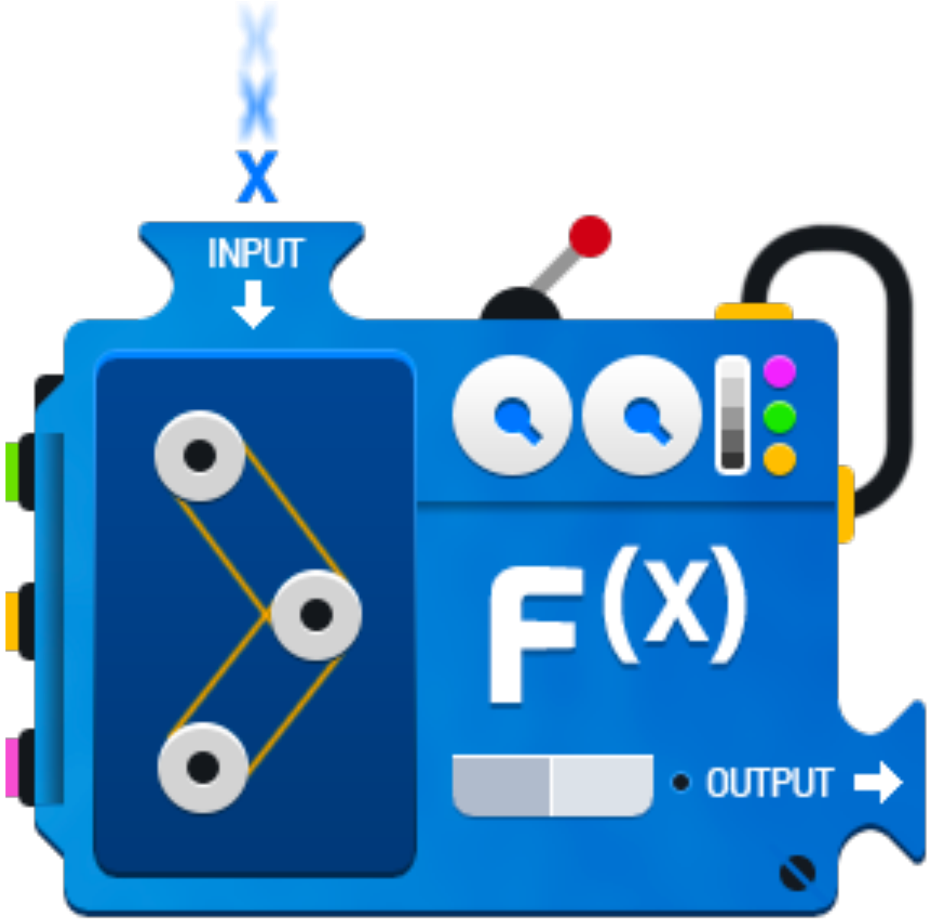


# Functional Programming

Alexey Buzdin

Roman Stranchewsky

# Introduction to Course

32 Academic Hours - 16 Lessons

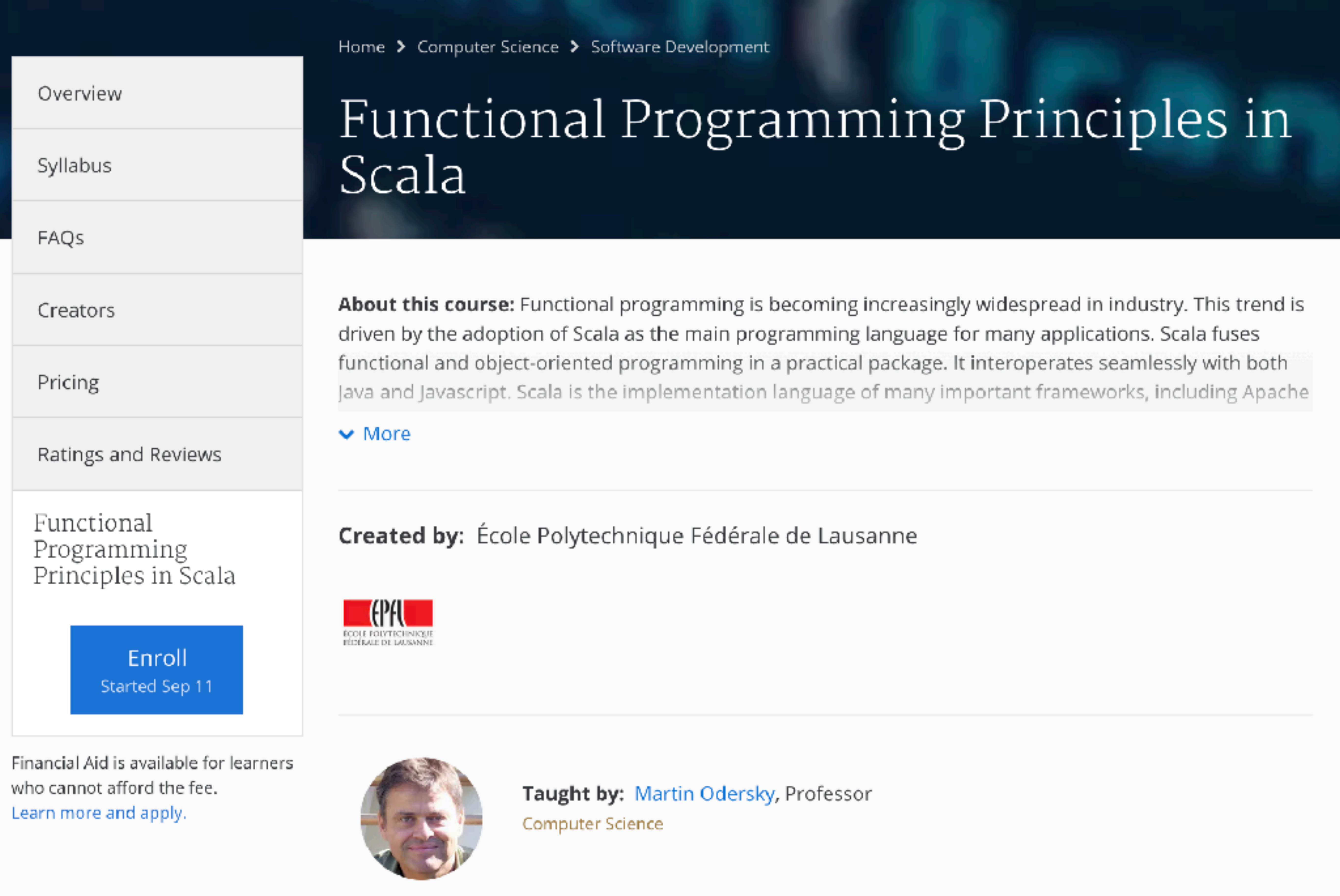
Exam – 60%, homeworks– 30%, activity during the course – 10%

Homework will be accepted as GitHub links only

slack

# Topics

1. Functional programming style.
2. Haskell language. Language elements.
3. Data types.
4. Recursion.
5. Lambda-calculus.
6. Program execution.
7. Imperative program analogies.



Home > Computer Science > Software Development

# Functional Programming Principles in Scala

- Overview
- Syllabus
- FAQs
- Creators
- Pricing
- Ratings and Reviews

Functional Programming Principles in Scala

Enroll
Started Sep 11

Financial Aid is available for learners who cannot afford the fee. Learn more and apply.

