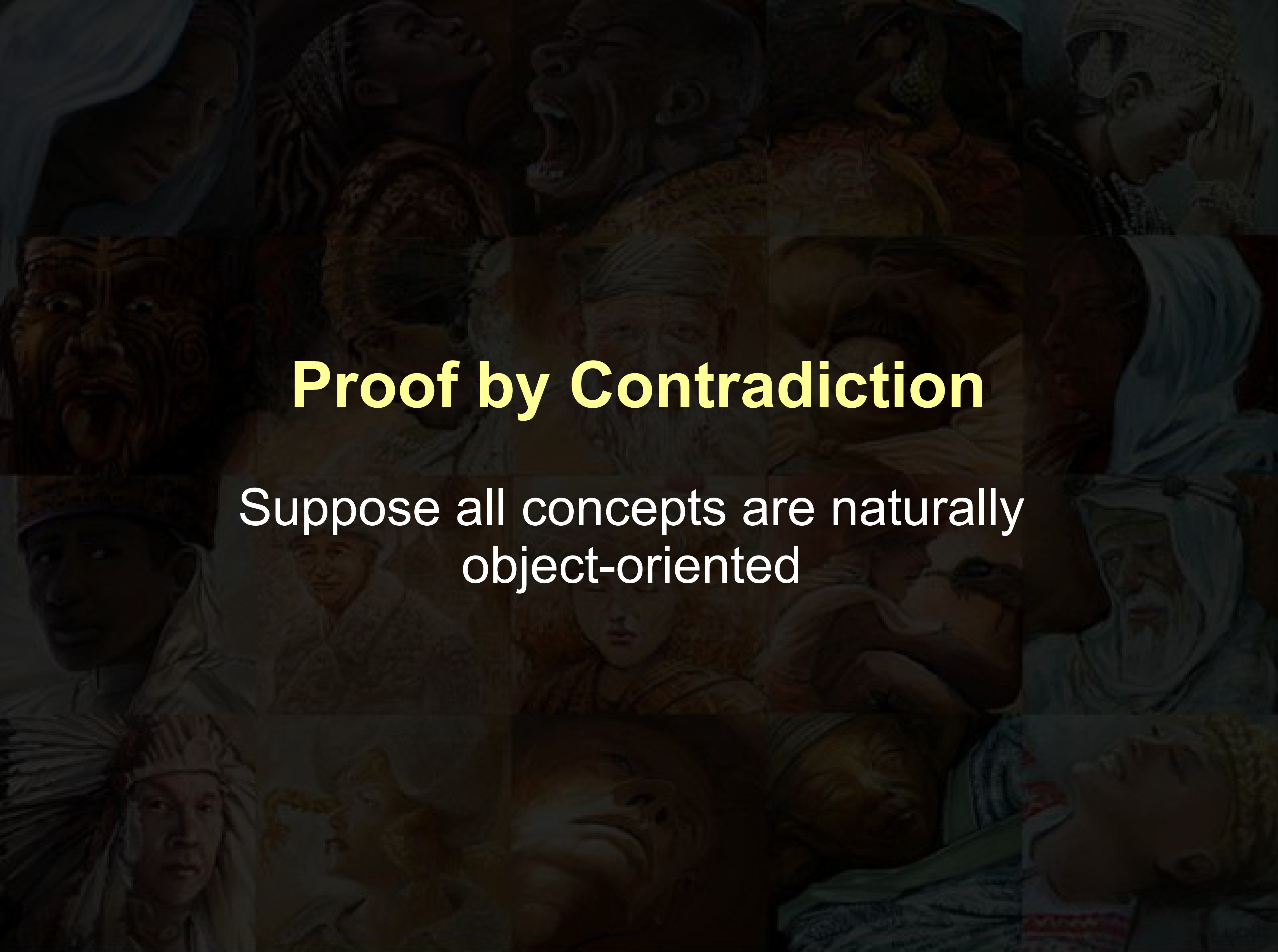
A Beautiful Paradigm

Functional Programming in Finance

Brian Lee Yung Rowe
Director, Engineering and Product
trueSEF

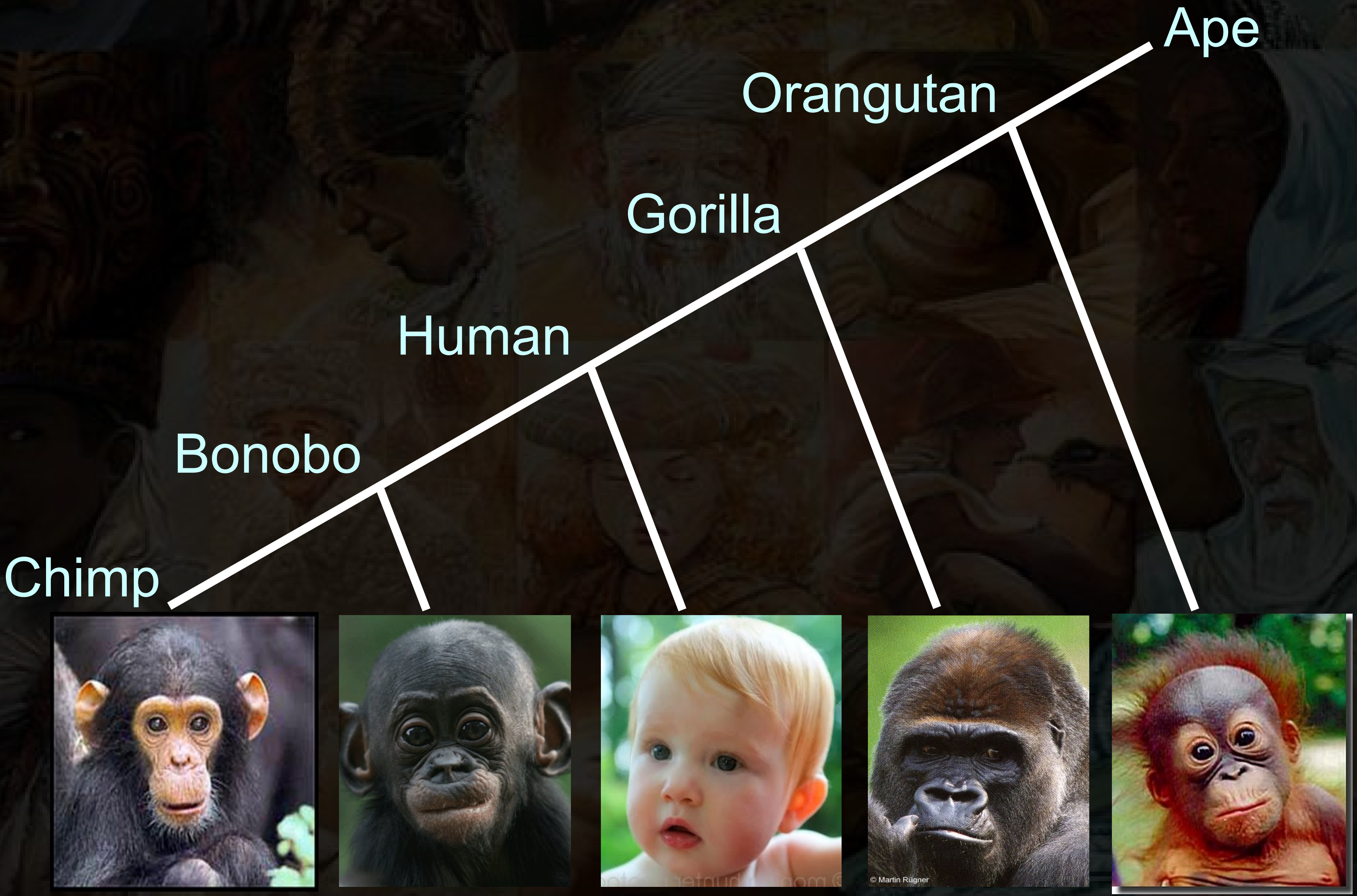
