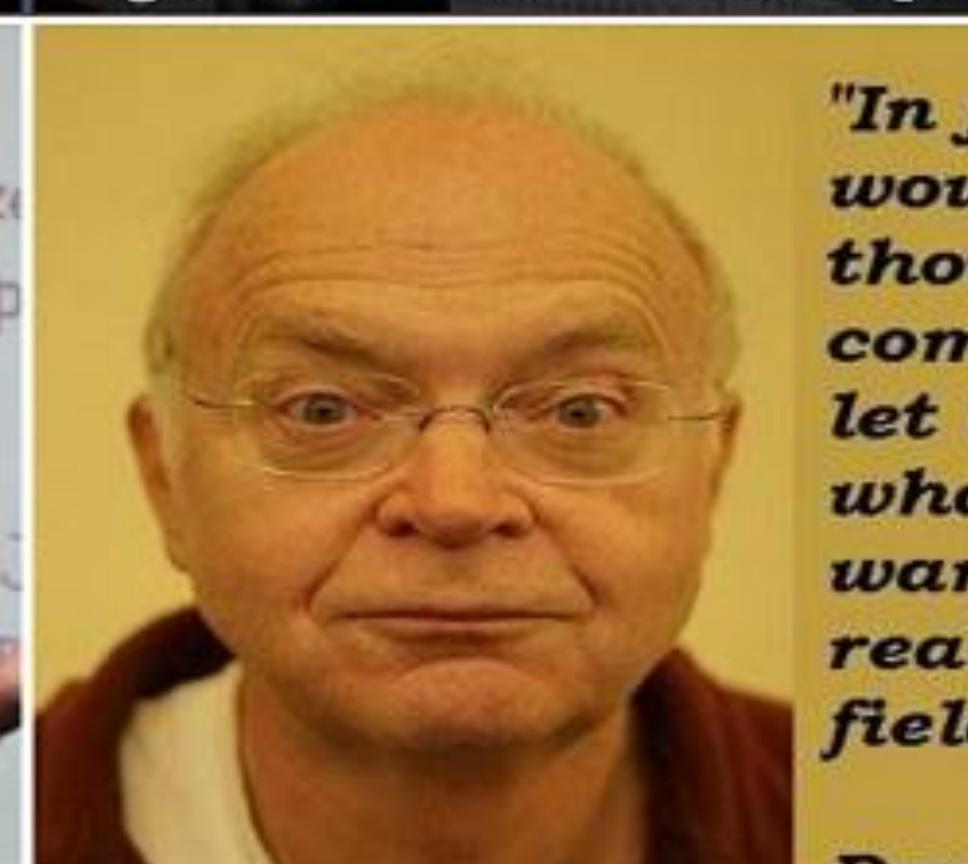
Ken Thompson



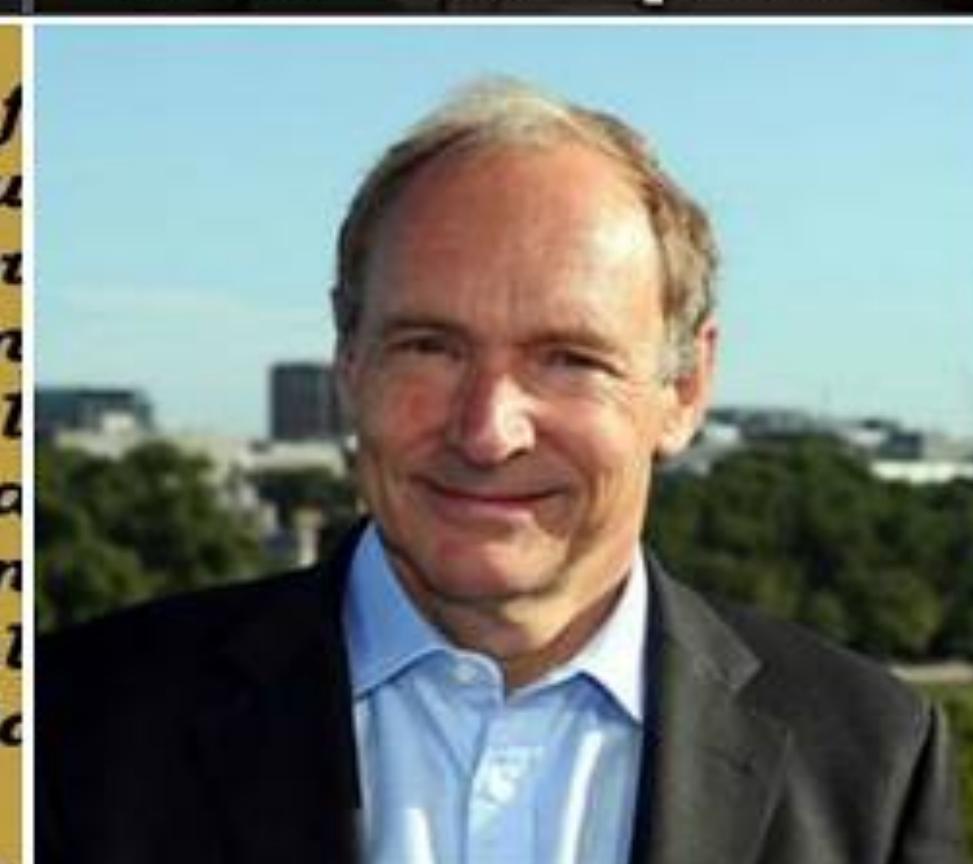
Niklaus Wirth



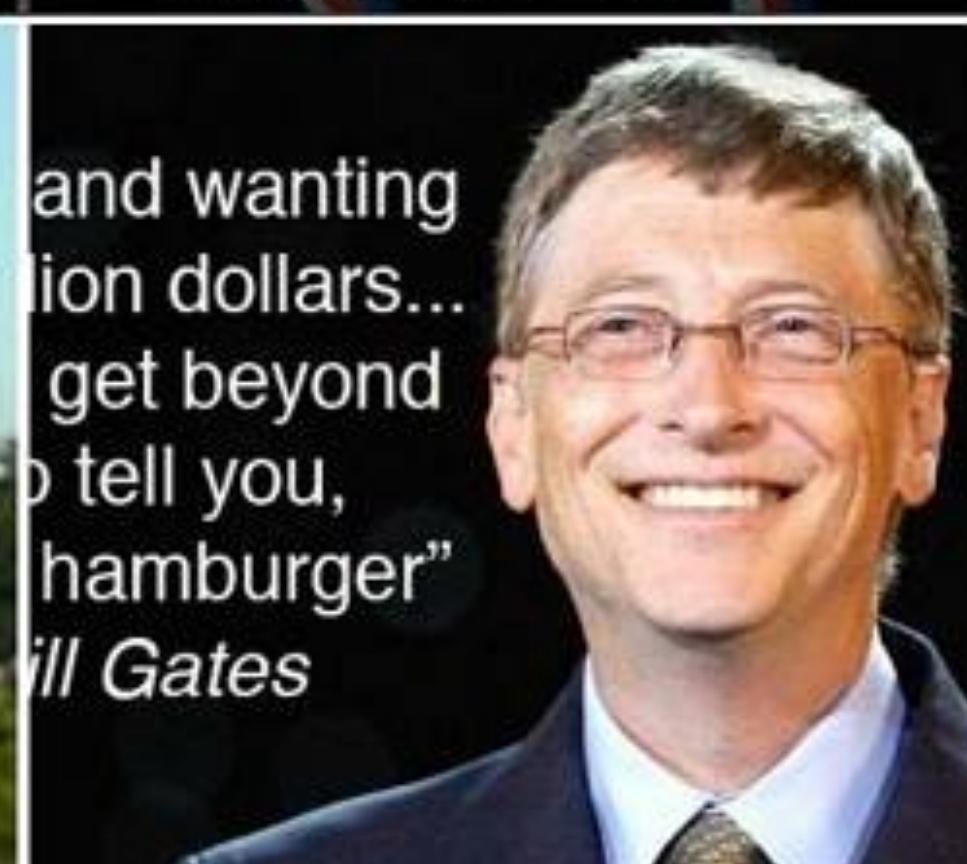
Brian Kernighan



Donald Knuth



Tim Berners-Lee



Bill Gates



Dennis Ritchie

Created
C



Bjarne Stroustrup

Created
C++