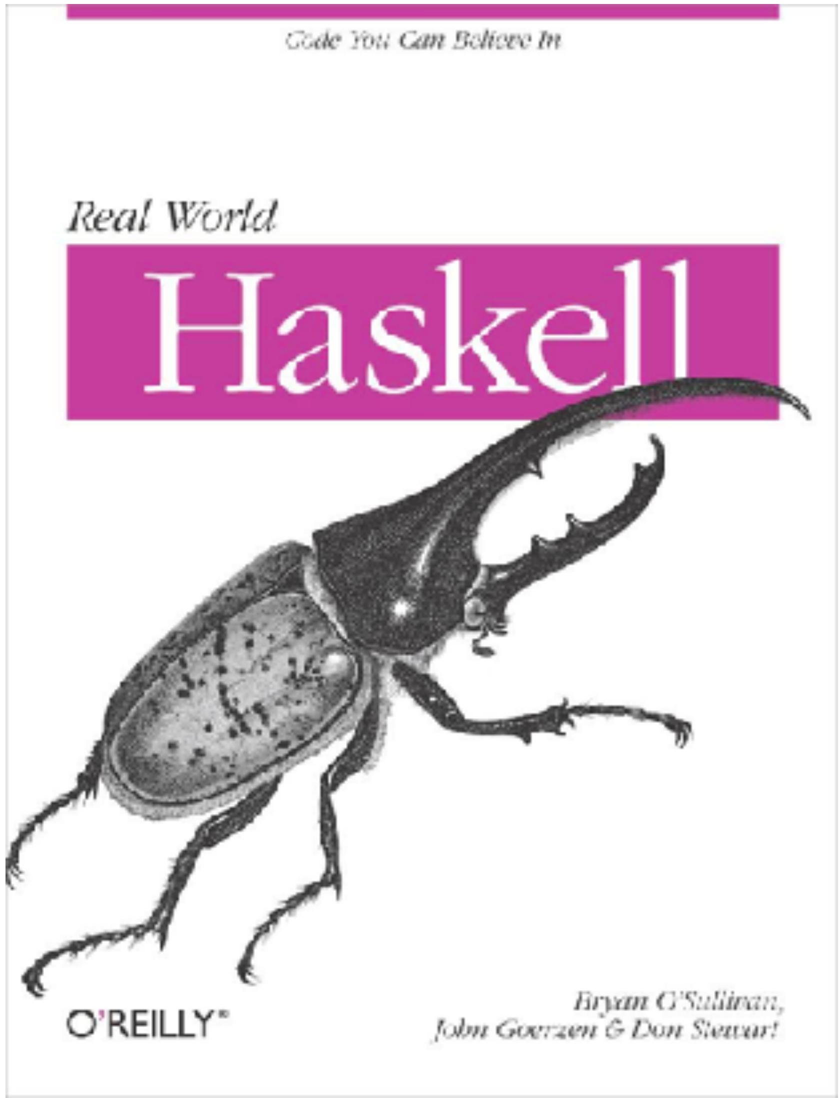
**About this course:** Functional programming is becoming increasingly widespread in industry. This trend is driven by the adoption of Scala as the main programming language for many applications. Scala fuses functional and object-oriented programming in a practical package. It interoperates seamlessly with both Java and Javascript. Scala is the implementation language of many important frameworks, including Apache

