```
Text(
  'Simple Statement or URL',
  style: TextStyle(
    color: Colors.red[200],
  ),
),
```

devfest

```
s.star,
r: Colors.red[500],

Text('23'),
```

Implementing an idempotent file upload system using Angular

Abdallah Yashir Ramsing

Software Consultant @ Klanik Mauritius

Google Developer Groups

Houzair Koussa

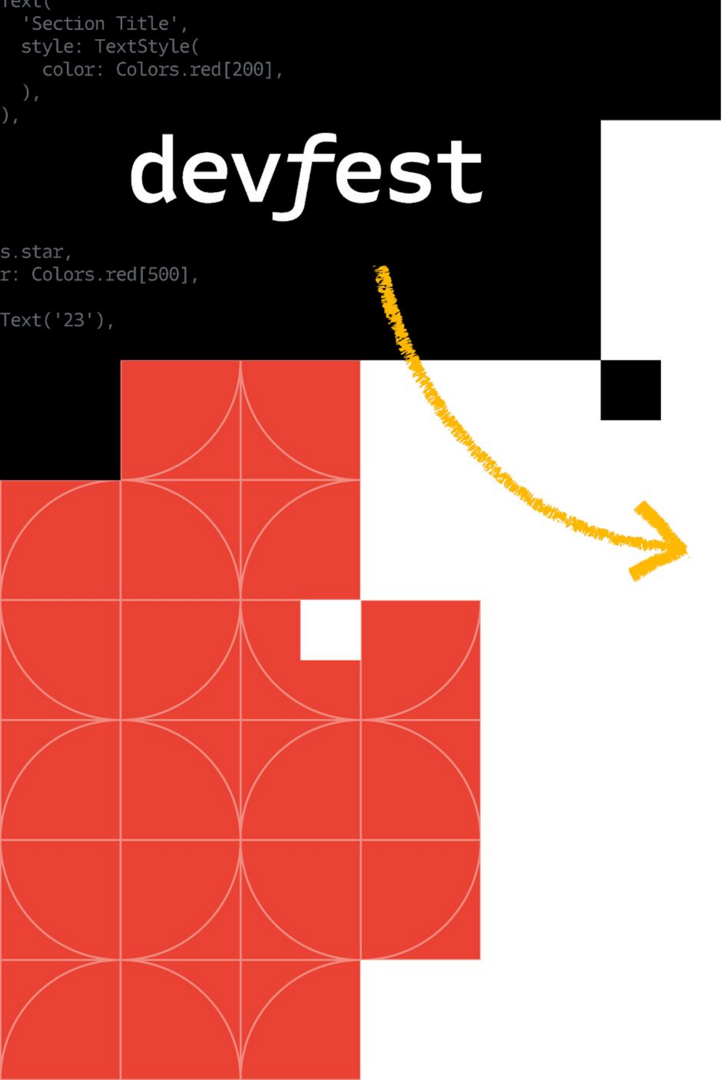
Software Engineer @ Stratalis

Maintainer @ thephilosophicalcode.com

Google Developer Groups

Agenda

1. Philosophically Analyse Idempotency As A Concept
2. Teams File Upload Story
3. Demo
4. Technology
5. Coding walkthrough
6. Q&A



Google Developer Groups

University Of Mauritius

Philosophically Analyse Idempotency As A Concept

Walkthrough

- Intuitive definition and examples of idempotency.
- Step back - formalisation.
- Formal definition of idempotency.
- Case studies -
    - Argument - Idempotency is relative.
    - Proof - Math.abs() is idempotent.

1. Intuitively...

- Core idea -

Repeated Actions. Single Result.

- Generally,
    - Button smashing at traffic lights.
    - Engineering - Single payment.
    - Maths - Multiplying by 0.
    - Law - double jeopardy.

2. Taking a step back

- Idempotency is a bit over the place.
- Talk about idempotency without loss of generality.
- To do that, we can formally define idempotency.
- Our definition acts as a model.
- A model makes it easier to analyse/reason about idempotency.

- Similar to modeling/defining Computation as Turing machines.