brian@truesef.com



Proof by Contradiction

Suppose all concepts are naturally object-oriented

Classes In the Real World



Ape

Orangutan

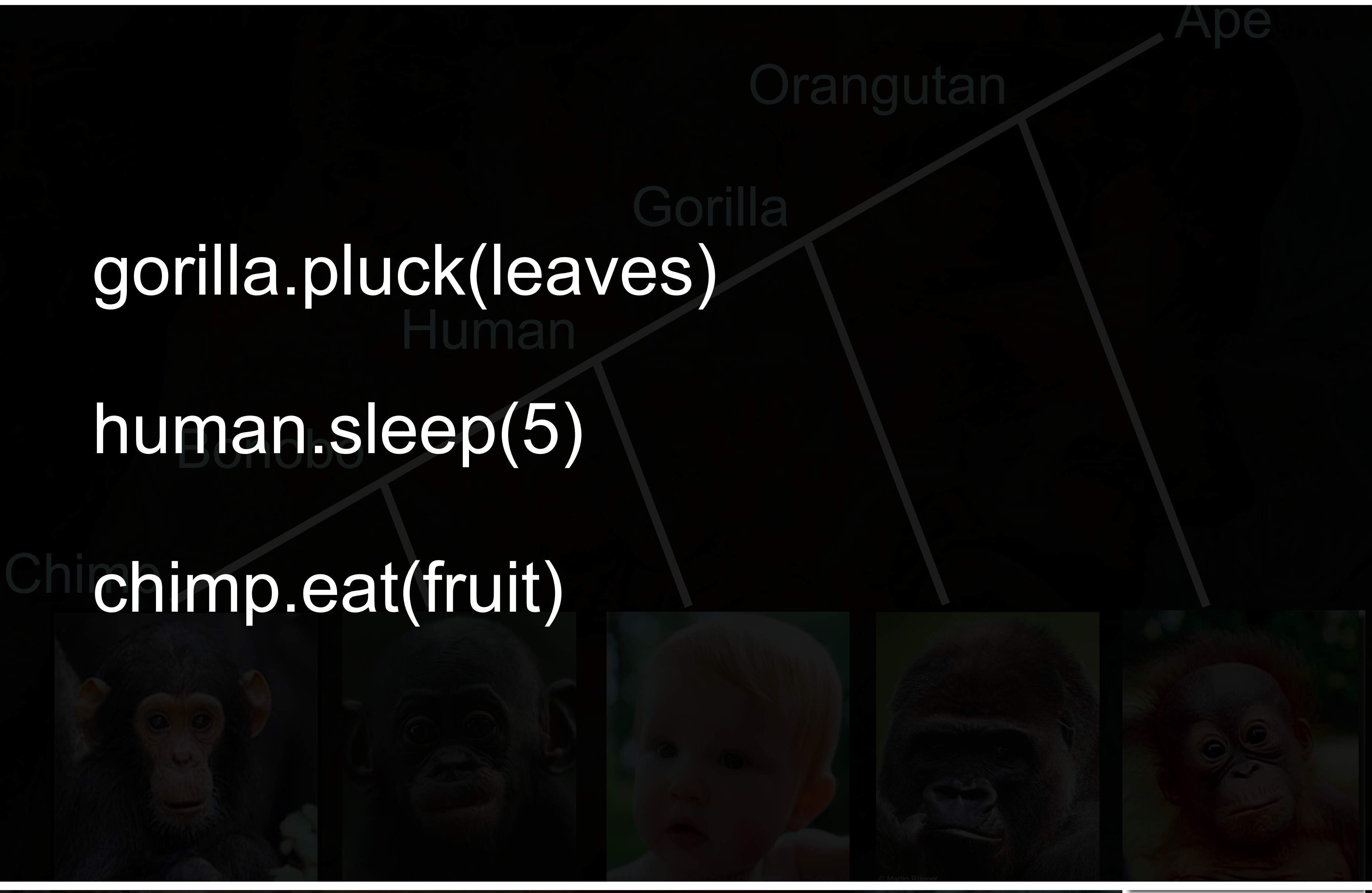
