

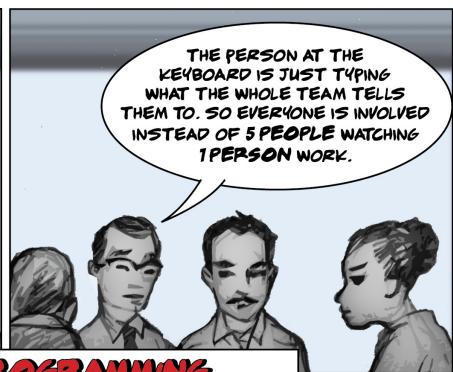


# Pair Programming

**" Mob Programming is all the brilliant people working on the same thing, at the same time, in the same space, on the same computer."**

-- Woody Zuill



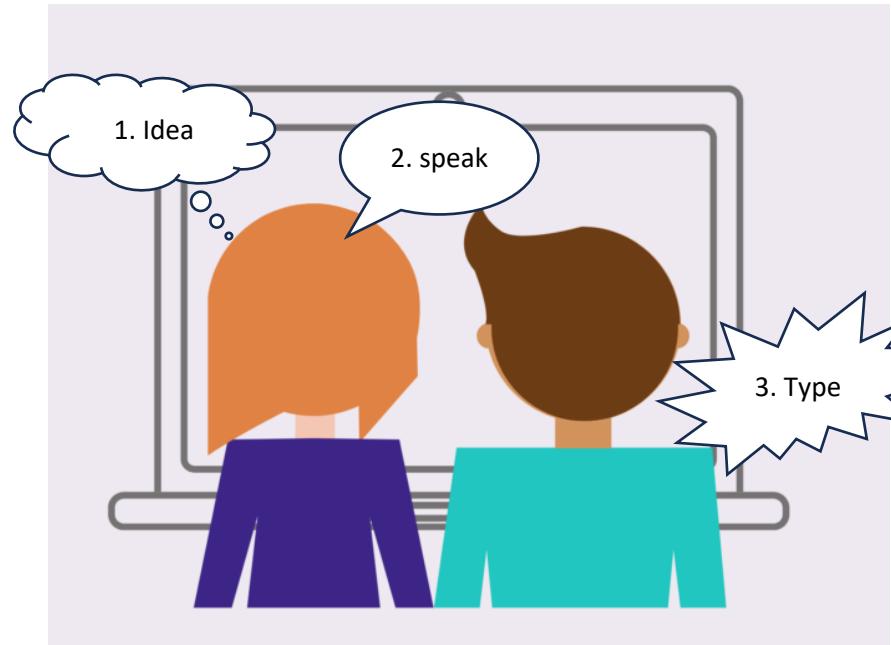


## THE MOB PROGRAMMING GUIDEBOOK

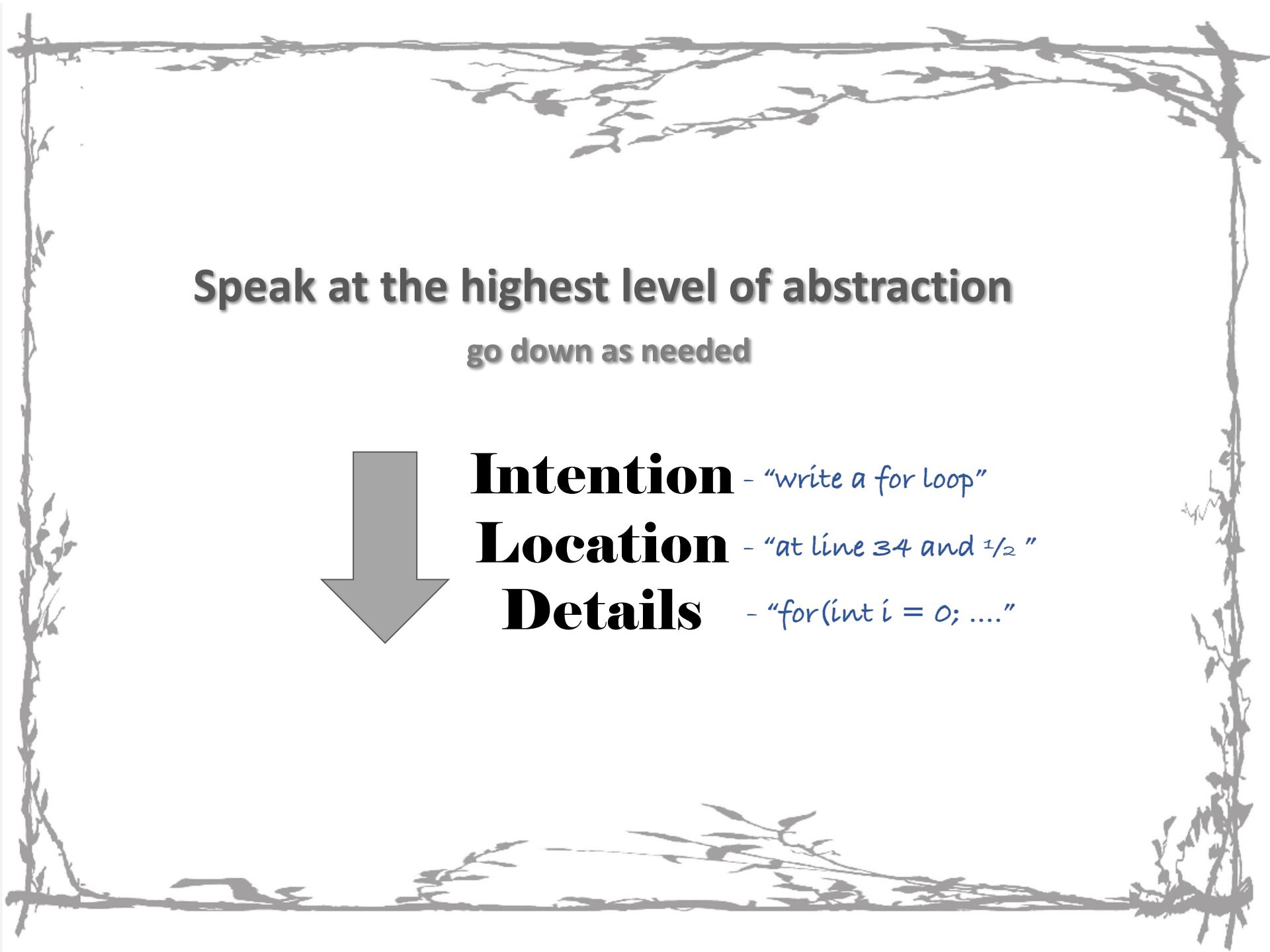
Llewellyn Falco & Maaret Pyhäjärvi