3. Formal Definition Of Idempotency

For any set S and a binary operator “•" that operates on the elements of S,

1. An element x of S is idempotent if x • x = x.

2. A operator “•” is idempotent if x • x = x *for all x of S*.

3.1 Example of an idempotent element

- The number “0” under multiplication.
- Our set is the set of Integers.
- Our binary operator is multiplication (*).

$$0 * 0 = 0$$

$$x \cdot x = x$$

3.2 Example of an idempotent operator

- The logical OR operator (“||”).
- Our set is the set of booleans { true, false }.

true || true = true

false || false = false

$x \cdot x = x$ *for all x*

4. Idempotency Is Relative

- Not to be confused with “subjective”.
- An element or operator is idempotent *with respect to something else*.
- So, there is no such thing as absolute idempotence.

- “This is idempotent” is not like saying “2 + 2 = 4”.

4.1 Idempotency Is Relative - Argument

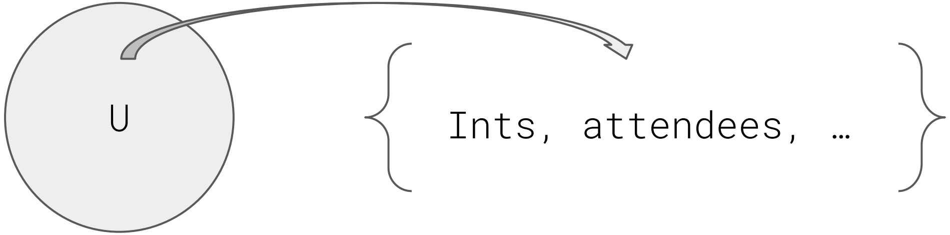
- It’s all about sets (It’s in the definition).

  For any set S and a binary operator “•" that operates on the elements of S …

- Ok but, what if S is the Universal Set.
- Universal Set - the set of all things (including other sets and itself)
- Intuitively, if an element is idempotent in the Universal Set, then it is absolutely idempotent.

4.1 Idempotency Is Relative - Argument

- The Universal Set cannot exist!
- Universal Set implies the existence of the set of all sets that do not contain themselves.



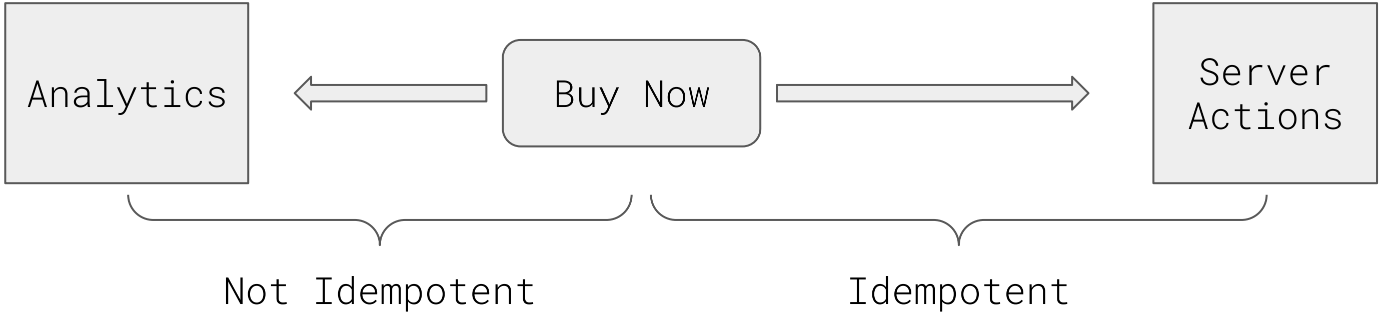
U

{ Ints, attendees, … }

- That leads to the barber’s paradox.
- Hence, S ≠ Universal Set.

4.2 Idempotency Is Relative. That is neat!

- More granular understanding of the software we build.
- Partition the non-idempotent and the idempotent.
- An action may be idempotent *to some relations but not others*.



5. Math.abs() is (annoyingly) idempotent

- Intuitively, Math.abs() is idempotent. It is known.
- Math.abs(Math.abs(Math.abs(5))) = 5.
- Annoyingly, Math.abs() doesn’t neatly fit our definition.

1. An element x is idempotent if x • x = x.
2. A binary operator “•” is idempotent if x • x = x for all x.

- Math.abs() is not a binary operator. It’s unary!