Anders Hejlsberg

Created
C#



James Gosling

Created
Java



Brendan Eich

Created
JavaScript



Rasmus Lerdorf

Created
PHP



Yukihiro Matsumoto

Created
Ruby



Guido Van Rossum

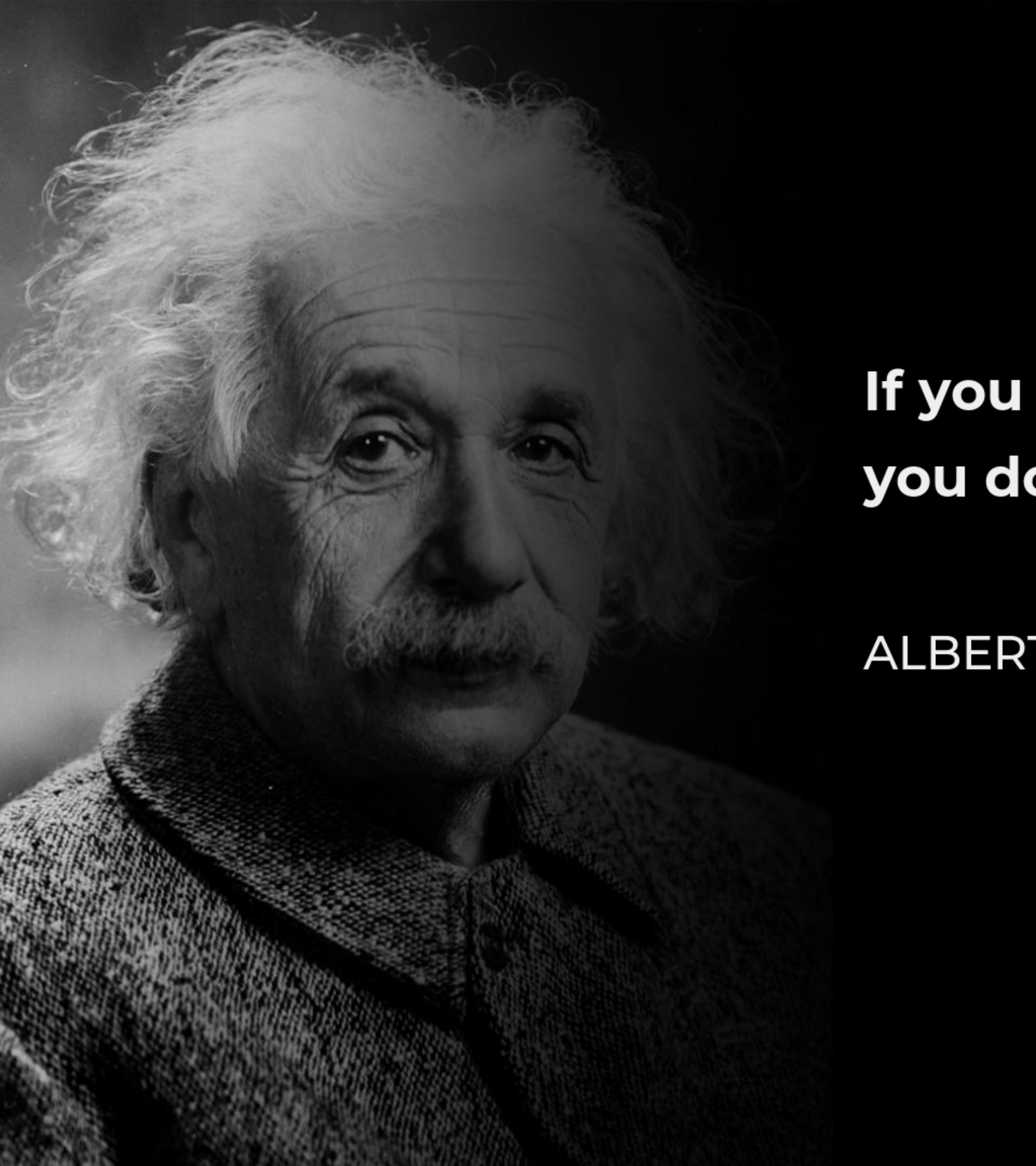
Created
Python



Larry Wall

Created
Perl

**Share what you learn
(blog / public talk / teaching)**

A black and white close-up portrait of Albert Einstein. He has his characteristic wild, grey hair and a full, bushy beard. His eyes are looking slightly off-camera with a thoughtful expression. The lighting is dramatic, casting deep shadows on one side of his face.