**"For an idea to go from  
your head to the computer  
it must go through  
someone else's hands"**

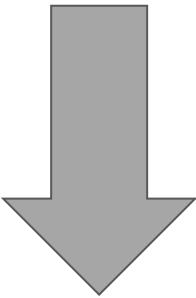


# Strong Style Pair Programming



# **Speak at the highest level of abstraction**

**go down as needed**



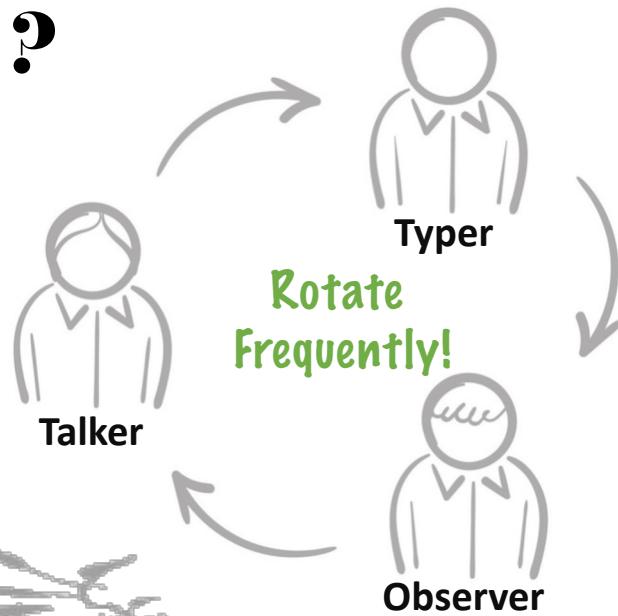
- Intention** - "write a for loop"
- Location** - "at line 34 and  $\frac{1}{2}$ "
- Details** - "for(int i = 0; ...."