Gorilla

Human

Bonobo

Chimp

Classes In the Real World

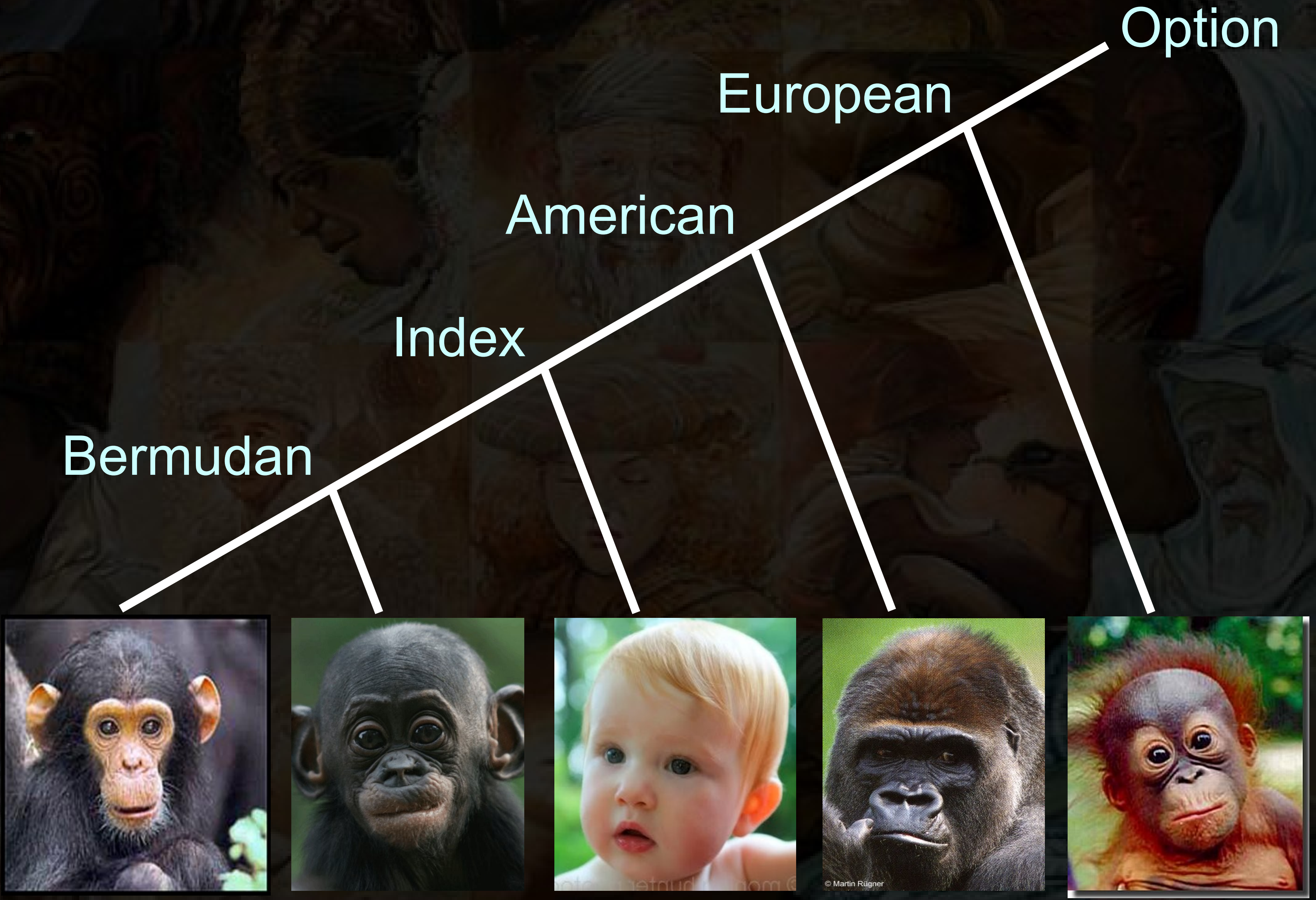


```
gorilla.pluck(leaves)

human.sleep(5)

chimp.eat(fruit)
```

