1.4: Blöcke

Methoden mit Block aufrufen

Alle Methoden akzeptieren einen Block und können diesen aufrufen. Der Block kann mittels `do ... end` oder mit `{ ... }` definiert werden.

```
(1..10).select do |n|
  n.odd?
end
```

Kurzform (Einzeiler):

```
(1..10).select { |n| n.odd? }
```

Block-Parameter

Blöcke können Argumente akzeptieren, dazu können gewünschte Parameter zwischen `|...|` aufgelistet werden.

Beispiel 1: Block mit einem Parameter

```
my_method { |arg1| ... }
```

Beispiel 2: Block mit mehreren Parametern

```
my_method { |arg1, arg2| ... }
```

Block in Methoden verwenden

Mit dem Schlüsselwort yield kann ein übergebener Block in einer Methode aufgerufen werden. An yield übergebene Argumente werden an den Block übergeben.

Beispiel 1: Ruft einen übergebenen Block einmal auf und gibt Ergebnis zurück, da yield die letzte Anweisung im Methodenrumpf ist.

```
def my_method1
  yield
end
```

Beispiel 2: Ruft einen Block zweimal auf.

```
def my_method2
  yield
  yield
end
```

Beispiel 3: Ruft einen Block mit mehreren Argumenten auf.

```
def my_method3
  yield 1, 2
end
```

Verwendungsbeispiel:

```
my_method3 { |a, b| a + b }
# => 3
```