5.1 Math.abs() is idempotent - Proof

- Notice that we can chain Math.abs() because

  Math.abs(Math.abs(x)) = Math.abs()

- This is like saying that

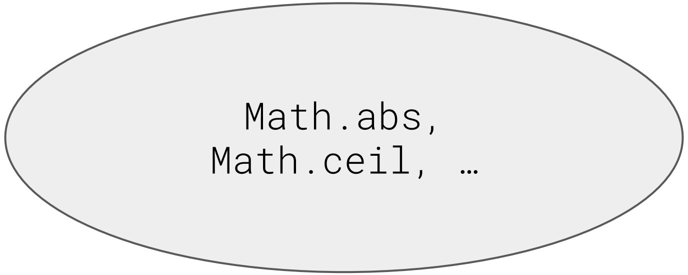
  Math.abs() composed with Math.abs() = Math.abs()

- Composition -

  F composed with G = F(G(x))

5.1 Math.abs() is idempotent - Proof

- Also, notice the type of Math.abs() - Number -> Number.
- Trivially, Math.abs() is a function mapping from the set of Numbers to the set of Numbers.
- Math.abs() is an element of the set of all functions mapping from Numbers to Numbers. Call this set “NN”.



Math.abs,
Math.ceil, …

5.1 Math.abs() is idempotent - Proof

- NN is a set of functions. So, we can compose its elements!
- Composition is a binary operator.
- Recall -
    - An element x of S is idempotent if x • x = x
    - Math.abs() composed with Math.abs() = Math.abs()

- S is NN and our binary operator "•" is composition.
- Hence, Math.abs() is idempotent.

5.2 Math.abs() is idempotent. That is neat!

- We treated Math.abs() as a thing.
- Composition as an operator on that thing.
- Emphasizes the idea of Functions as first-class citizens.

Teams File Upload Story

1. Video demo
2. Share with one colleague
3. Share with another
4. Upload the same file
5. Try to “outsmart” the app

```
Text(
  'Simple Statement or URL',
  style: TextStyle(
    color: Colors.red[200],
  ),
),
```

devfest

```
s.star,
r: Colors.red[500],

Text('23'),
```

DEMO

Design & Trade-offs

1. Use of Angular
2. Backend technologies - GoLang, Kotlin
3. Trade-off: more styling than proper coding standard
4. TailwindCSS & Flowbite
5. Ruby on Rails (RAD)

Angular

Angular is an application-design framework and development platform for creating efficient and sophisticated single-page apps.

1. Batteries included - routing, forms, rxjs
2. Component bases - reusable
3. TypeScript
4. Free & Open Source
5. CLI - Scaffolding
6. Excellent support & integration