Classes In Finance



Option

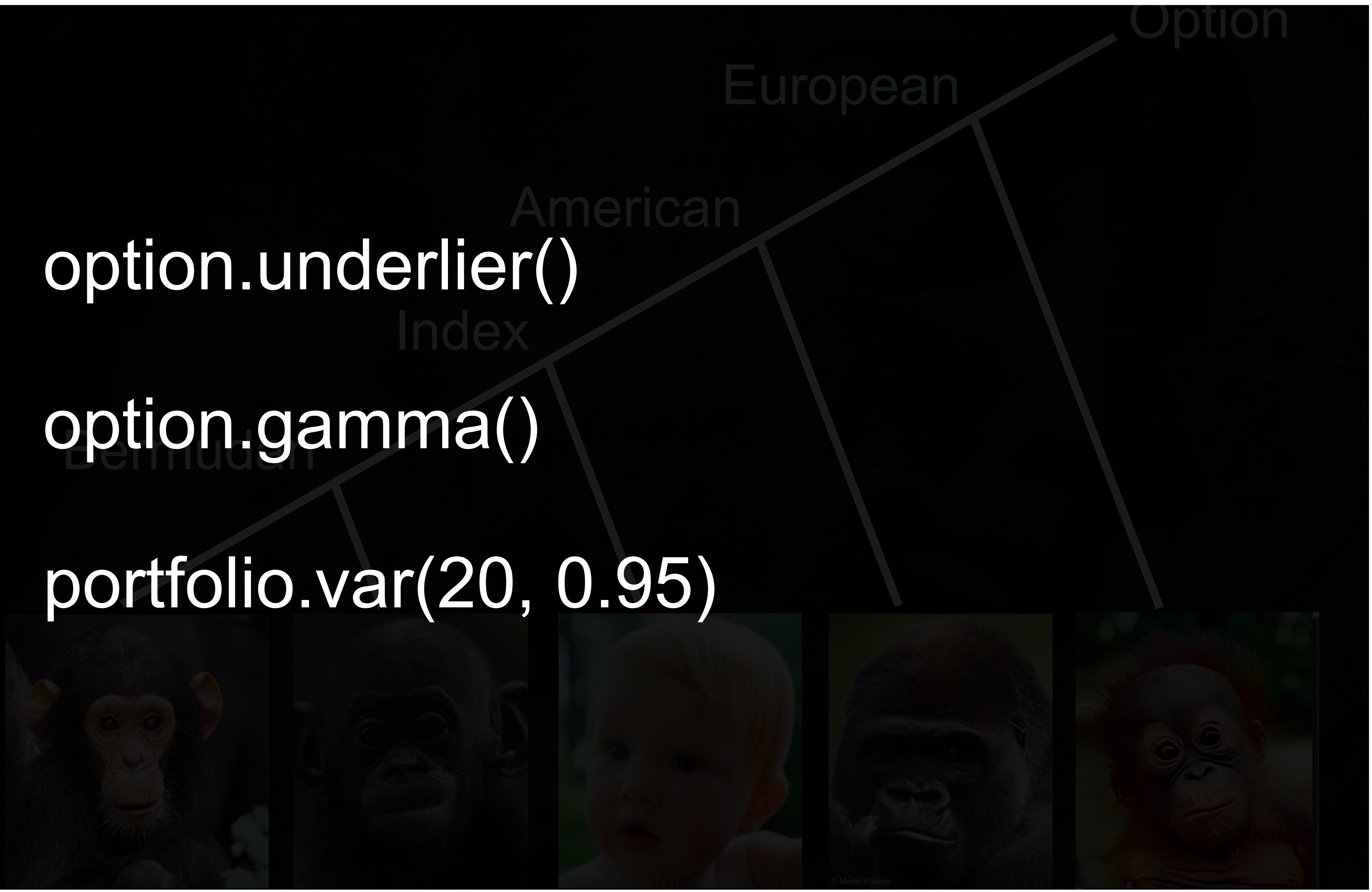
European

American

Index

Bermudan

Classes In Finance

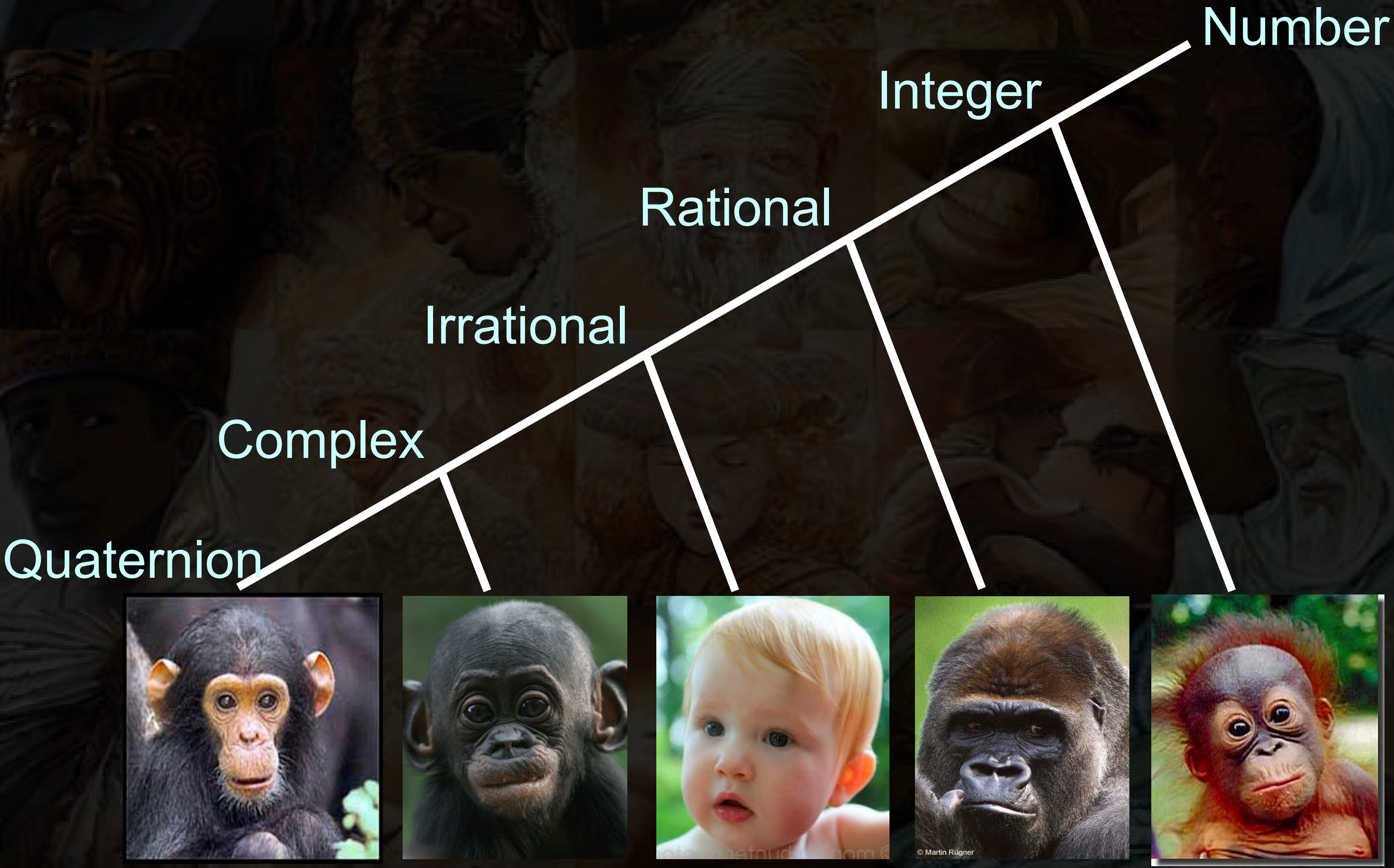


```
option.underlier()

option.gamma()

portfolio.var(20, 0.95)
```