# The 5 Whys



← Root Cause

# **When is the last time you rotated ?**



# Rotation Styles



**On Task**



**On Time**

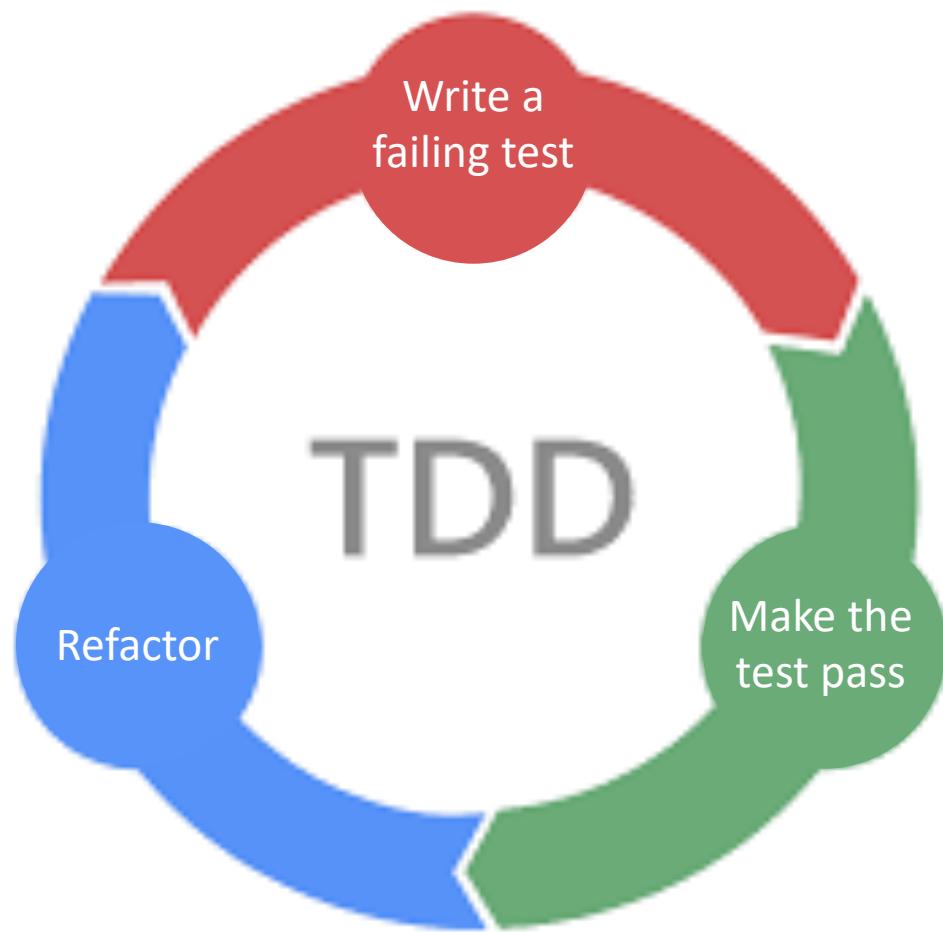


**On Idea**

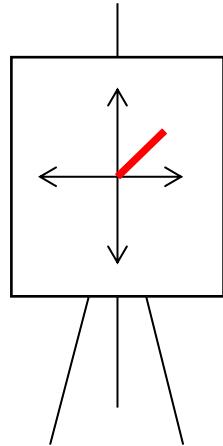


Kindness  
Consideration  
and  
Respect

**Working  
Agreements**  
*Make for better pairs & mobs*



**Start Here!**  
with **drawing**  
and natural **language**



Whiteboard

Testing Circle

// Create side (0,0) – (3,4)  
// Verify length

English

Result

Code

Side (0,0) – (3,4) length = 5

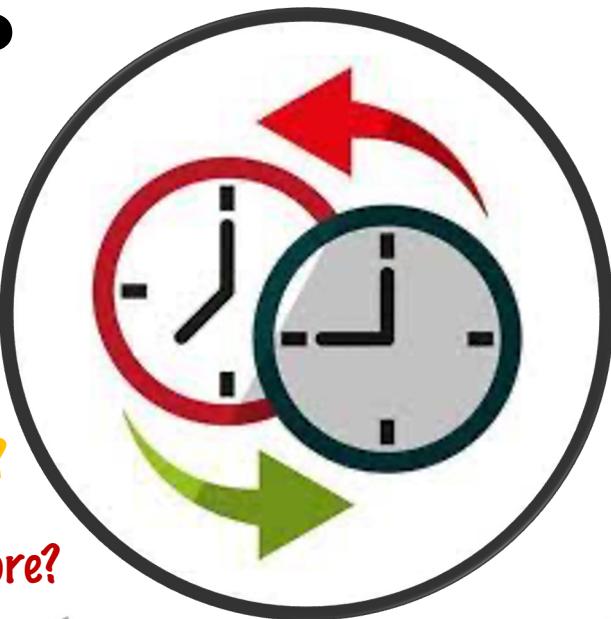
Side s = new Side(0,0,3,4);  
Approvals.Verify(s + " length = " +s.Length);

# **When was the last time you ran your tests?**

1 minute?