More

**Created by:** École Polytechnique Fédérale de Lausanne

**Taught by:** Martin Odersky, Professor
Computer Science

**http://ej.uz/courseraFun**



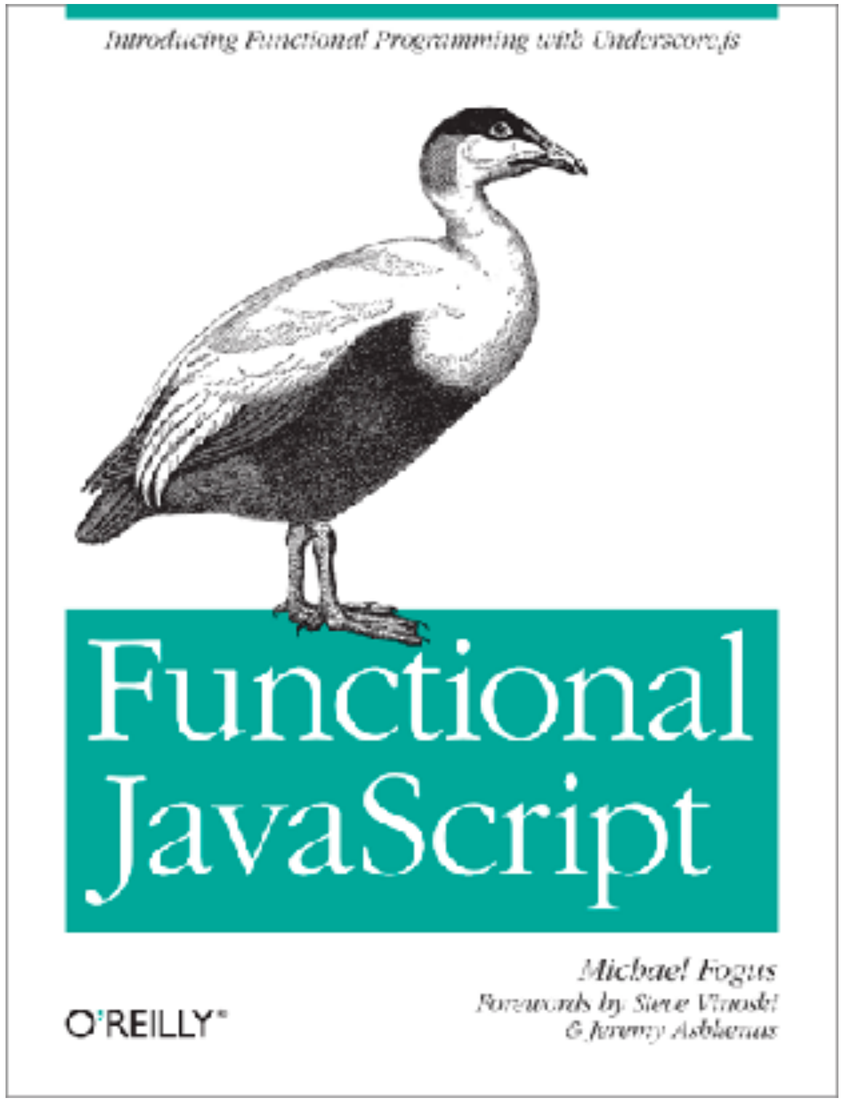
Code You Can Believe In

Real World

Haskell

O'REILLY®

Bryan O'Sullivan,