Classes In Math



Number

Integer

Rational

Irrational

Complex

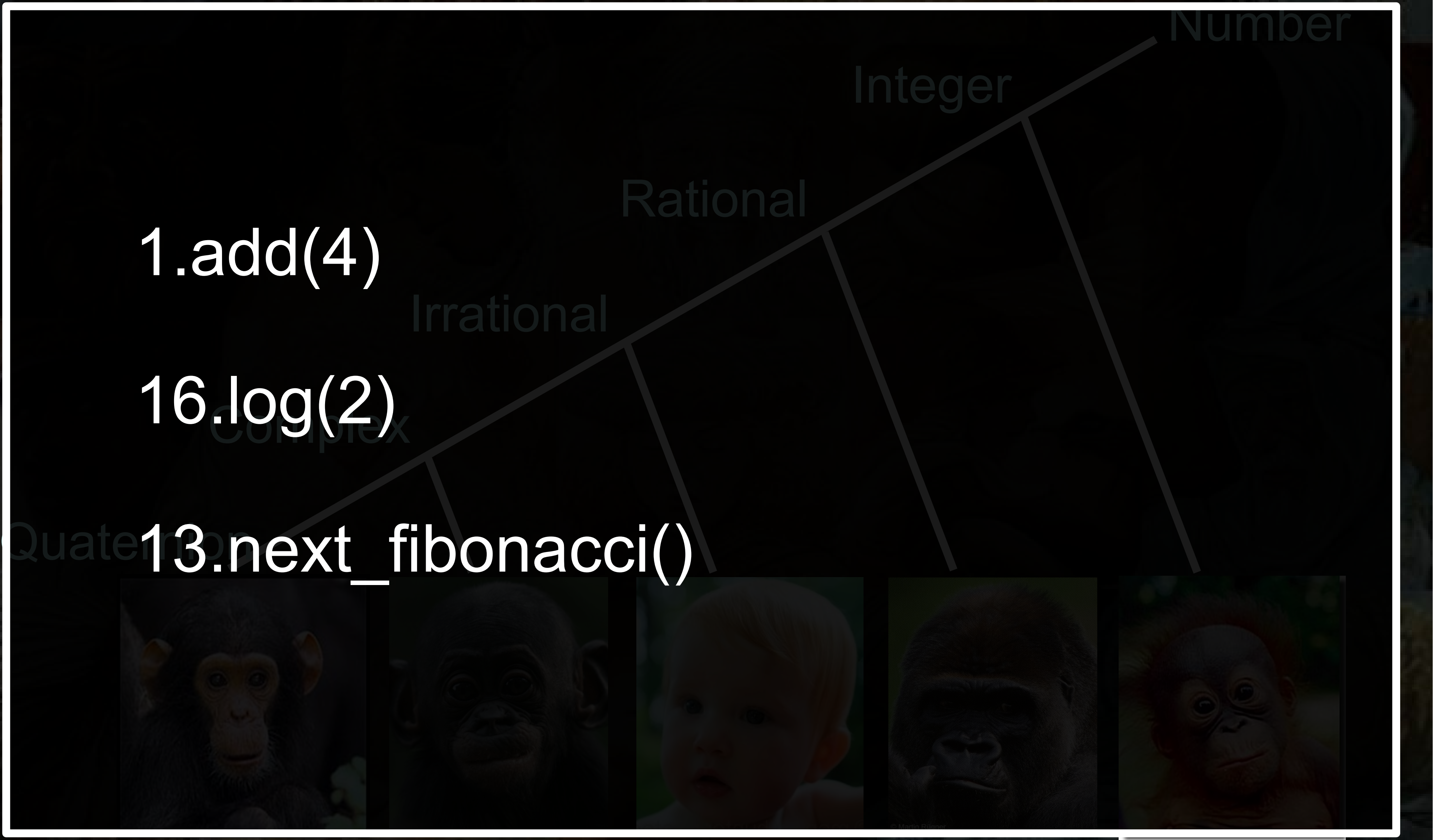
Quaternion

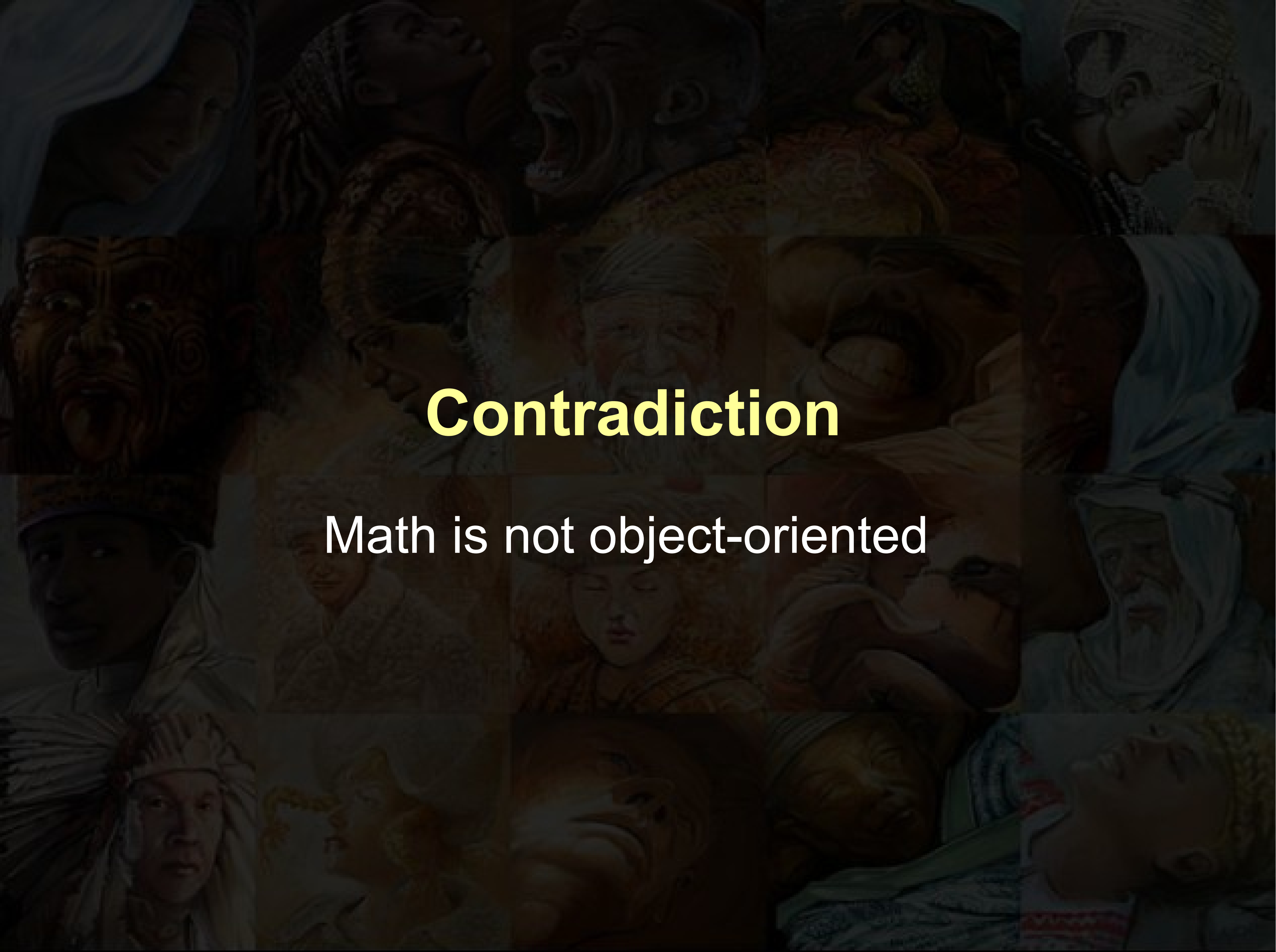
Classes In Math

```
1.add(4)

16.log(2)

13.next_fibonacci()
```