**If you can't explain it to a six year old,
you don't understand it yourself.**

ALBERT EINSTEIN

Writing is a
"MUST" skill

Please, No excuse



Learn hard thing

Get out of the comfort zone

Your **value** is in
what you can do
that others can't



You won't be paid
well if you can do
something that
anyone can learn in
a few months



**Hard is what feels
uncomfortable to do**

- If you're not good at **math**, take the time to **learn it**
- If **algorithms** are challenging, take the time to **learn it**
- It's **impossible** to learn everything, so focus on something that interests you—**start simple**

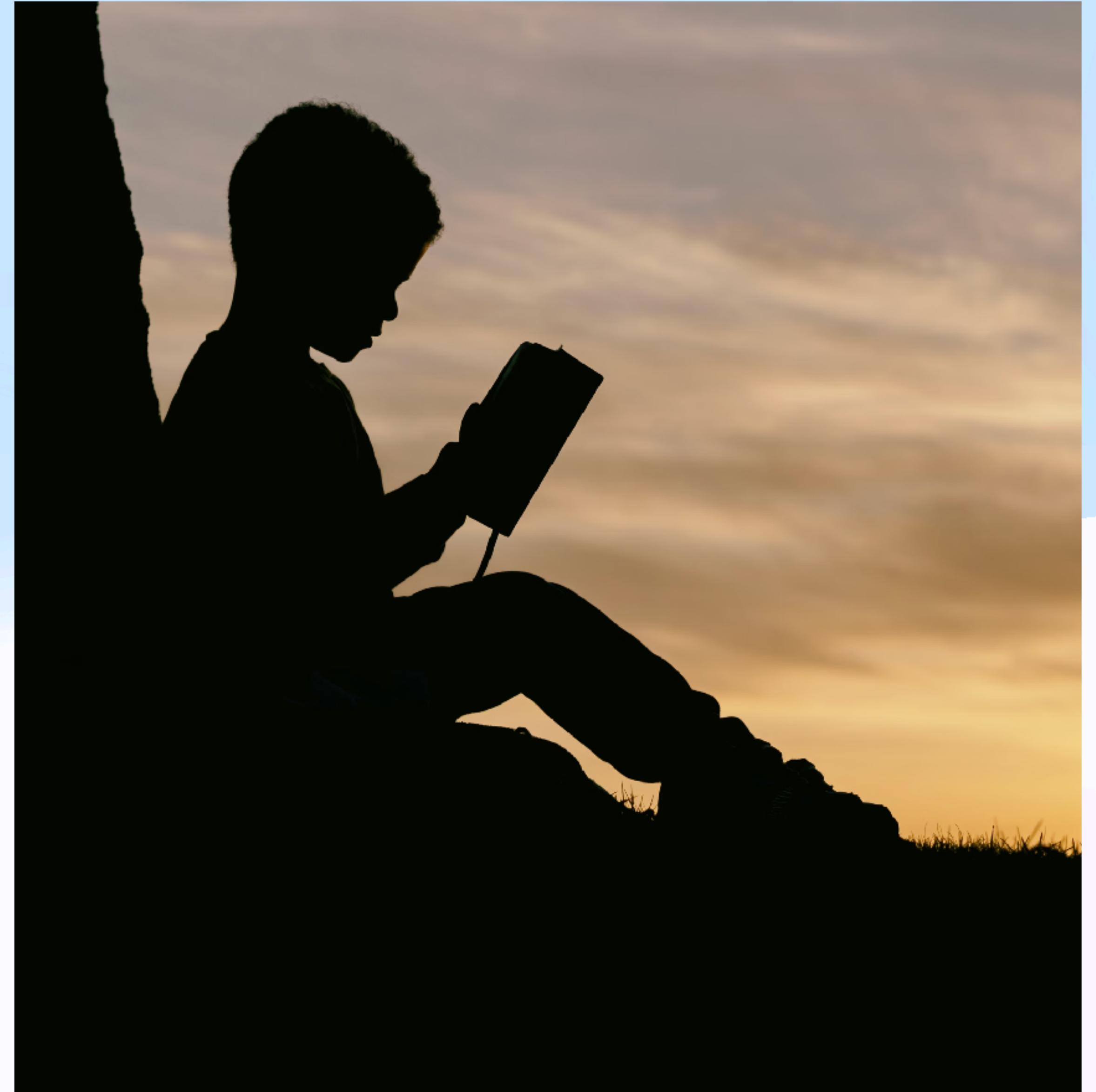


Don't underestimate others

Don't overestimate others

Respect vs Equality

**Being smart isn't
something you're