John Goerzen & Don Stewart

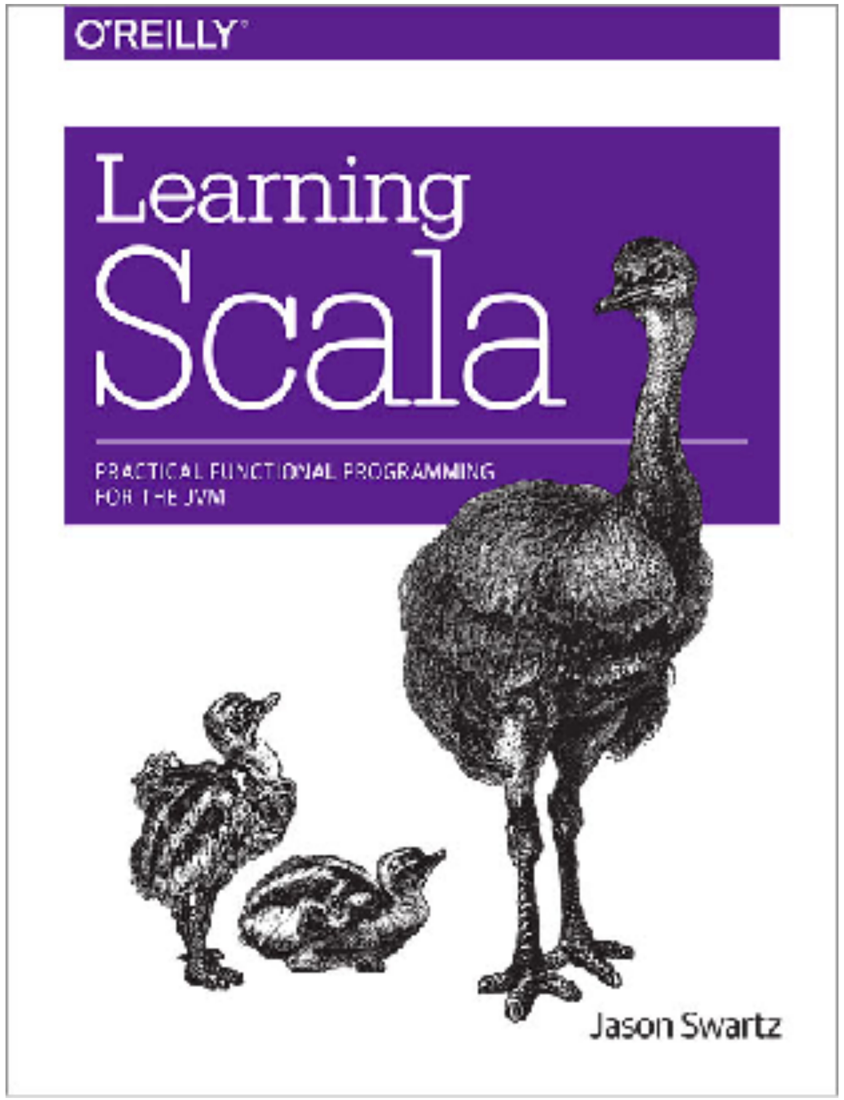


Introducing Functional Programming with Underscore.js

Functional JavaScript

O'REILLY®

Michael Fogus
Forewords by Steve Vinoski
& Jeremy Ashkenas



O'REILLY®

Learning Scala

PRACTICAL FUNCTIONAL PROGRAMMING