3 minutes?

more?



# Test & Commit || Revert (TCR)

ADVANCED  
TECHNIQUE

What it is

Every time you run your tests, you will either:

1. Commit everything, because **all tests passed**
2. Revert everything (or everything except the test) because **it failed**



Kent Beck  
Creator of TCR  
& Extreme Programming

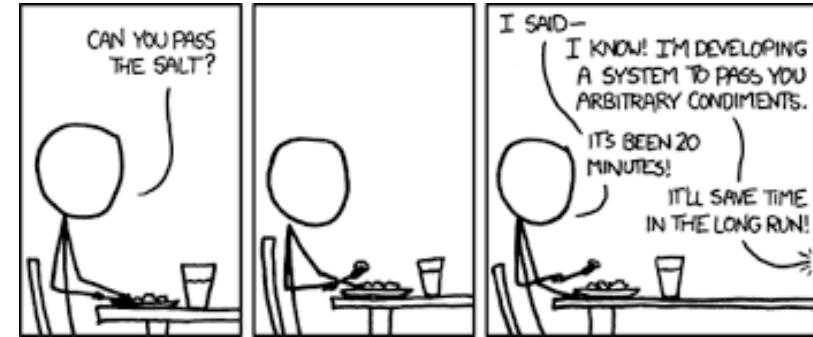
**Smaller steps**  
**Faster Feedback**  
**Better safety net**

Java starter project



[github.com/LarsEckart/tcr-extension.starterproject](https://github.com/LarsEckart/tcr-extension.starterproject)

You  
Ain't  
Gonna  
Need  
It



xkcd.com

*Write the simplest thing  
that could possibly work*

# **Z**ero

# **O**ne

# **M**any

# **B**oundaries

# **I**nterfaces

# **E**xceptions

# **S**cenarios

*Guide your tests by  
looking at these 7 cases*

Blog



[blog.wingman-sw.com/tdd-guided-by-zombies](http://blog.wingman-sw.com/tdd-guided-by-zombies)



Pattern by  
James Grenning

- 1. Specification** - what am I going to write?
- 2. Feedback** - did it work?
- 3. Regression** - does it still work?
- 4. Granularity** - why did it stop working?

Blog



## The 4 Benefits of Tests

Not like this....



1



2



3



4

Like this!