born with;
it's something you
can learn**





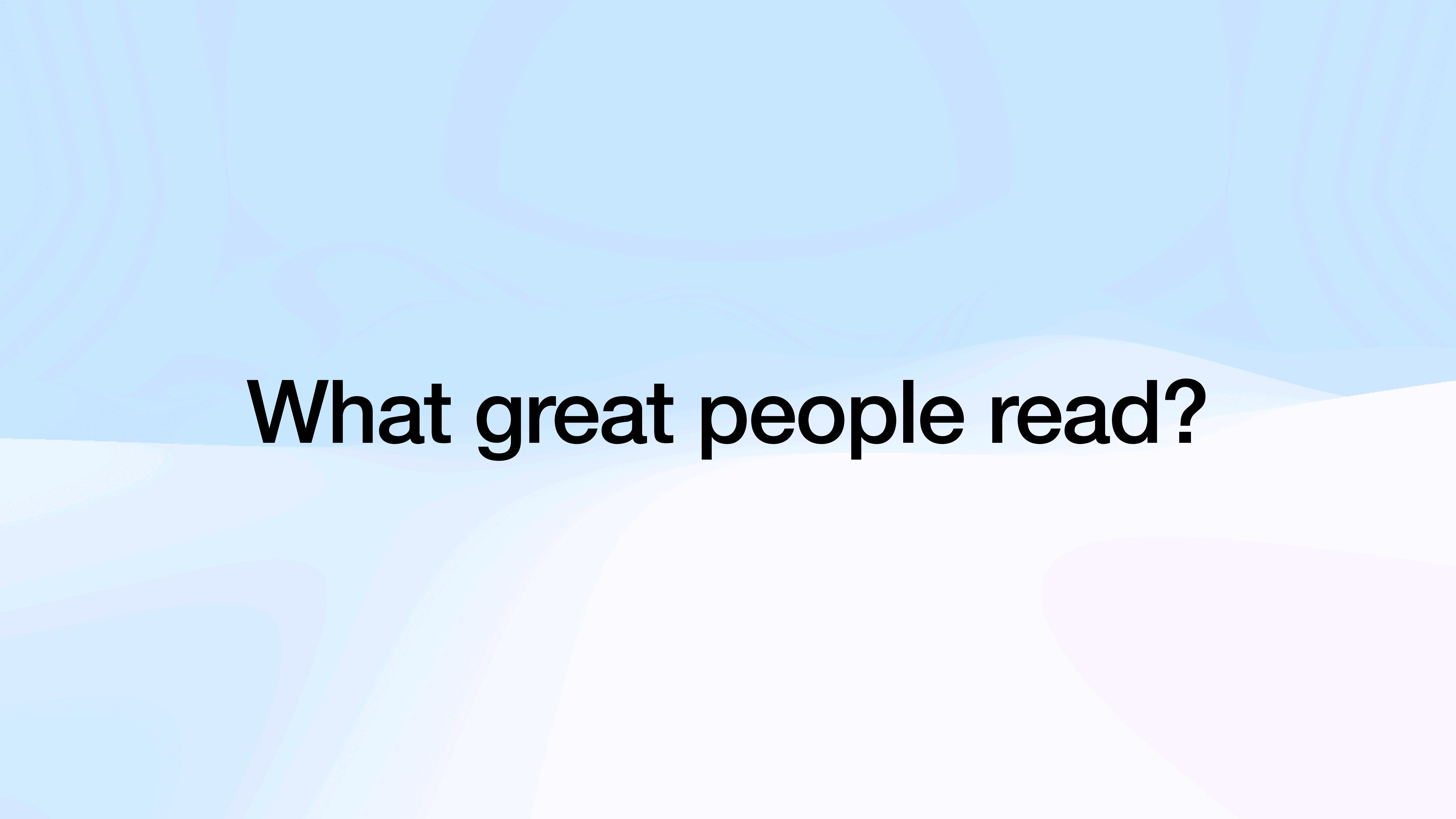


**Learn from books and research
papers, not just blogs or tutorials**

**Learn to read paper from top
conference
(ACM / IEEE)
(UseNix also a good source)**

Learn how to read books

**Great engineers read dozens of
great books every year
(technical + non-technical)**



What great people read?

**Follow great people in Goodreads,
see what they read in twitter**





Don't follow the hypes

Learn the basics first

AI/Machine Learning?

AI/Machine Learning?

No, Start by learning probability

Web programming / ReactJS?

Web programming / ReactJS?

No, learn how **HTTP** works
learn how **Database** works

Domain Driven Design?

Domain Driven Design?