FOR THE JVM

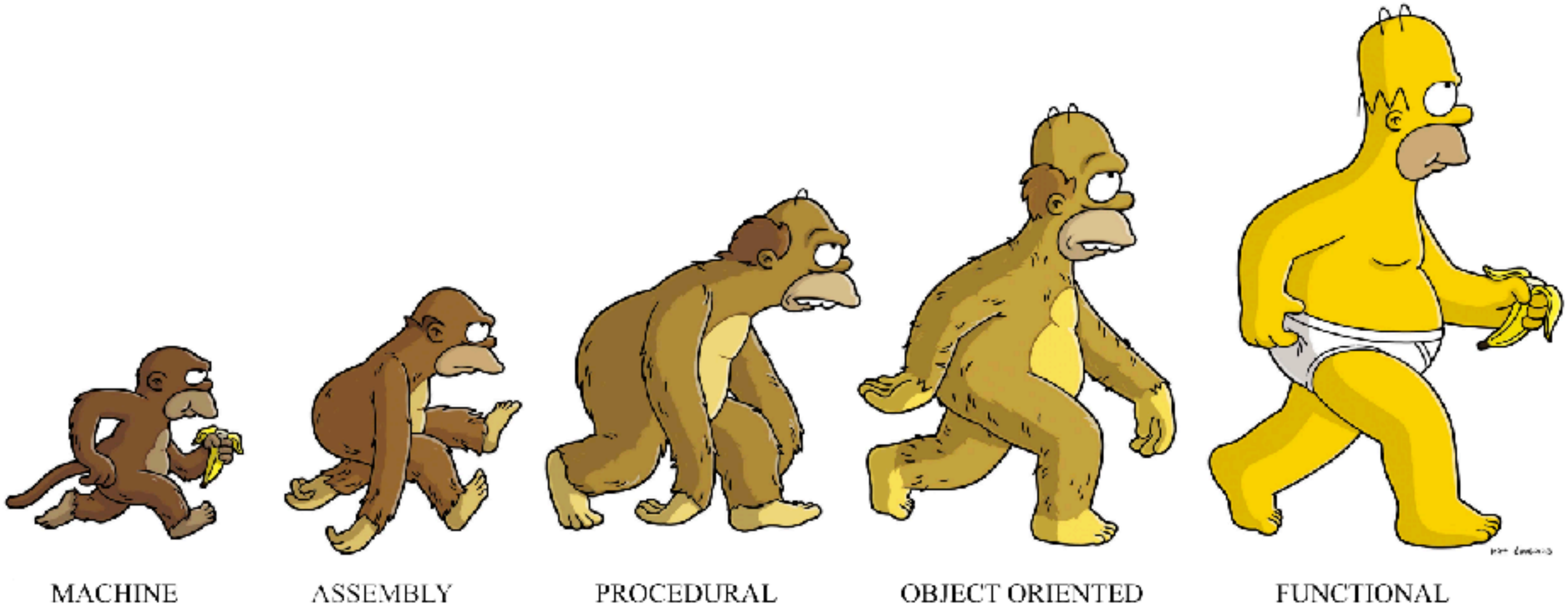
Jason Swartz

**https://www.oreilly.com/**

# GitHub

# GitHub

https://education.github.com/

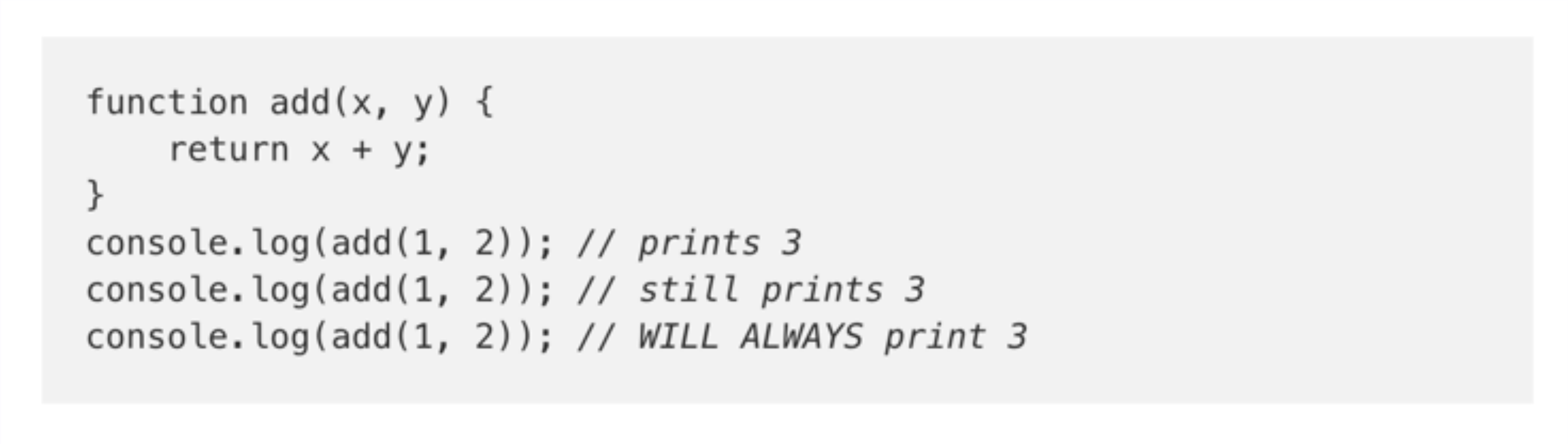


https://medium.com/@cscalfani/so-you-want-to-be-a-functional-programmer-part-1-1f15e387e536