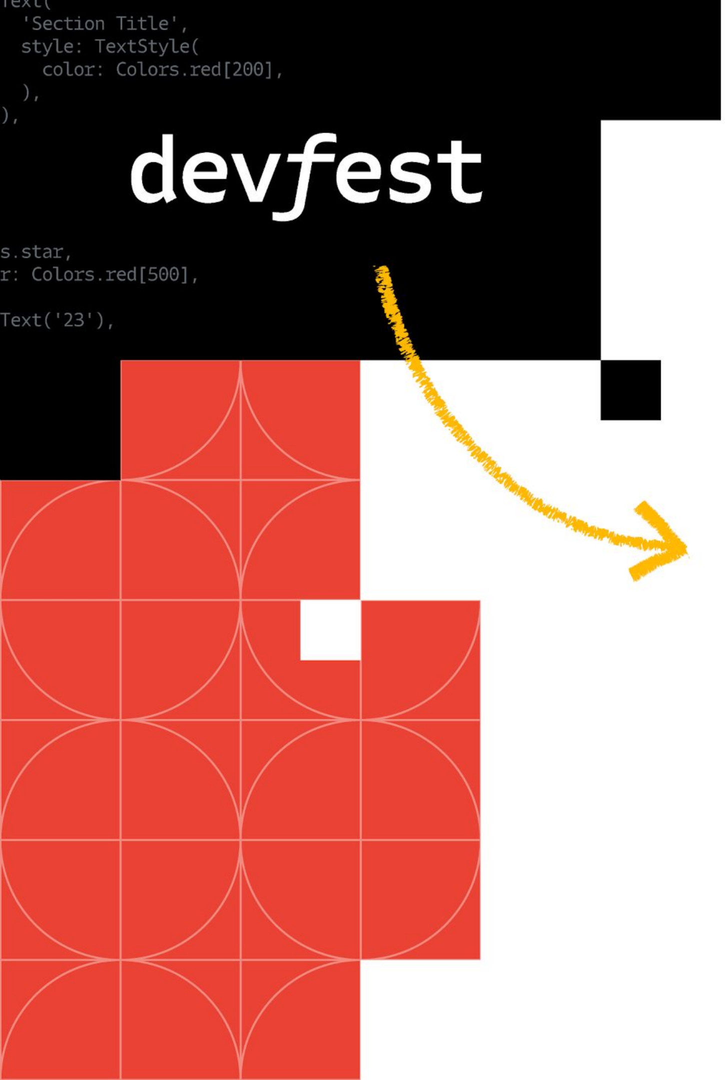
Ruby on Rails

Rails is a full-stack framework. It ships with all the tools needed to build amazing web apps on both the front and back end.

Used by:

- Shopify
- GitHub
- GitLab
- 37 signals
- Zendesk
- Twitch
- Fleetio

```
1.  Rendering HTML templates
2.  Sync databases
3.  Managing emails
4.  Live change via WebSockets
5.  Enqueuing jobs
```

