



Hashes

```

h = {:key1 => value1, :key2 => value2 }
h.empty? # => false
h.length # => 2

h[:key1]= value3
h[:key1] # => value3
h[:key3] # => nil

h.each do |s, w|
  puts "#{s} => #{w}"
end

```

Arrays

```

a = [21, 22, 33, 44, 55 ]
a.length # => 5

a[0] = 11
a[0] # => 11
a[5] # => nil
a[-1] # => 55
a[1,2] # => [ 22, 33 ]
a[1..3] # => [ 22, 33, 44 ]
a[1...3] # => [ 22, 33 ]

a.each do |item|
  puts item
end

```

Code-Blöcke

```

def op(a, b)
  yield a, b
end

op(42,24) { |x,y| x + y } # 66
op(42,24) { |x,y| x - y } # 18

def method(a, &block)
  b = block.call(a)
  # b => 43
end

method(42) { |x| x +=1 }

```

Ausnahmen

```

def parse(file)
  ...
  raise IOError, "Datei leer"
  ...
end

...
begin
  parse(file)
rescue IOError => io_error
  ...
rescue => err
  puts err.message
  puts err.backtrace
ensure
  file.close
end

```

Schleifen

```

for i in 0..10 do
  puts i # 0,1,2,...,9
end

for i in 1..10 do
  puts i # 1,2,3,...,10
end

while bedingung do ... end

until bedingung do ... end

```

Bedingungen

```

if bedingung1
  ...
elsif bedingung2
  ...
else
  ...
end

if !bedingung ...
unless bedingung ...

a = 42 if b > 0
a = 42 unless b <= 0

case a
when 42: puts "yipee!"
when 24: puts "buh!"
else puts "hee?"
end

```

Symbole

```

"name".object_id # 443352
"name".object_id # 443332

:name.object_id # 880910
:name.object_id # 880910

hash[:key] = 42
has_many :books

```

Symbole sind im Gegensatz zu Strings atomar und werden daher überall dort verwendet, wo keine neue Instanz benötigt wird, z.B. als Schlüssel in Hash oder bei der Angabe von Namen.