# Awesome.

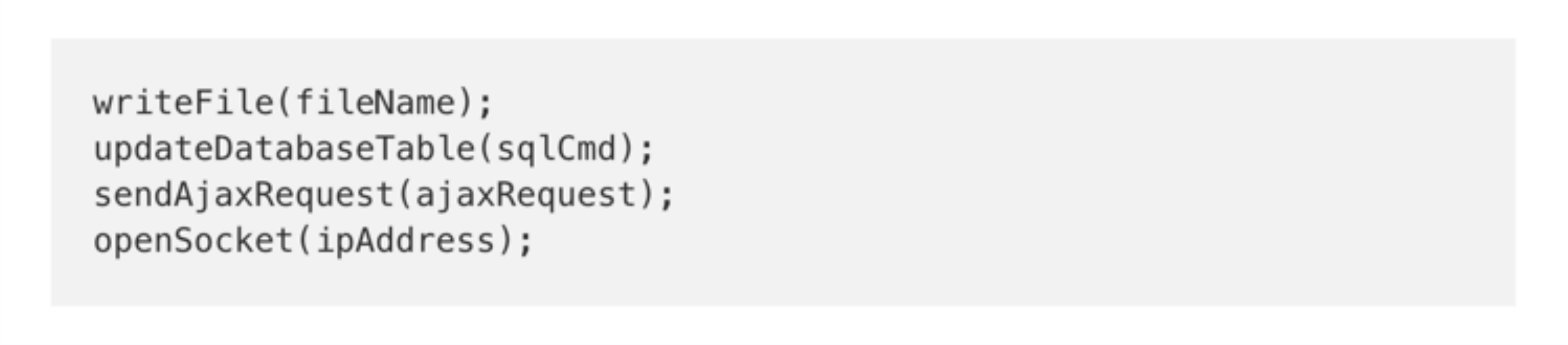
**https://github.com/xgrommx/awesome-functional-programming**

# Pure



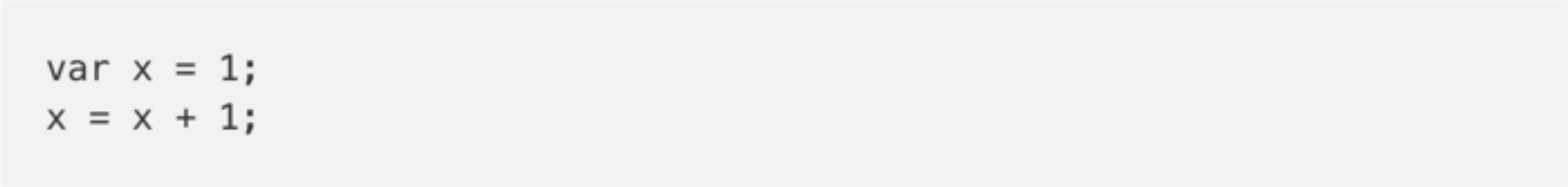
```
function add(x, y) {
    return x + y;
}
console.log(add(1, 2)); // prints 3
console.log(add(1, 2)); // still prints 3
console.log(add(1, 2)); // WILL ALWAYS print 3
```

# Impure

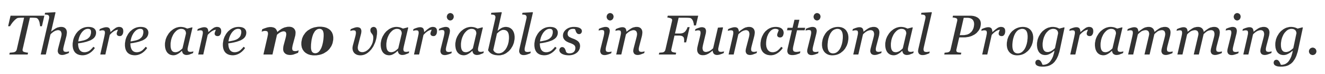


```
writeFile(fileName);
updateDatabaseTable(sqlCmd);
sendAjaxRequest(ajaxRequest);
openSocket(ipAddress);
```