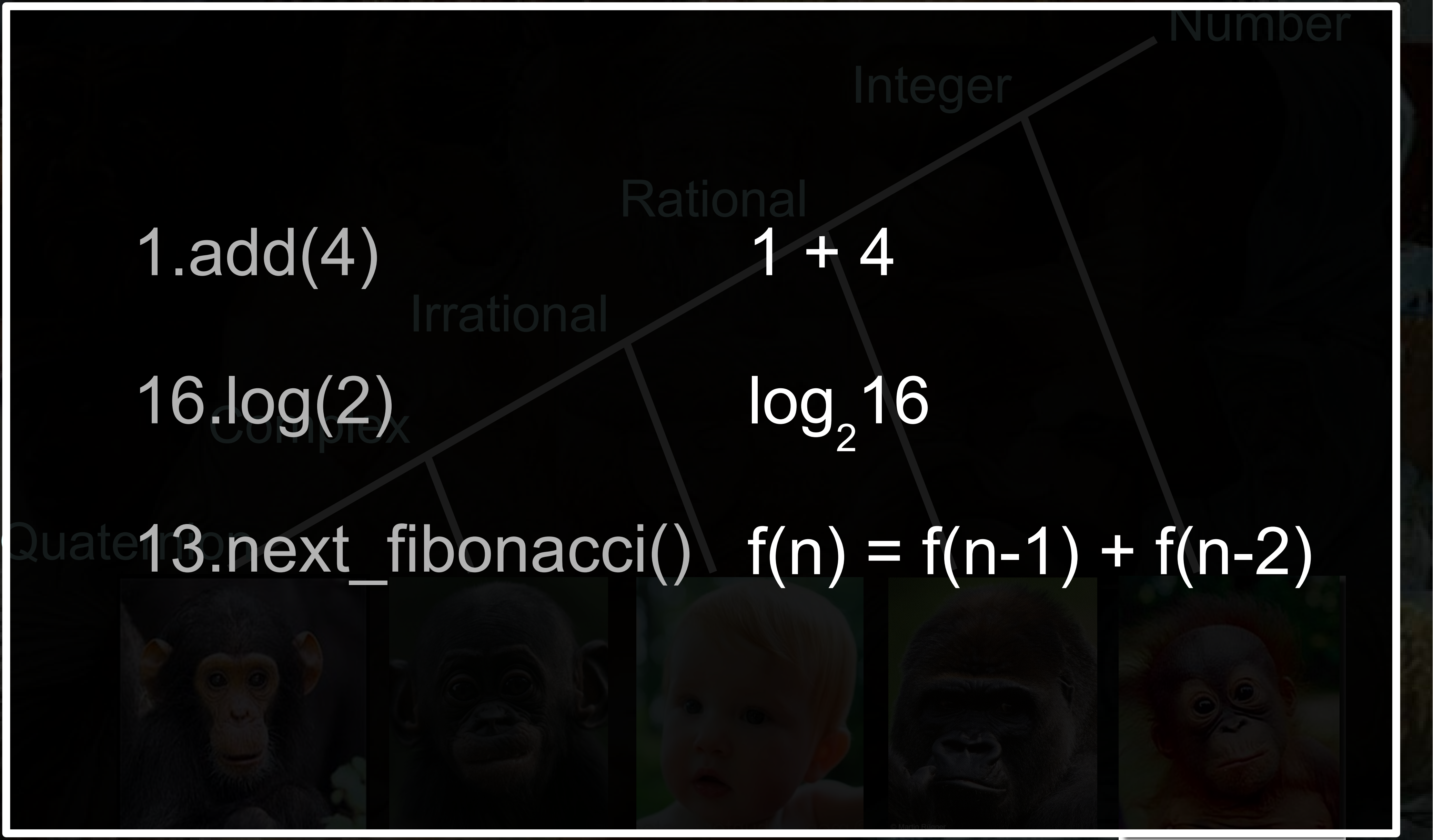




Contradiction

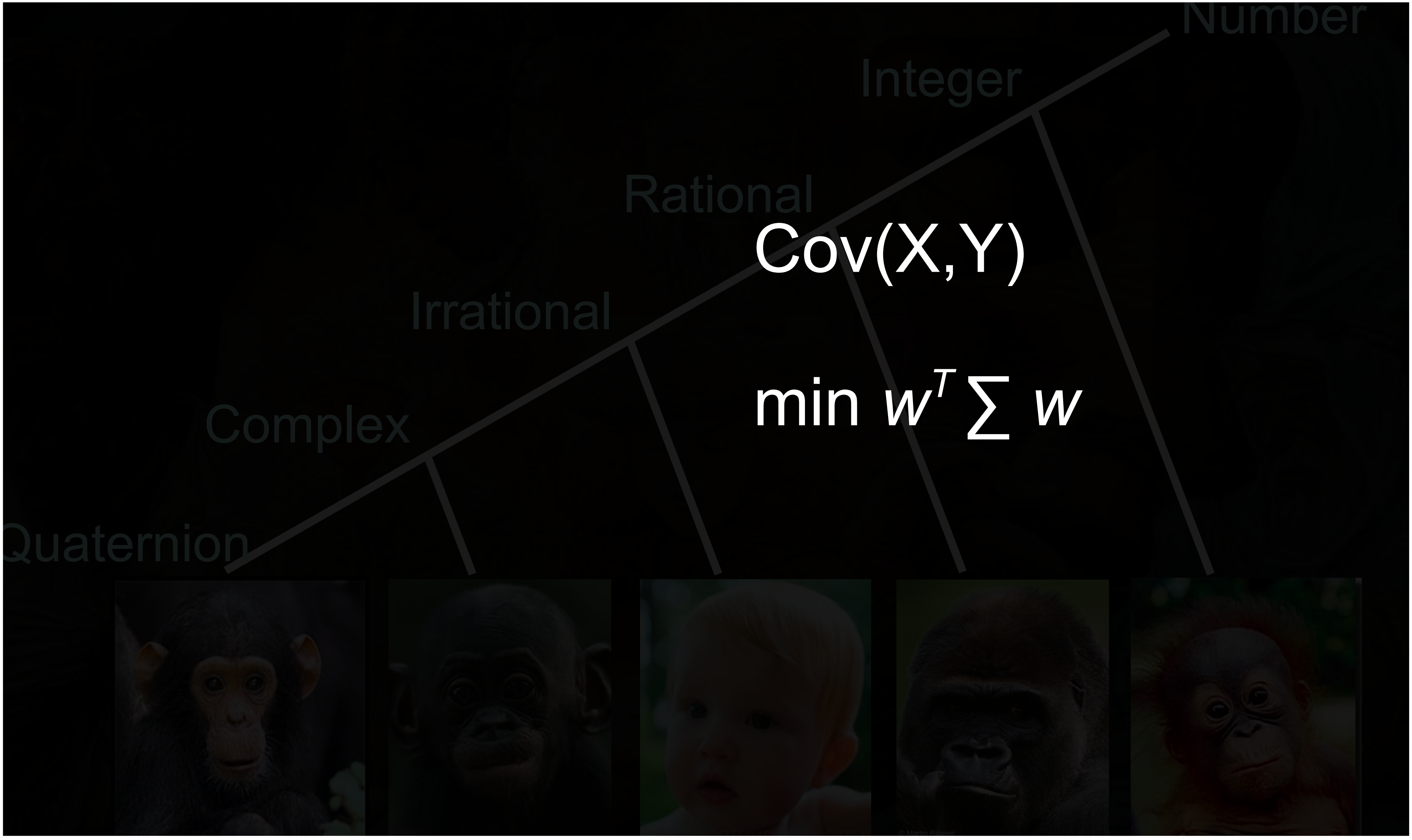
Math is not object-oriented

Math Is Functional



1.add(4)	$1 + 4$
16.log(2)	$\log_2 16$
13.next_fibonacci()	$f(n) = f(n-1) + f(n-2)$