1



2



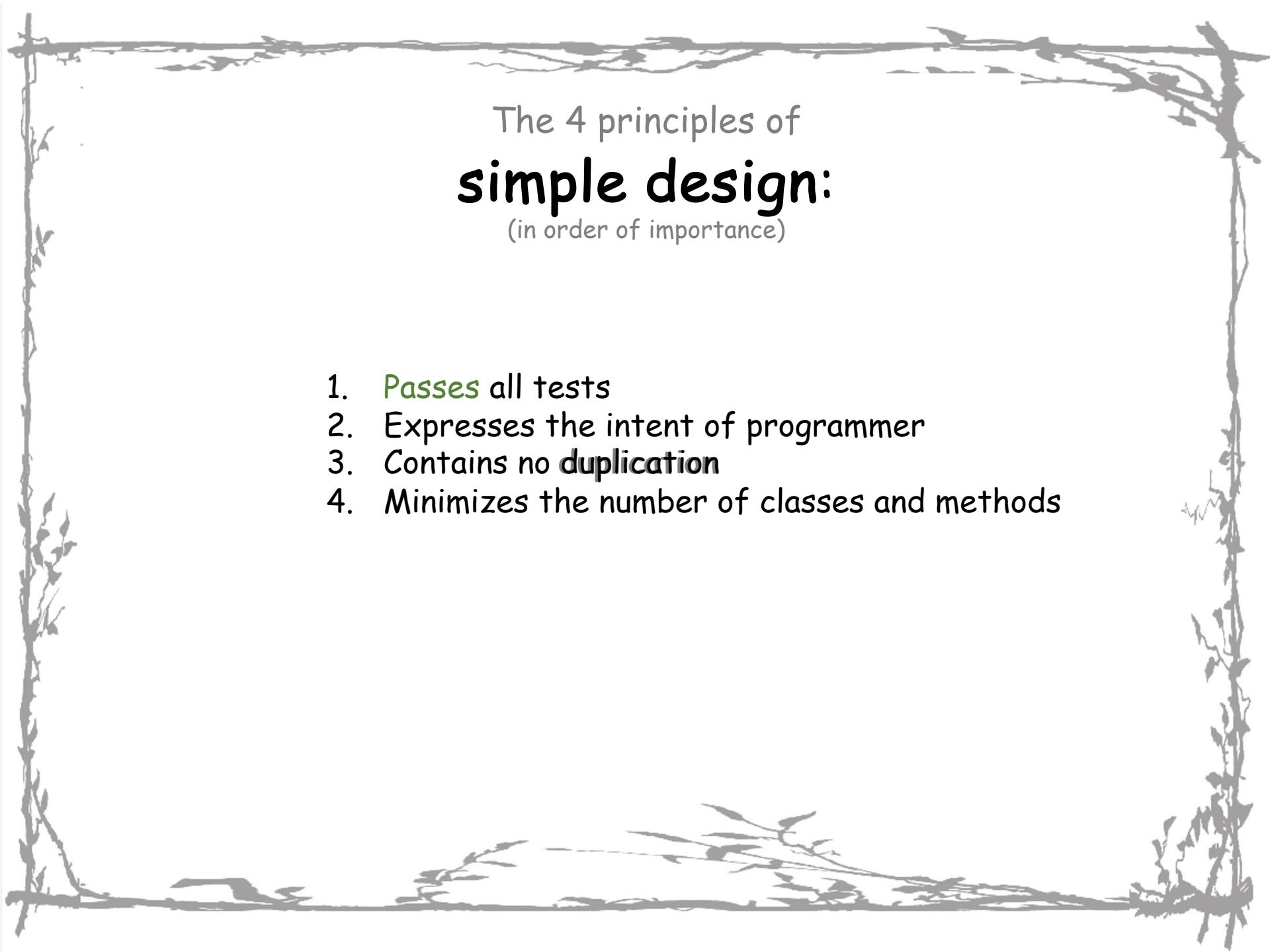
3



4



5

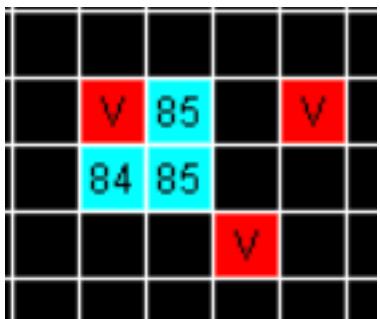


# The 4 principles of **simple design:** (in order of importance)

1. **Passes** all tests
2. Expresses the intent of programmer
3. Contains no **duplication**
4. Minimizes the number of classes and methods

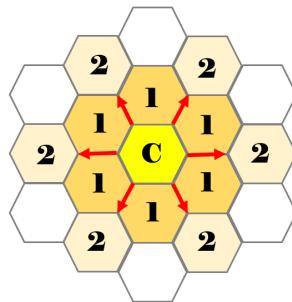
# Variations to Game of Life:

Vampires



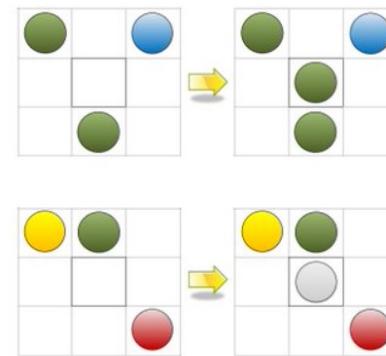
Cells age,  
turn into undying vampires,  
that kill each other

Hex



The board isn't square

Cells with Friends



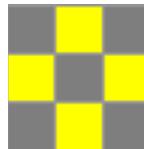
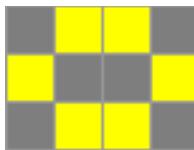
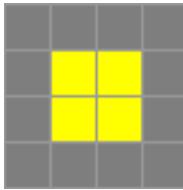
Cells have color  
which is passed onto their  
children

More info at:

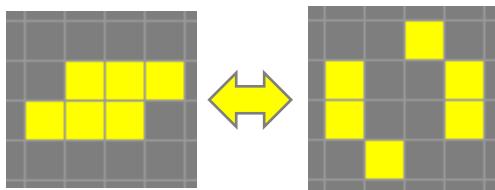
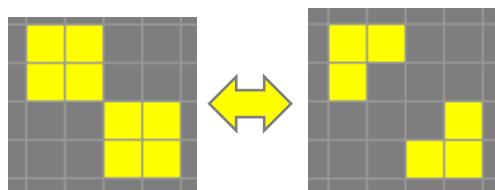
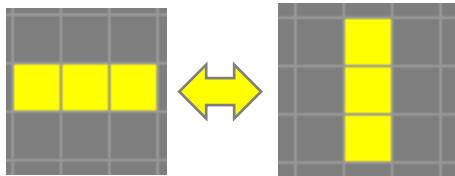


# Interesting Game of Life Structures

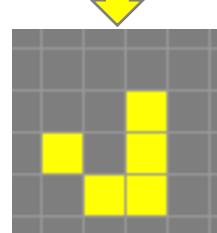
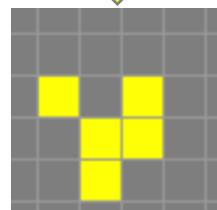
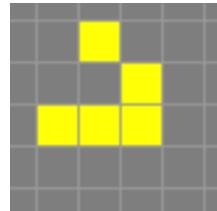
*Unchanging*



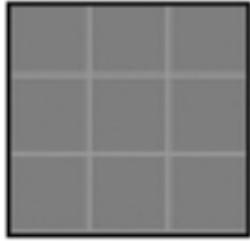
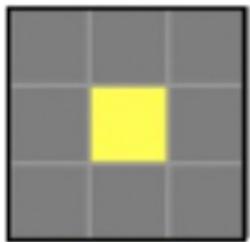
*Oscillating*



*Traveling*

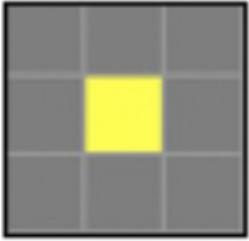
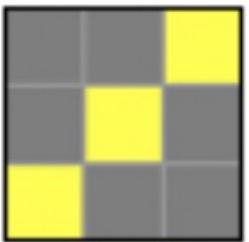


1 neighbor or less



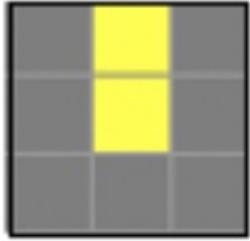
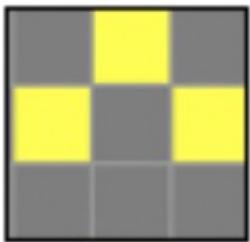
Middle square  
**dies**  
from **Starvation**

2-3 neighbors



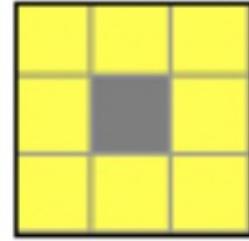
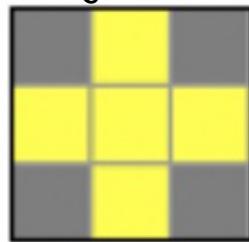
Middle square  
**survives**  
from **Subsistence**

3 neighbors



Middle square  
**is born**

4 or more  
neighbors



Middle square  
**dies**  
from **Overpopulation**

## Game of Life Rules

# Need an environment?

## CyberDojo

[cyber-dojo.org](http://cyber-dojo.org)

*Online coding and testing in many languages*

## Starter Projects

[github.com/LearnWithLlew/StarterProjects](https://github.com/LearnWithLlew/StarterProjects)

*Clone and go*

## exercism

[exercism.org](http://exercism.org)