# Not possible

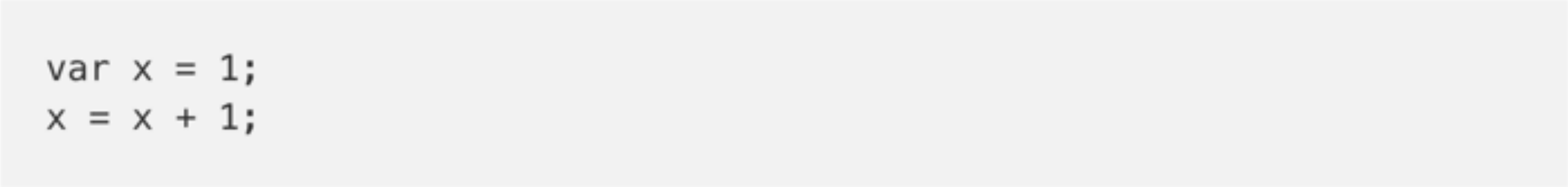


```
var x = 1;
x = x + 1;
```



*There are **no** variables in Functional Programming.*

# Not possible



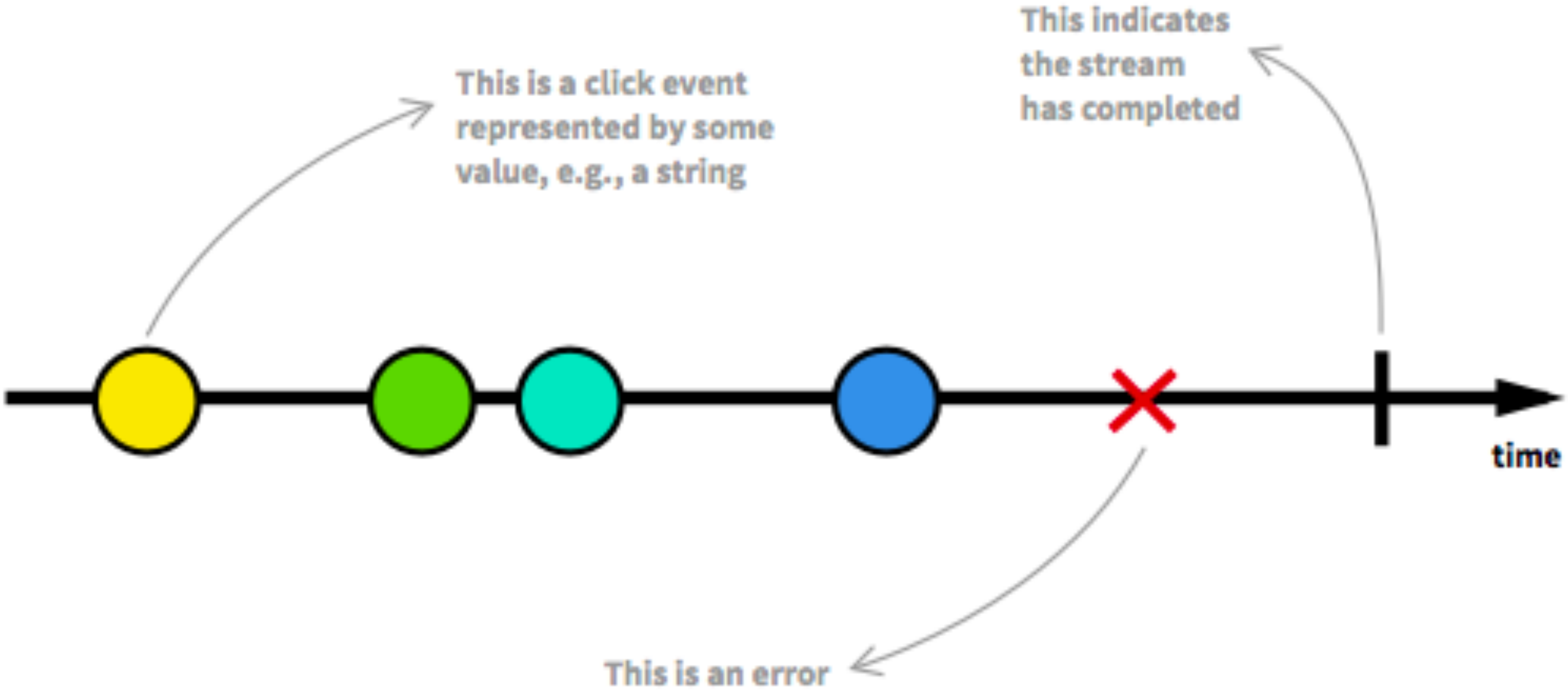
```
var x = 1;
x = x + 1;
```

*There are* ***no*** *variables in Functional Programming.*

*There are* ***no*** *loops in Functional Programming.*

https://en.wikipedia.org/wiki/Functional_programming

# Reactive programming


This is a click event represented by some value, e.g., a string
This indicates the stream has completed
time
This is an error

@AlexeyBuzdin

@_romstr_