

Alcor course explained with CSS advertisements images

Malters, 18.04.2022

Florian Thalmann



My goal of today



Recap of the past weeks



Use power of images in
CSS advertisement



Test if it worked 😊

GIVE ME
EDUCATION!





ALCOR
academy



Walking

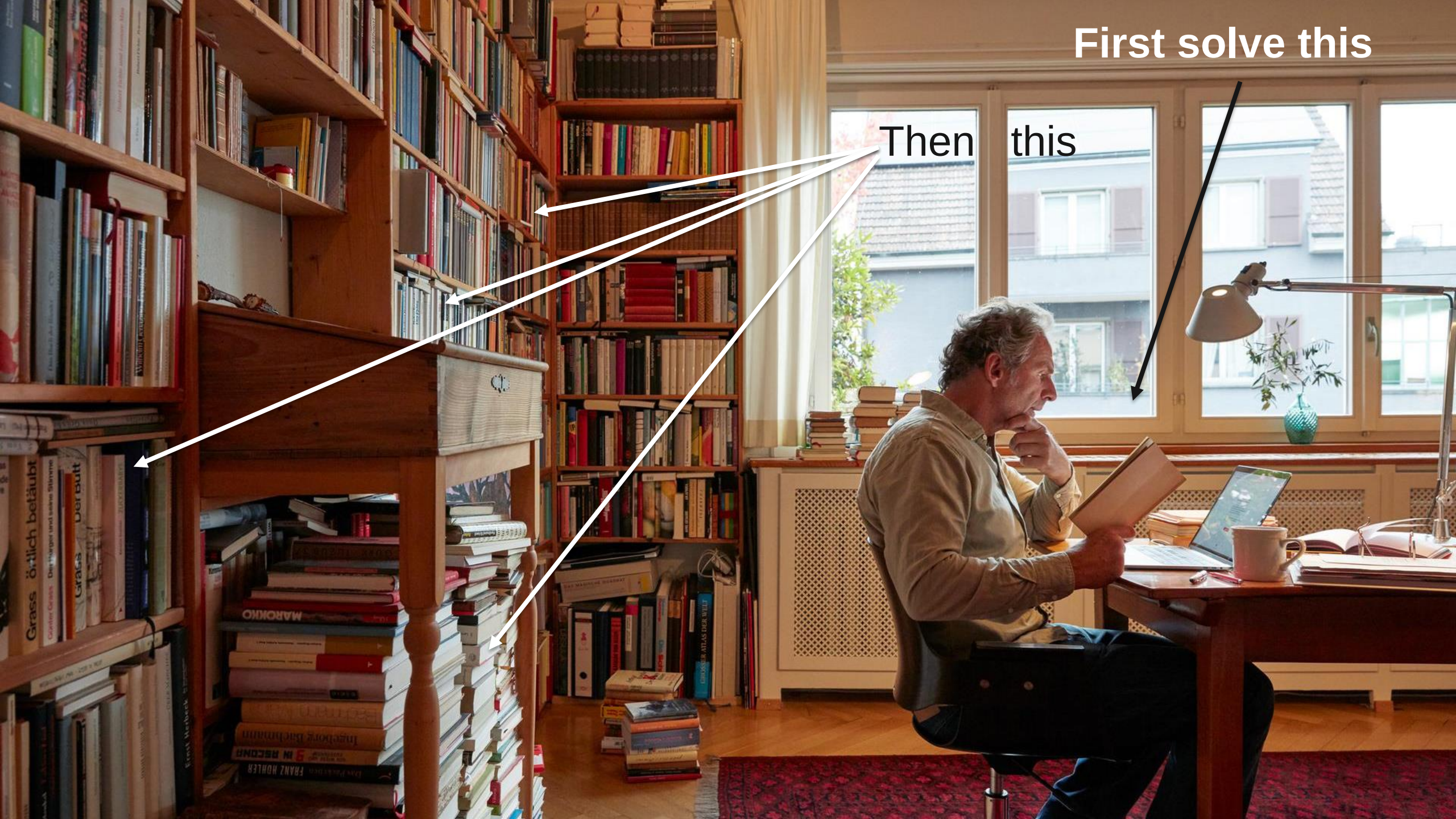


Fake implementation

Obvious implementation

Triangulation with the next test





First solve this

Then this



Object
Calisthenics
rules?! WTF!

Transformation
Priority Premise??
HELP!