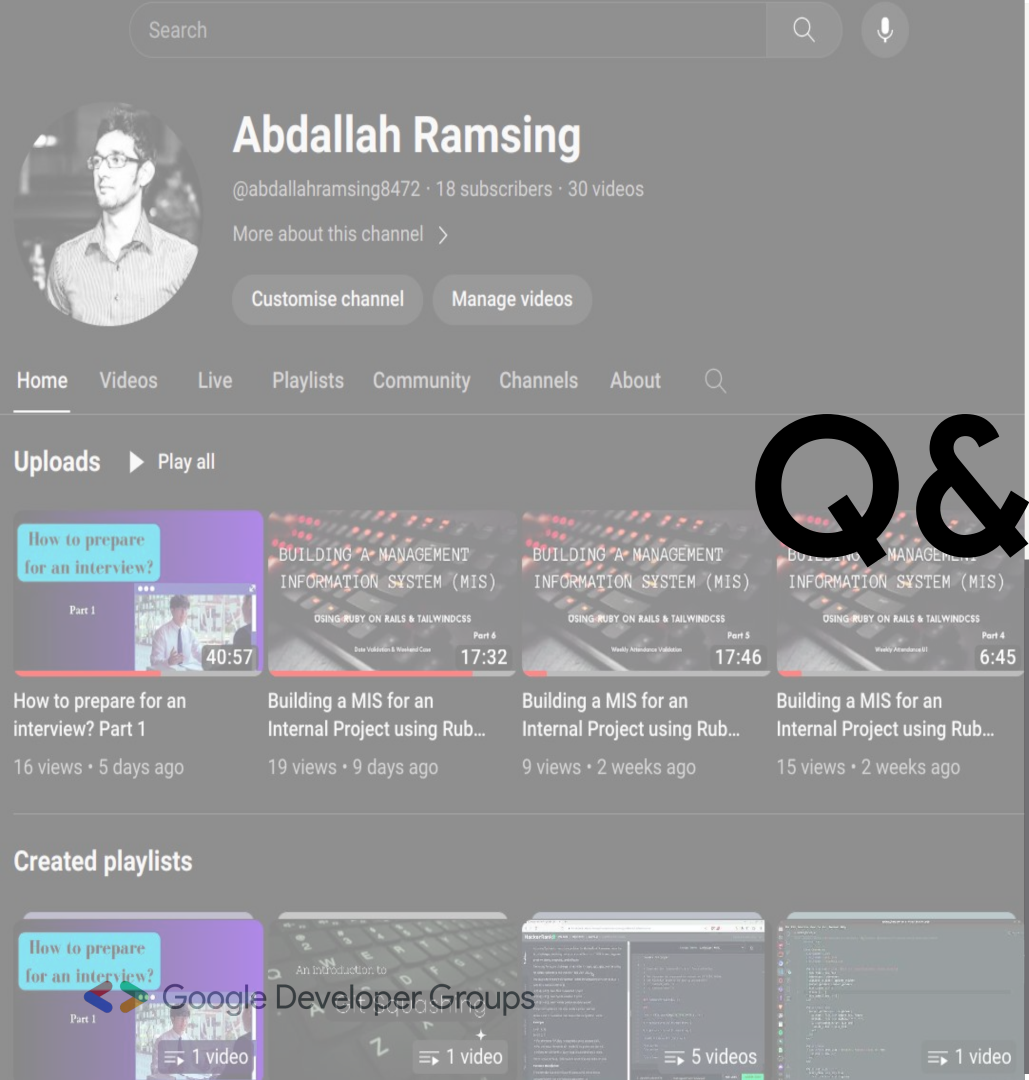


Google Developer Groups

University of Mauritius

Codes walkthrough

Q&A



Search

Abdallah Ramsing

@abdallahramsing8472 · 18 subscribers · 30 videos

More about this channel >

Customise channel | Manage videos

Home | Videos | Live | Playlists | Community | Channels | About

Uploads ▶ Play all

How to prepare for an interview?
Part 1
40:57
BUILDING A MANAGEMENT INFORMATION SYSTEM (MIS)
USING RUBY ON RAILS & TAILWINDCSS
Part 6
17:32
Part 5
17:46
Part 4
6:45

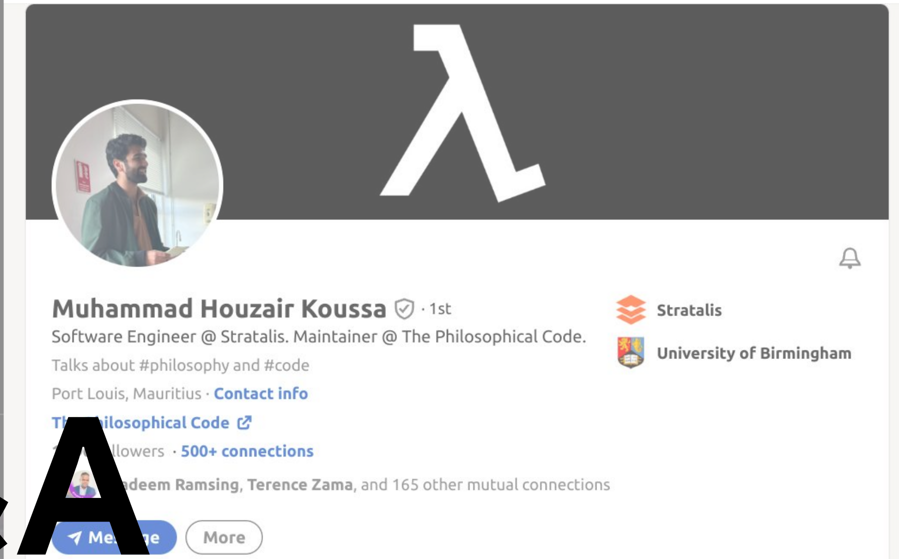
How to prepare for an interview? Part 1
16 views • 5 days ago

Building a MIS for an Internal Project using Rub...
19 views • 9 days ago

Building a MIS for an Internal Project using Rub...
9 views • 2 weeks ago

Building a MIS for an Internal Project using Rub...
15 views • 2 weeks ago

Created playlists



Muhammad Houzair Koussa · 1st

Software Engineer @ Stratalis. Maintainer @ The Philosophical Code.