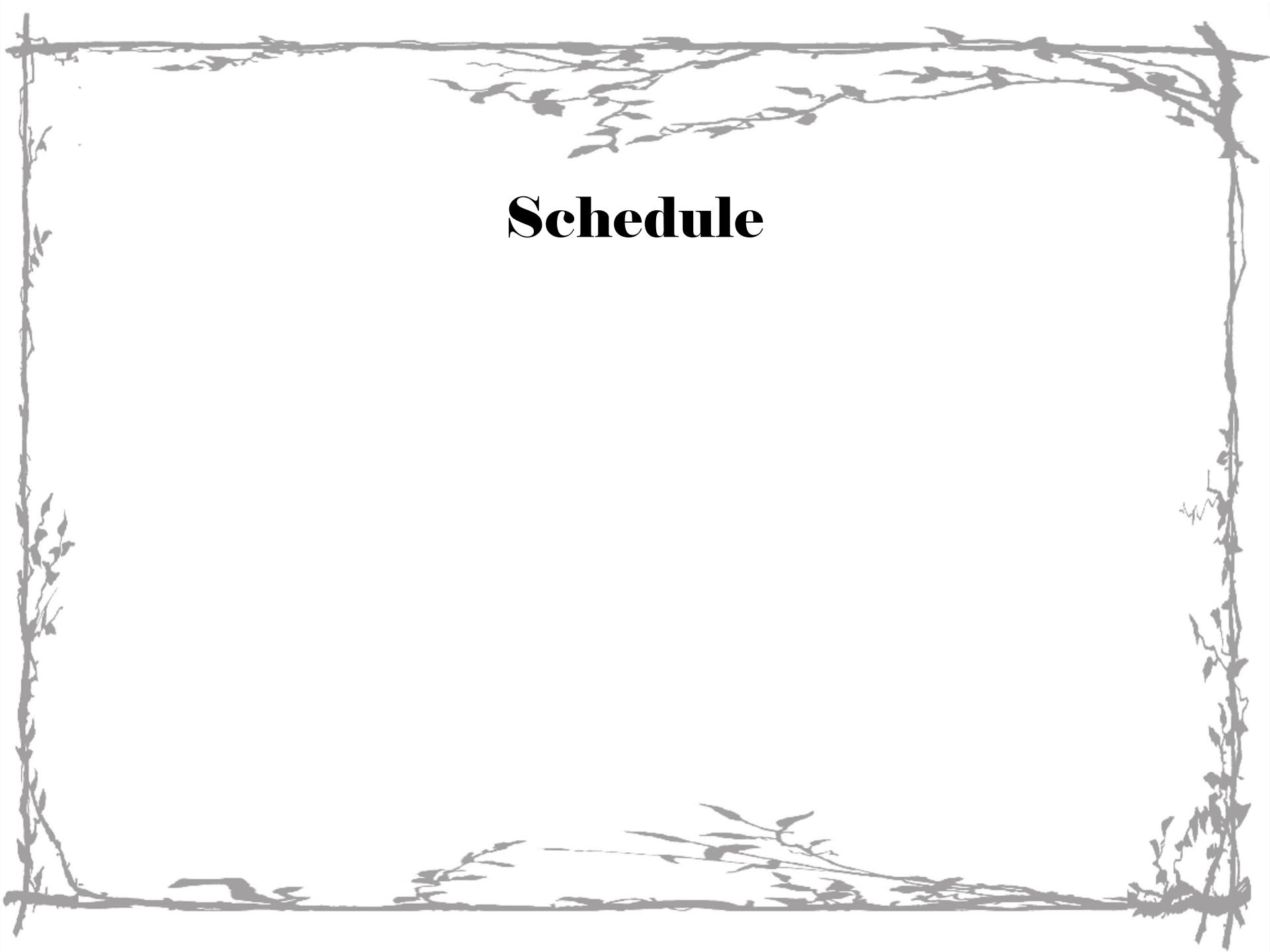
*64 languages with setup instructions*

## Gitpod.io

[gitpod.io/#<http://your\\_github\\_url>](https://gitpod.io/#<http://your_github_url>)

