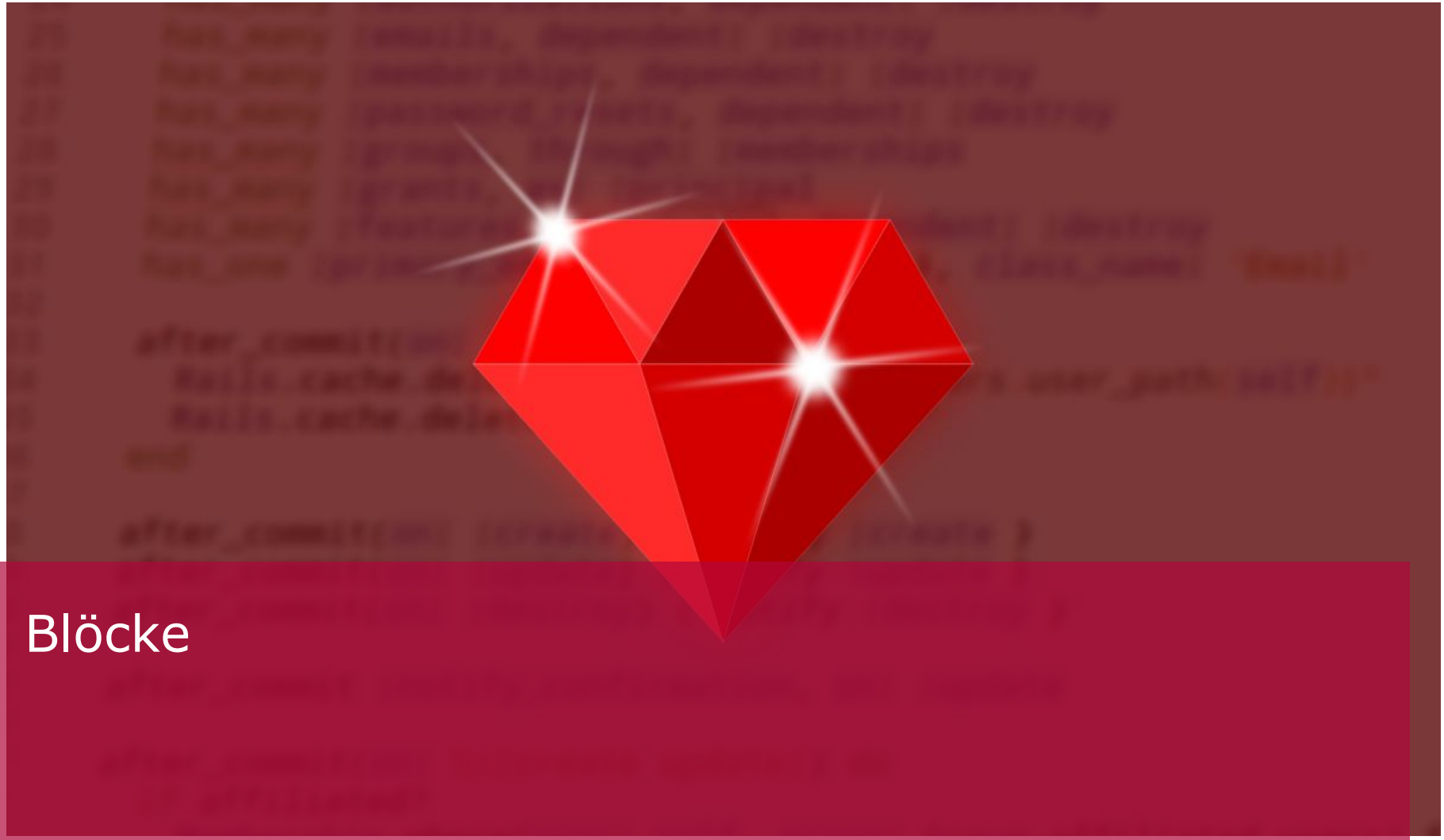


Blöcke

Blöcke

```ruby
[1, 2, 3].each do |i|
  puts i
end
# Gibt nacheinander 1, 2 und 3 aus
```

- Viele Methoden akzeptieren Blöcke und können diesen aufrufen
  - Methode implementiert Algorithmus / Prozedur
  - Aufrufender Code füllt beabsichtigte "**Lücken**" in der Prozedur aus

Der **Block** ist der Code zwischen `do … end`. Ein Block akzeptiert zusätzliche Parameter mit `|arg1, arg2|`.

Blöcke: Beispiele

Beispiel: Aus der Liste der Zahlen von eins bis fünf alle geraden Zahlen heraussuchen

```ruby
[1, 2, 3, 4, 5].select do |i|
  i.even?
end
# => [2, 4]
```

Oder in Kurzform:

```ruby
[1, 2, 3, 4, 5].select { |i| i.even? }
# => [2, 4]
```

Blöcke: Beispiele

Beispiel: Jede Zahl in einer Liste verdoppeln

```ruby
[1, 2, 3, 4, 5].map { |i| i * 2 }
# => [2, 4, 6, 8, 10]
```

Rückgabewert: eine neue Liste

```ruby
numbers = [1, 2, 3, 4, 5]
numbers.map { |i| i * 2 }
# => [2, 4, 6, 8, 10]

numbers
# => [1, 2, 3, 4, 5]
```

Blöcke: Beispiele

Beispiel: Alle Zahlen einer Liste nacheinander auf 0 aufsummieren

```ruby
[1, 2, 3, 4].reduce(0) { |sum, number| sum + number }
# => 10
```

Vergleich mit imperativen Sprachen

Hasso Plattner Institut

Ruby

```ruby
[1, 2, 3, 4].reduce(0) { |sum, number| sum + number }
# => 10
```

Java

```java
int sum = 0;
int[] list = {1, 2, 3, 4};

for (int i = 0; i < list.length; i++) {
    sum = sum + list[i];
}

return sum;
// => 10
```

Intermezzo: Die Reihe

Einfach Reihe von z.B. Zahlen, aber keine Liste!