Transformation Priority Premise - What is "Obvious implementation" ?

```
#  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
  
() => nil  
nil => constant  
constant => constant+  
constant => scalar  
statement => statements  
unconditional => conditional  
scalar => array  
array => container  
statement => loop  
conditional => tail recursion  
recursion => function  
expression => mutation  
variable => switch case
```

STARTING CODE

```
return nil  
return "1"  
return "1" + "2"  
return argument  
return arguments  
dog  
[dog, cat]  
a + b  
if(condition)  
a + recursion  
today - birthday  
day
```

FINAL CODE

```
return nil  
return "1"  
return "1" + "2"  
return argument  
if(condition) return arguments  
(dog, cat)  
(dog = "DOG", cat = "CAT")  
a + recursion  
while(condition)  
recursion  
CalculateAge()  
var day = 10; day = 11;
```

Object Calisthenics rules

1. Only one level of indentation per method
2. Don't use the ELSE keyword
3. Wrap all primitives and strings
4. First class collections (wrap all collections)
5. Only one dot per line `dog.Body.Tail.Wag()` => `dog.ExpressHappiness()`
6. No abbreviations
7. Keep all entities small
[10 files per package, 50 lines per class, 5 lines per method, 2 arguments per method]
8. No classes with more than two instance variables
9. No public getters/setters/properties
10. All classes must have state



<https://media.istockphoto.com/vectors/karate-kong-fu-flying-kick-man-silhouette-vector-id1056239976?k=20&m=1056239976&s=170667a&w=0&h=9N09rMSpA1KrW5WMT0ZOHS1LmHWaTIIIBT1JvUXHBMM=>

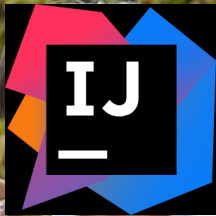


A man in a light blue shirt and dark pants is working on a modern house. He is standing on a stone wall, possibly a retaining wall or a foundation, and is using a red power tool. The house has a mix of wood siding and stone. There are some wooden sawhorses and other construction materials nearby. The background shows some greenery and a clear sky.

IDE agility: know your shortcuts

The 5 main atomic refactors of your IDE

- **Rename** – Change the name of classes, methods, variables....
- **Extract** – Extract a class (or methods or variables...), creating a new abstraction.
- **Inline** – The inverse of extract – inline a method (or variable), deconstructing an abstraction.
- **Move** – Move a class (or methods or variables...) to some other place in the codebase.
- **Safe delete** – Delete code and its usages in the code base.



Code smells



BLOATERS

COUPLERS

DISPENSABLES

OO Abusers

CHANGE
PREVENTERS

Code smells

CHEATSHEET

Switch Statements OOA Replace Conditional with Polymorphism Replace Type Code with Subclasses Replace Type Code with State/Strategy Move Accumulation to Visitor Replace Conditional Dispatcher with Command Replace Parameter with Explicit Methods Introduce Null Object	Inappropriate Intimacy COU Move Method Move Field Extract Class Hide Delegate Replace Inheritance with Delegation	Large Class BLO Extract Class Extract Subclass Extract Interface Replace Data Value with Object Replace Conditional Dispatcher with Command Replace Implicit Language with Interpreter Replace State-Altering Conditionals with State
Primitive Obsession BLO Replace Data Value with Object Encapsulate Composite with Builder Introduce Parameter Object Extract Class Move Embellishment to Decorator Replace Conditional Logic with Strategy Replace Implicit Language with Interpreter Replace Implicit Tree with Composite Replace State-Altering Conditionals with State Replace Type Code with Class Replace Type Code with State/Strategy Replace Type Code with Subclasses Replace Array With Object	Duplicated Code DIS Chain Constructors Extract Composite Extract Method Extract Class Form Template Method Introduce Null Object Factory Method Pull Up Method Pull Up Field Substitute Algorithm Adapter	Long Method BLO Extract Method Compose Method Introduce Parameter Object Move Accumulation to Collecting Parameter Move Accumulation to Visitor Decompose Conditional Preserve Whole Object Replace Conditional Dispatcher with Command Replace Conditional Logic with Strategy Replace Method with Method Object Replace Temp with Query
Divergent Change CHP Extract Class	Shotgun Surgery CHP Move Method Move Field Inline Class	Feature Envy COU Extract Method Move Method Move Field
Long Parameter List BLO Replace Parameter with Method Introduce Parameter Object Preserve Whole Object	Data Clumps BLO Extract Class Preserve Whole Object Introduce Parameter Object	Parallel Inheritance Hierarchies CHP Move Method Move Field
Middle Man COU Remove Middle Man Inline Method Replace Delegation with Inheritance	Data Class DIS Move Method Encapsulate Field Encapsulate Collection	Message Chains COU Hide Delegate Extract Method Move Method
Speculative Generality DIS Collapse Hierarchy Rename Method Remove Parameter Inline Class	Temporary Field OOA Extract Class Introduce Null Object	Lazy Class DIS Collapse Hierarchy Inline Class
Refused Bequest OOA Push Down Field Push Down Method Replace Inheritance with Delegation	Alternative Classes with Different Interfaces OOA Unify Interfaces with Adapter Rename Method Move Method	Incomplete Library Class COU Introduce Foreign Method Introduce Local Extension
Comments DIS Rename Method Method Introduce Extract Assertion	Dead Code DIS	