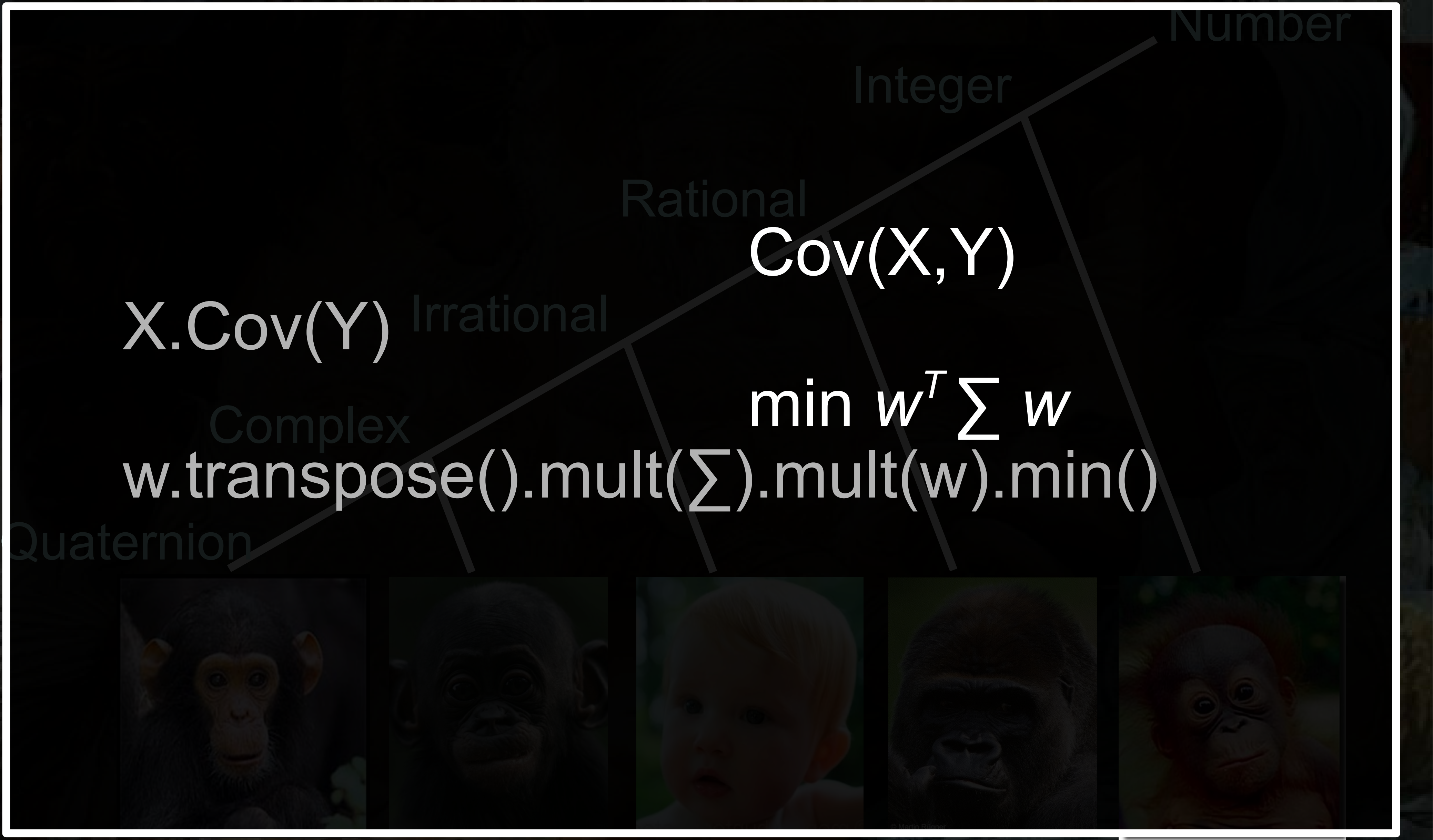
Finance Is Functional



Cov(X,Y)

$\min w^T \sum w$

Finance Is Functional



© Martin Rügner

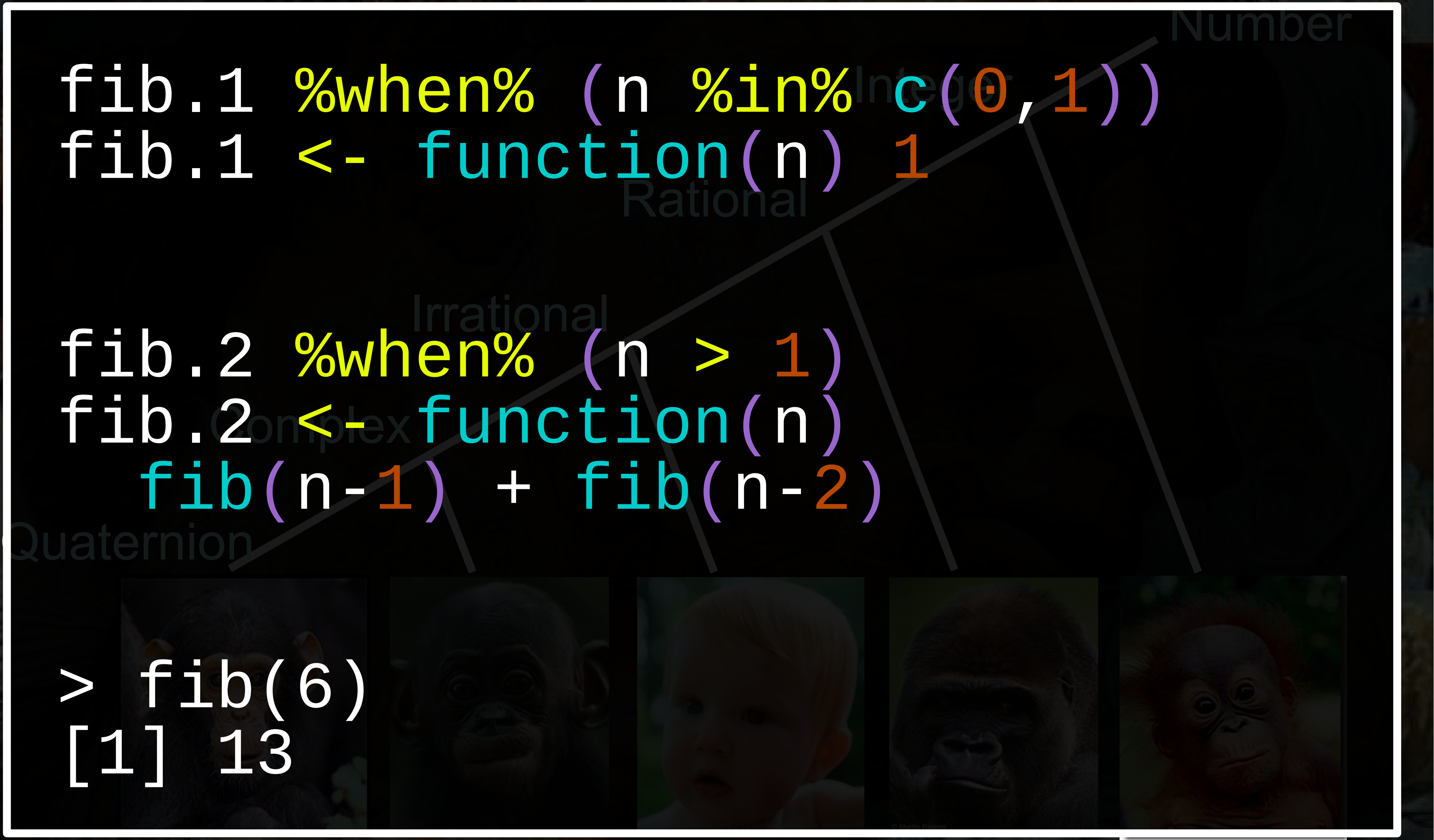
Cov(X,Y)

X.Cov(Y)

min $w^T \sum w$

w.transpose().mult(∑).mult(w).min()

R Is Functional



```
fib.1 %when% (n %in% c(0,1))
fib.1 <- function(n) 1


fib.2 %when% (n > 1)
fib.2 <- function(n)
  fib(n-1) + fib(n-2)


> fib(6)
[1] 13
```

R Is Functional

```
coupon.pct %when% (bond %isa% Bond &&
                   bond$coupon < 1)
coupon.pct <- function(bond)
  100 * bond$coupon / bond$freq

coupon.dv %when% (bond %isa% Bond)
coupon.dv %must% (result > 0)
coupon.dv <- function(bond)
  bond$coupon / bond$freq

> b <- create(Bond,coupon=.035,freq=2)
> coupon(b)
[1] 1.5
```

R Is Functional

```
install.packages("futile.paradigm")
```