*VS Code online with linux  
Can install and run via the terminal*

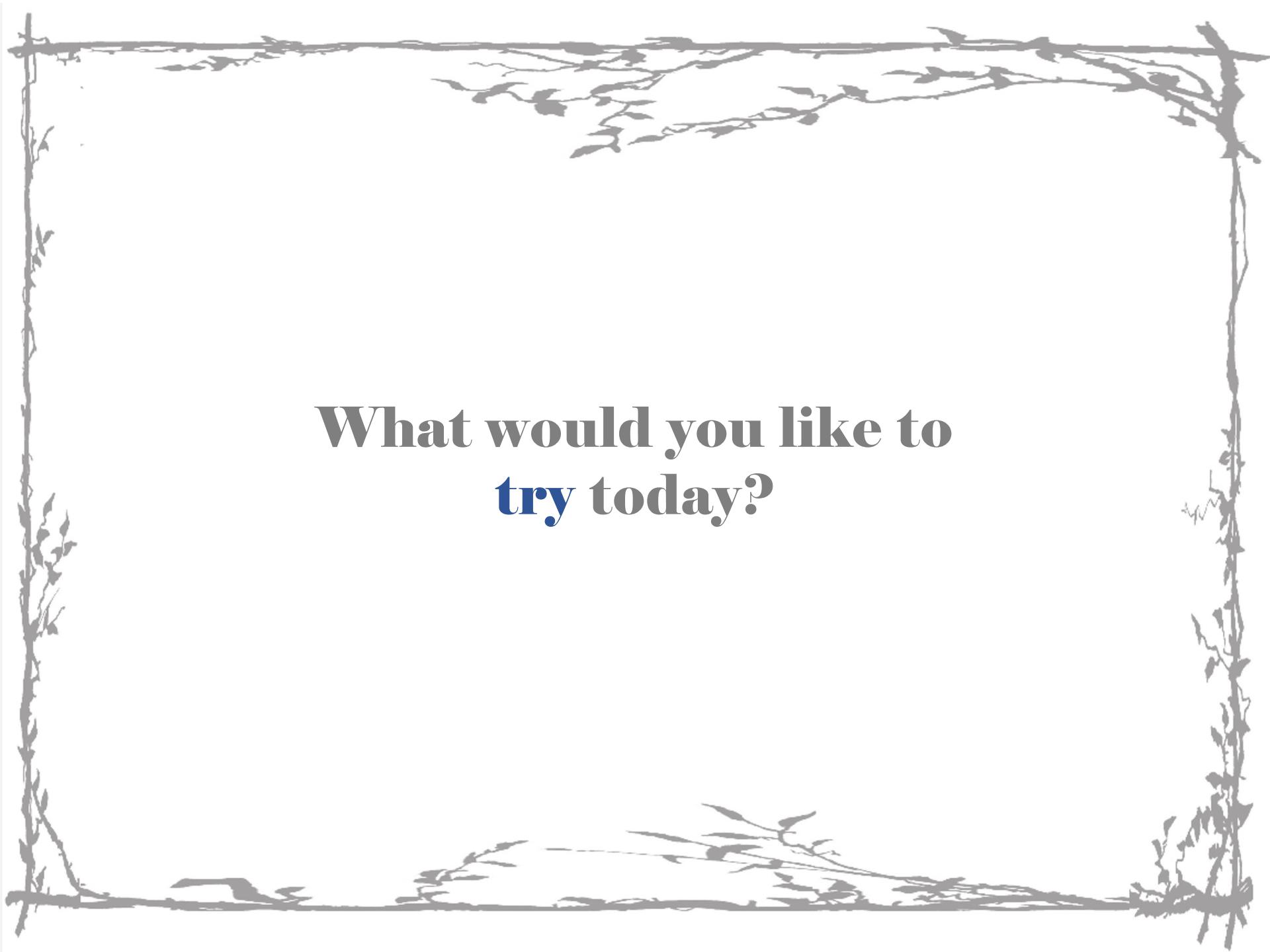




# Schedule



**What Surprised you  
today?**



**What would you like to  
try today?**

# What did you **learn** today?

I TRY NOT TO MAKE FUN OF PEOPLE FOR ADMITTING THEY DON'T KNOW THINGS.

BECAUSE FOR EACH THING "EVERYONE KNOWS" BY THE TIME THEY'RE ADULTS, EVERY DAY THERE ARE, ON AVERAGE, 10,000 PEOPLE IN THE US HEARING ABOUT IT FOR THE FIRST TIME.

FRACTION WHO HAVE = 0%  
HEARD OF IT AT BIRTH

FRACTION WHO HAVE ≈ 100%  
HEARD OF IT BY 30

US BIRTH RATE ≈ 4,000,000/year

NUMBER HEARING  
ABOUT IT FOR THE  
FIRST TIME ≈ 10,000/day

IF I MAKE FUN OF PEOPLE,  
I TRAIN THEM NOT TO TELL ME  
WHEN THEY HAVE THOSE MOMENTS.  
AND I MISS OUT ON THE FUN.

"DIET COKE AND MENTOS  
THING"? WHAT'S THAT?

OH MAN! COME ON, WE'RE  
GOING TO THE GROCERY STORE.  
WHY?

YOU'RE ONE OF  
TODAY'S LUCKY  
10,000.



*Many people have **strong opinions**  
about things they have **not experienced***

## Hate it with data



*Don't make up **your mind**  
about something without **trying it**.*

# Deliberate Practice

## THE FIVE PRINCIPLES OF DELIBERATE PRACTICE