No, understand clearly

Object Oriented first

**Every technology
exists for a reason,
know the reason**

Lastly



Problem-Solving: The Core of Engineering



We **not** just to get the **result**
we learn to **build** a learning skill

Say no to:

'XXX for beginners'

'Learn YYY in 24 hours'

Responsibility vs Passion

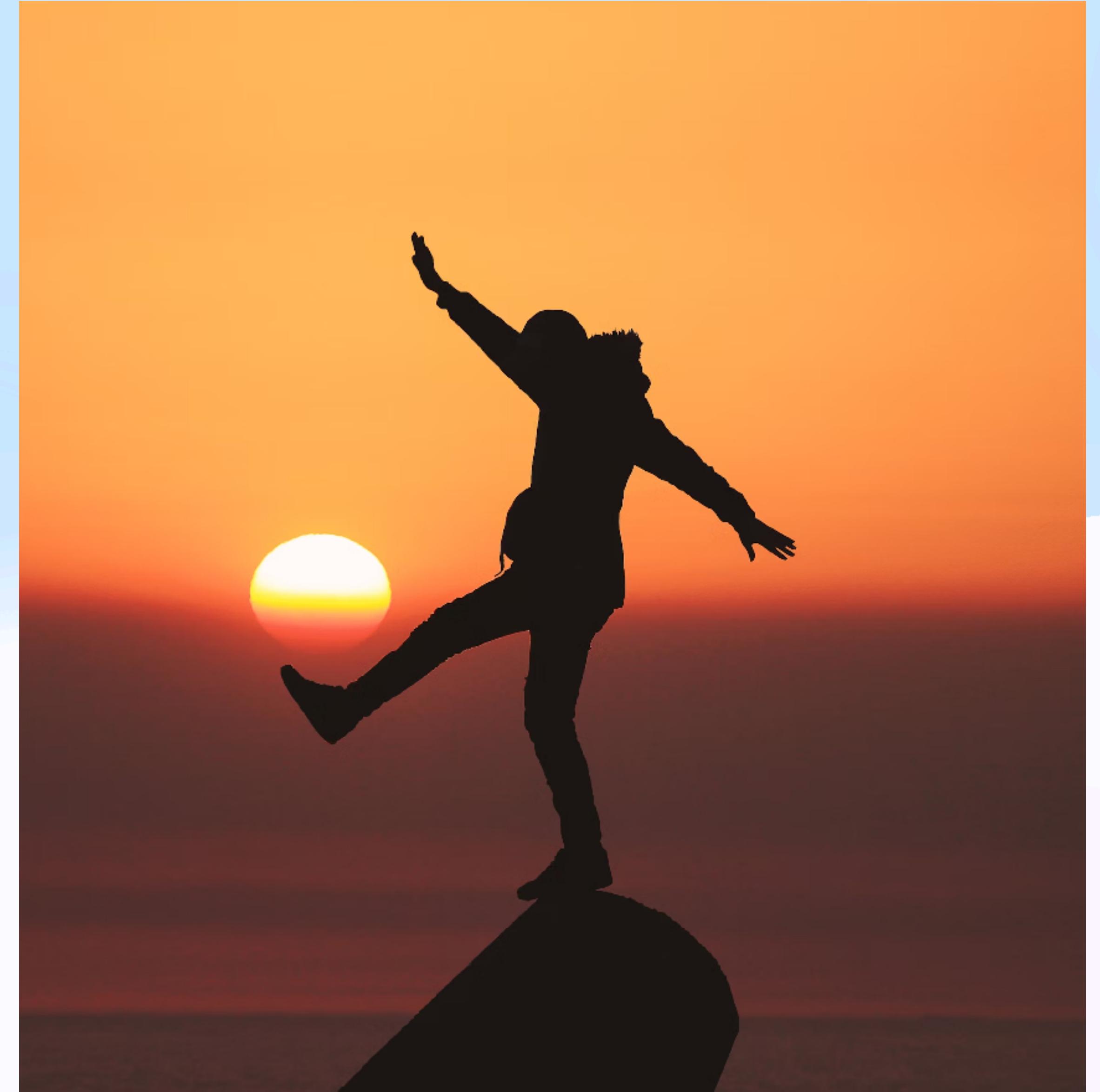
เรียนทำให้รามีงานทำ

กิจกรรมทำให้เราทำงานเป็น

You'll forget everything
I say today

**It's all about
being "better"**

**Am I better than
I was yesterday?**





Am I better than
I was yesterday?

