



Parallelism

HÖGSKOLAN
I SKÖVDE

Features

```
import scala.concurrent.ExecutionContext.Implicits.global
import scala.concurrent.{Future, Await}
import scala.concurrent.duration._
import scala.util.{Success, Failure}
```

The logo of Högskolan i Skövde is a circular emblem. It features a central shield with a crown on top, flanked by two lions. The shield is surrounded by a laurel wreath. Below the shield, the year '1977' is inscribed. The entire logo is rendered in a dark, semi-transparent style.

HÖGSKOLAN
I SKÖVDE

Features

```
val f = Future[Double] {  
    // some (expensive?) operation to be executed in another thread.  
}  
  
f.isCompleted  
  
f.value match {  
    case None => "Not yet completed"  
    case Some(Success(v)) => s"Result is $v"  
    case Some(Failure(e)) => s"Something went wrong: $e"  
}
```

HÖGSKOLAN
I SKÖVDE

Features

```
val f = Future[Double] {  
    2 + 2  
  
}  
  
f.isCompleted      // = true  
  
f.value match {  
    case None => "Not yet completed"  
    case Some(Success(v)) => s"Result is $v" ←  
    case Some(Failure(e)) => s"Something went wrong: $e"  
}
```

HÖGSKOLAN
I SKÖVDE

Features

```
val f = Future[Double] {  
  5 / 0  
  
}  
  
f.isCompleted      // = true  
  
f.value match {  
  case None => "Not yet completed"  
  case Some(Success(v)) => s"Result is $v"  
  case Some(Failure(e)) => s"Something went wrong: $e" ←  
}
```

HÖGSKOLAN
I SKÖVDE

Features

```
val f = Future[Double] {  
  Thread.sleep(2000)  
  2 + 2  
}  
  
f.isCompleted      // = false  
  
f.value match {  
  case None => "Not yet completed" ←  
  case Some(Success(v)) => s"Result is $v"  
  case Some(Failure(e)) => s"Something went wrong: $e"  
}
```

HÖGSKOLAN
I SKÖVDE

Features

```
val f = Future[Double] {  
  Thread.sleep(2000)  
  2 + 2  
}.onComplete {  
  case None => "Not yet completed"  
  case Some(Success(v)) => s"Result is $v"  
  case Some(Failure(e)) => s"Something went wrong: $e"  
}
```

The logo of Högskolan i Skövde is a circular emblem. It features a central shield with a stylized bird (possibly a phoenix) rising from flames. The shield is flanked by two stars and a book. The entire emblem is encircled by a laurel wreath. Below the wreath, the year '1977' is inscribed.

HÖGSKOLAN
I SKÖVDE

Features

```
val f = Future[Double] {  
  Thread.sleep(2000)  
  2 + 2  
}  
  
Await.result(f, 3.seconds) // = 4  
  
f.value match {  
  case None => "Not yet completed"  
  case Some(Success(v)) => s"Result is $v" ←  
  case Some(Failure(e)) => s"Something went wrong: $e"  
}
```

HÖGSKOLAN
I SKÖVDE

Fork

```
val df = DataFrame.read("/path/to/data.csv")
df.map(...)           // takes ~8 seconds
val (df1, df2) = df.split(0.5)
val f = Future[DataFrame] { df1.map(...) } // takes ~4 seconds
val res2 = df2.map(...) // takes ~4 seconds
val res1 = Await.result(f, 3.seconds)
val dfMapped = res1 + res2
```

HÖGSKOLAN
I SKÖVDE

Fork

```
def fork[A, B] (a: => A, b: => B): (A, B) = {  
  val f = Future { b }  
  val r1 = a  
  val r2 = Await.result(f, 5.seconds)  
  (r1, r2)  
}
```

The logo of Högskolan i Skövde is a circular emblem. It features a central shield with a stylized tree and a building. The shield is surrounded by a laurel wreath. Below the shield, the year '1977' is inscribed.

HÖGSKOLAN
I SKÖVDE

Fork

```
val df = DataFrame.read("/path/to/data.csv")  
val (df1, df2) = df.split(0.5)  
val (res1, res2) = fork(df1.map(...), df2.map(...))  
val dfMapped = res1 + res2
```

The logo of Högskolan i Skövde is a circular emblem. It features a central shield with a stylized tree and a sun. The shield is surrounded by a laurel wreath. Below the wreath, the year '1977' is inscribed. The entire logo is rendered in a dark, semi-transparent style.

HÖGSKOLAN
I SKÖVDE

Fork

```
val df = DataFrame.read("/path/to/data.csv")
val (df1, df2) = df.split(0.5)
val (df1a, df1b) = df1.split(0.5)
val (df2a, df2b) = df2.split(0.5)
val ((a, b), (c, d)) = fork(
  fork(df1a.map(...), df1b.map(...)),
  fork(df2a.map(...), df2b.map(...))
)
a + b + c + d
```

HÖGSKOLAN
I SKÖVDE

Parallel collections

```
val l = (1 to 100000).toList  
val pl = l.par      // ParVector  
pl.map(...)
```

The logo of Högskolan i Skövde is a circular emblem. It features a shield with a lion rampant on the left and a stylized tree on the right. The shield is surrounded by a laurel wreath. Below the wreath, the year '1977' is inscribed.

HÖGSKOLAN
I SKÖVDE

Akka

```
import akka.actor.{Actor, ActorSystem, Props}  
import akka.event.Logging  
import akka.pattern.ask  
import scala.concurrent.duration._
```

The logo of Högskolan i Skövde is a circular emblem. It features a shield in the center with a stylized tree and a building. The shield is surrounded by a laurel wreath. Below the shield, the year '1977' is inscribed.

HÖGSKOLAN
I SKÖVDE

Akka

```
class Processor extends Actor {  
  val log = Logging(context.system, this)  
  
  def receive = {  
    case Processor.Map(df) => sender() ! df.map(...)   
    case _ => log.info("received unknown message")  
  }  
}  
  
object Processor {  
  case class Map(df: DataFrame)  
}
```

HOGSKOLAN
I SKÖVDE

Akka

```
val system = ActorSystem("CruncherSystem")  
val dc = system.actorOf(Props[Processor], "Cruncher")  
val df = DataFrame.read("/path/to/data.csv")  
val f = dc.ask(Processor.Map(df)) (5.seconds)
```



1977
HÖGSKOLAN
I SKÖVDE