Beispiel: Zahlenreihe von 1 bis inklusive 100

```ruby
1..100
# => 1..100
```

Beispiel: Alle Zahlen 1 bis 30 ausgeben

```ruby
(1..30).each do |i|
  puts i
end
# Gibt nacheinander 1, 2, 3, …, 30 aus
```

Blöcke: Beispiele

Erfüllen alle Elemente der Liste (oder mindestens eins) eine bestimmte Bedingung?

Beispiel: Sind alle Zahlen der Liste ungerade?

```
[1, 3, 5, 7, 9].all? { |i| i.odd? }
# => true
```

Beispiel: Ist zwischen 1 und 999 irgendeine gerade Zahl?

```
(1..999).any? { |i| i.even? }
# => true
```

Blöcke: Beispiele

Erfüllen alle Elemente der Liste (oder mindestens eins) eine bestimmte Bedingung?

Beispiel: Besteht einer der Namen aus mindestens fünf Zeichen?

```
["Jane", "James", "John"].any? { |name| name.length > 4 }
# => true
```

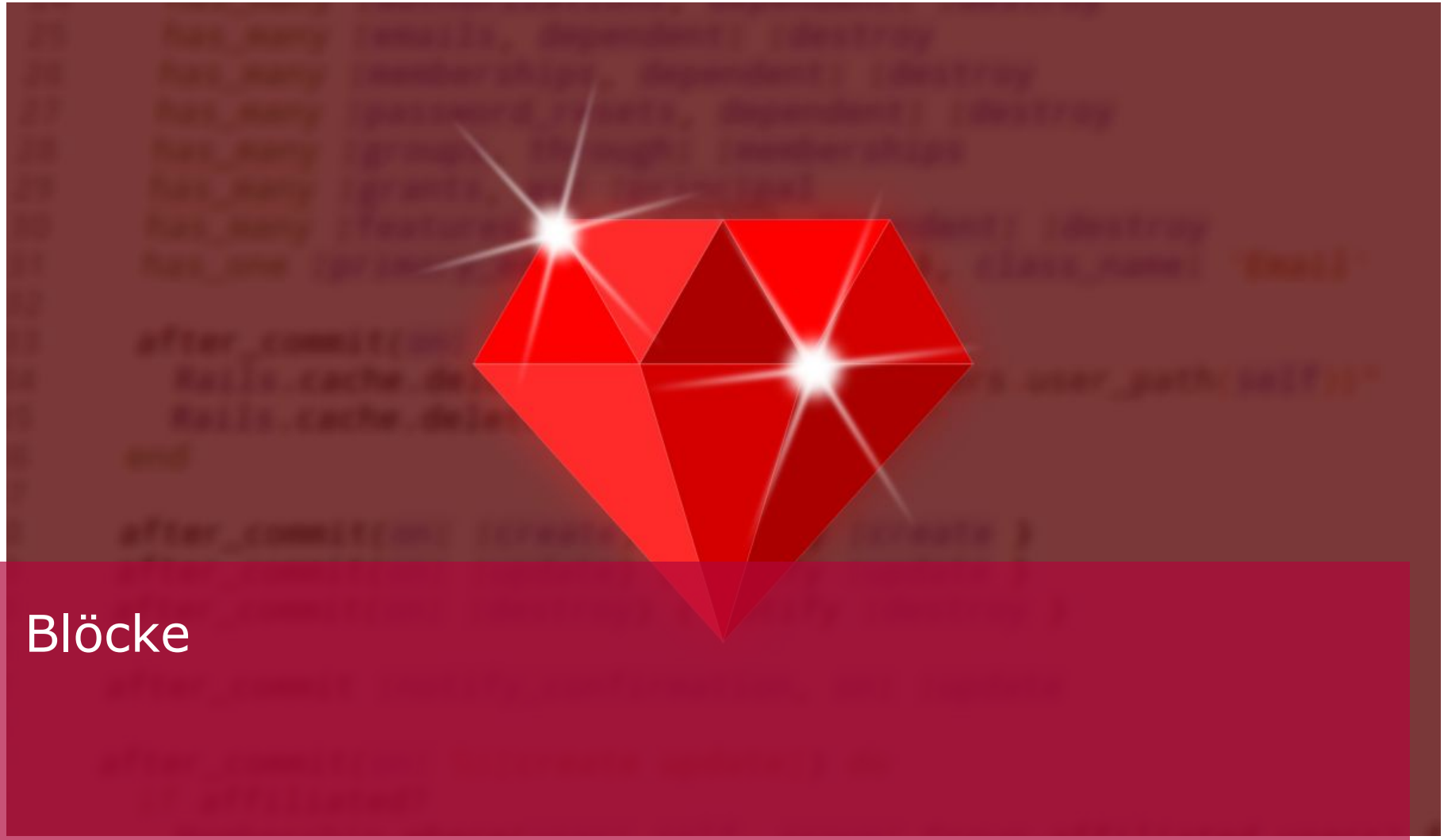
Beispiel: Fangen alle Namen mit “J” an?

```
["Jane", "James", "John"].all? { |name| name[0] == "J" }
# => true
```

Blöcke: Beispiele

Verketten von Methoden und Blöcken

```ruby
["John", "Jane", "Jonathan"]
  .select { |name| name.length < 5 }
  .map    { |name| "Hello #{name}!" }
  .each   { |message| puts message }
# Hello John!
# Hello Jane!
```



Blöcke