Talks about #philosophy and #code

Port Louis, Mauritius · Contact info

Th ilosophical Code

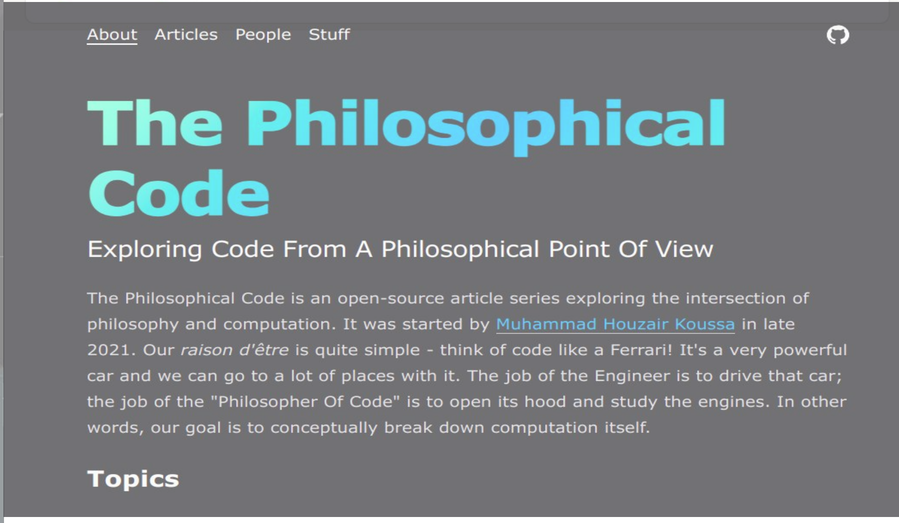
llowers · 500+ connections

deem Ramsing, Terence Zama, and 165 other mutual connections

Stratalis

University of Birmingham

Me ge | More



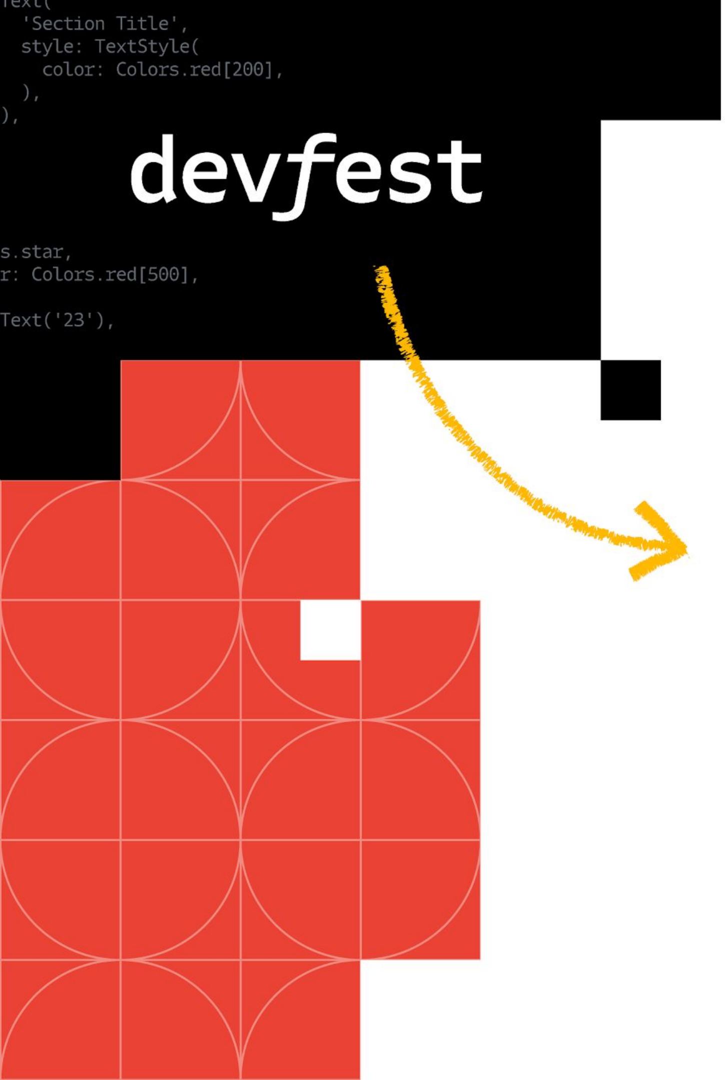
About | Articles | People | Stuff

The Philosophical Code

Exploring Code From A Philosophical Point Of View

The Philosophical Code is an open-source article series exploring the intersection of philosophy and computation. It was started by Muhammad Houzair Koussa in late 2021. Our *raison d'être* is quite simple - think of code like a Ferrari! It's a very powerful car and we can go to a lot of places with it. The job of the Engineer is to drive that car; the job of the "Philosopher Of Code" is to open its hood and study the engines. In other words, our goal is to conceptually break down computation itself.

Topics



```
Text(
  'Section Title',
  style: TextStyle(
    color: Colors.red[200],
  ),
),
```

```
s.star,
r: Colors.red[500],

Text('23'),
```

Google Developer Groups

University Of Mauritius

Thank you!

Read more:
https://thephilosophicalcode.com/articles/idempotency-in-action