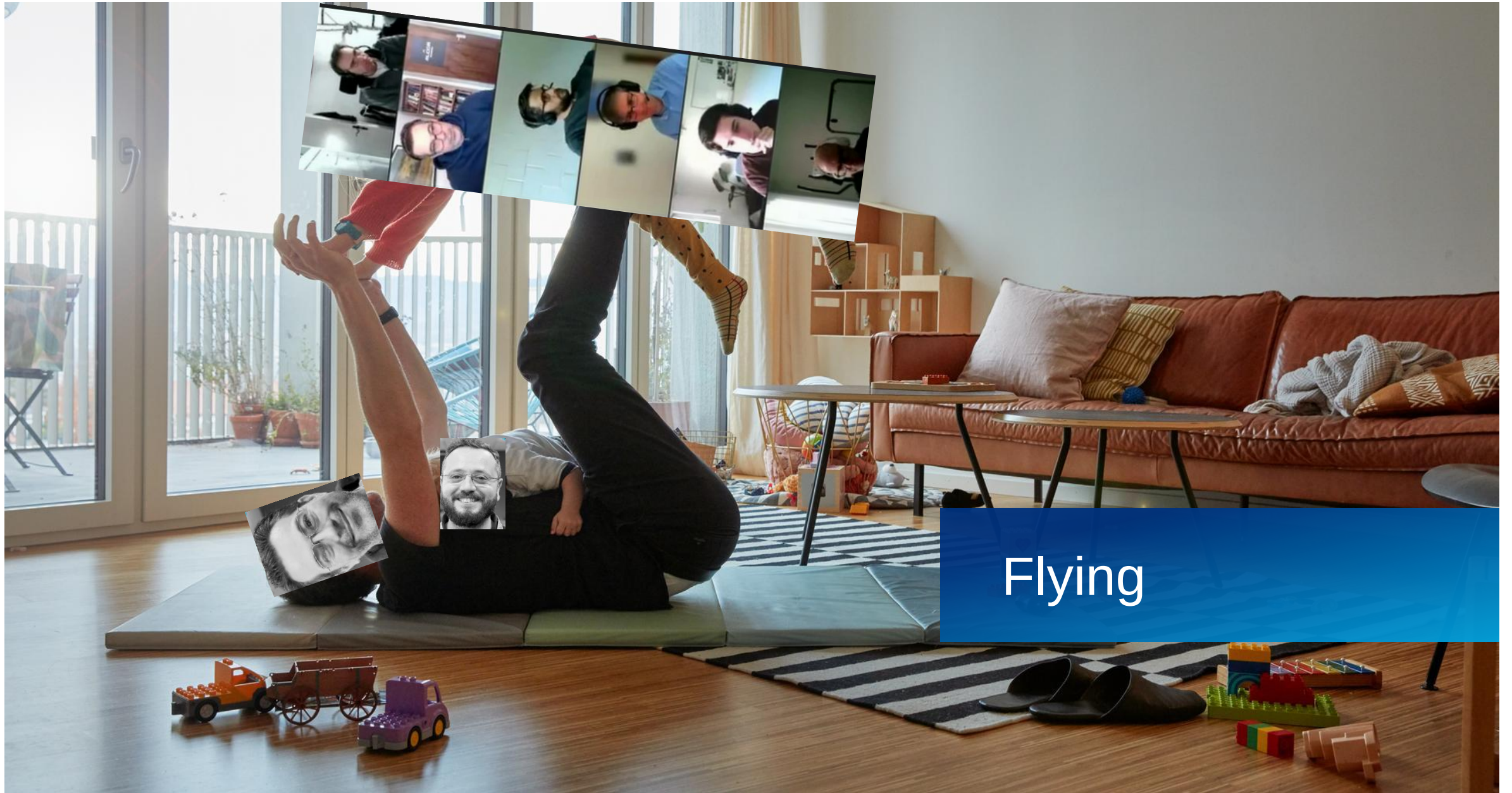
SOLID



Cohesion

Coupling





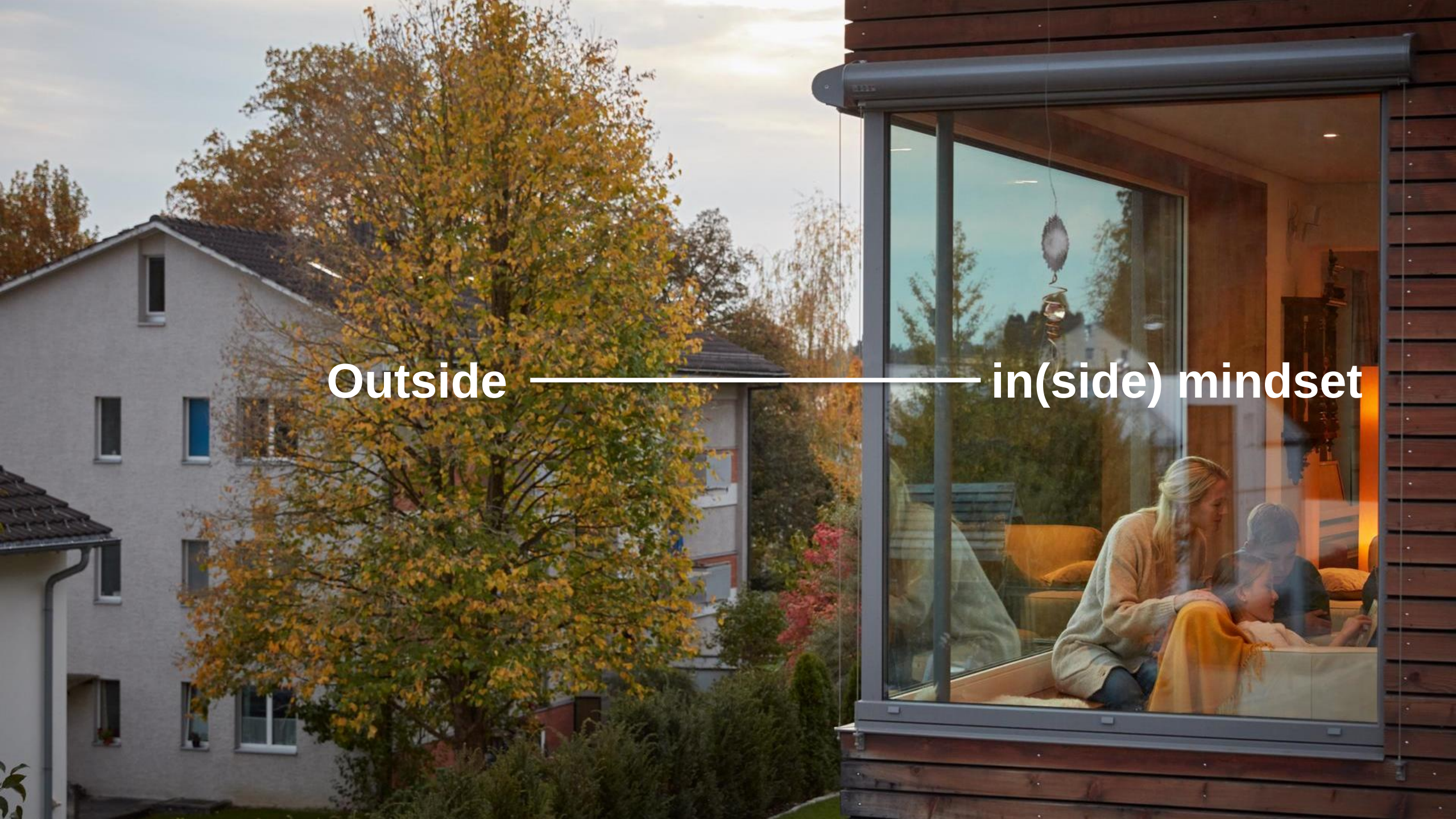
Flying

Connascence

Test Doubles



Outside ————— in(side) mindset







Did it work?





Questions?



Thank you



florian.thalmann@css.ch



<https://www.linkedin.com/in/florian-thalmann-22a68413b/>

- Images:
 - CSS
 - <https://alcor.academy/>
- Cheat sheets: Alcor academy courses

Sources