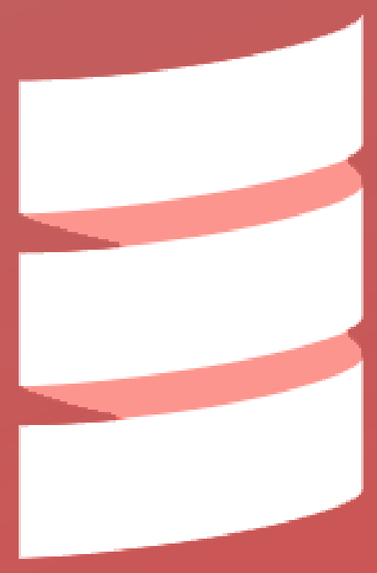


Scala

Fuses Object-Oriented & Functional Paradigms

The Scalable Language Of The Future

“Scala was designed to show that a fusion of functional and object-oriented programming is possible and practical.”

- SCALA'S CREATOR MARTIN ODERSKY

Scala is a pure object-orientated language. Every value is an object.

Scala is also a functional language in the sense that every function is a value

Scala's name is derived from 'Scalable Language'

As such Scala has a keen eye towards the development of scalable applications.

“Scala is deep where other languages are broad.”

- SCALA'S CREATOR MARTIN ODERSKY

Scala provides developers with an incredibly dense toolset. Allowing developers to select the best way to tackle any given situation.

VISIT THE SCALA WEBSITE AT

https://www.scala-lang.org/

SCAN ME

Scala

```scala
public class Car {
    var id: Int = _
    var name: String = _
    var parts: List[Part] = Nil
}

public class Part {
    var id: Int = _
    var name: String = _
}
```

Java

```java
public class Car {
    private int id;
    private String model;
    private List<Part> parts;

    public Car() {
        parts = new ArrayList<Part>();
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public int getModel() {
        return model;
    }

    public void setModel(String model) {
        this.model = model;
    }

    public List<Part> getParts() {
        return parts;
    }

    public void setParts(List<Part> parts) {
        this.parts = parts;
    }
}

public class Part {
    private int id;
    private String name;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public int getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

All In One Concise & Elegant Language

What's Going On Here?

Here are two classes being declared in both Scala and Java .

Both implementations have the same functionality - They define a Car class which in turn has its own list of Parts which is also defined.

We can see that Java achieves this functionality within 54 lines of code. While Scala manages this in a cool 10 lines of code.

Scala reduced the amount of boilerplate code required as compared to Java projects. Thus increasing development efficency.

Fully Interopable With

And Just When You Thought It Couldn't Get Any Better..

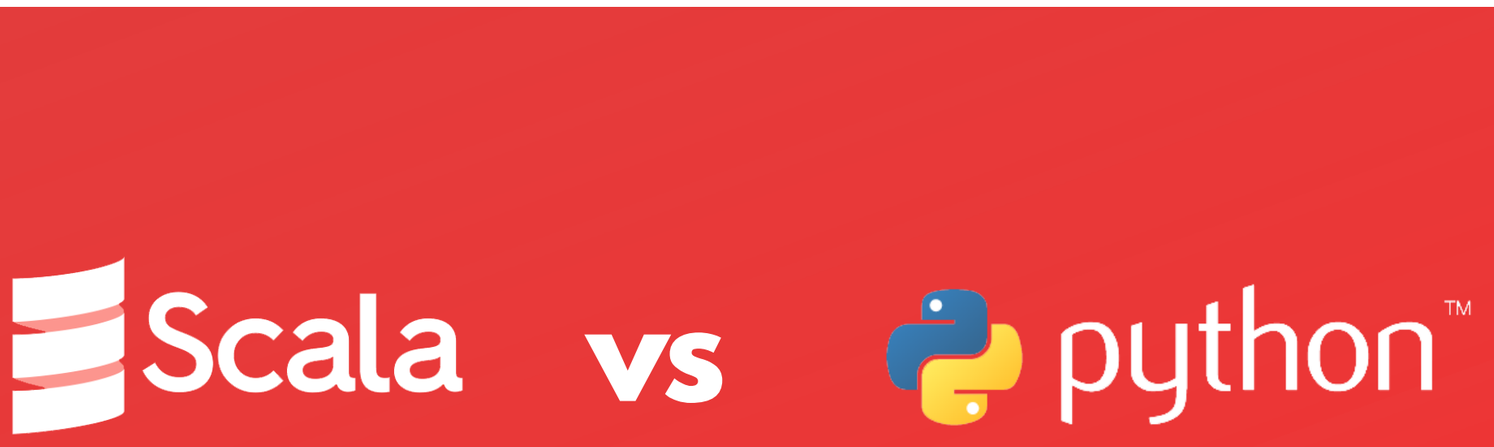
Scala was developed on the Java Virtual Machine, and as such reaps the performance rewards of utilising such an established VM. However this is not the only advantage Scala receives from this fact. Scala was designed to be fully interoperable with Java. Meaning that all of Java's libraries are available for use within Scala.

Powerful Concurrency

Scala offers an Actor model for use in concurrency. In a system that employs this model all entities are Actors.
Actors can communicate with each other by sending messages, contrive actions from received messages and also create other Actors. In addition to this model Scala also offers Futures, and being interoperable with Java, also offers Java's standard Thread model. This project compares these two models and explores what they mean for Scala as a language as a whole.
Please ask for more details!

Popular Use's For Scala:

- Data Science & Big Data
- Concurrenct Applications
- Web Applications



Apache Spark is a general-purpose, unified framework most often used for applications within data science.
The frameworks biggest attraction is that it allows for seamless distributed processing across a cluster.

Apache Spark is an open-source development written in Scala, which is a key driver in Scala adoption. Part of this project focuses on how Scala fares against it's main competitor on the platform - Python.
Please ask for more details.

ALTERNATIVELY, MAYBE ASK HOW SCALA'S MOST VOCAL BACKERS USE THE LANGUAGE