



Attribute? Methoden!

Attribute

Attribute

- Speichern Zustand im Objekt
- Sind nicht direkt von außerhalb erreichbar

Konvention

- Name: Kleinbuchstaben mit Unterstrich zwischen Worten
- Zugriff immer über Methoden, z.B.:

#attribute und #attribute=(value)

```
def first_name
  @first_name
end

def first_name=(val)
  @first_name = val
end
```

```
obj.first_name
# => "Peter"

obj.first_name = "Tom"
obj.first_name="Tom"
# => "Tom"
```

Attribute

- Methoden für Zugriff auf Attribute generieren

```
attr_reader :name
```

```
attr_writer :name
```

```
# Reader und Writer  
attr_accessor :name
```

```
# Mehrere Reader  
attr_reader :name, :age
```

```
# Alias für attr_reader  
attr :name, :age
```

```
def name  
  @name  
end
```

```
def name=(new_name)  
  @name = new_name  
end
```

Konstanten

Konstanten

- Speichern von sich nicht ändernden Werten
- In Klassen oder Modulen

Konvention

- Bezeichner vollständig in Großbuchstaben

```
class Task
  MAX_DURATION = 120
end

Task::MAX_DURATION # => 120
Math::PI           # => 3.141592653589793
```

Methoden

Objektmethoden

- Namenskonvention: Kleinbuchstaben mit Unterstrich

```
class Task
  def description
    "An awful lot of work"
  end

  def short_description
    description[0..11]
  end
end

Task.new.short_description
# => "An awful lot"
```

Methoden

Klassenmethoden

- Methoden auf dem Klassenobjekt

```
class Task
  def self.max_working_hours
    8
  end

  def overtime?
    @duration > self.class.max_working_hours
  end
end
```

Definition von
Klassenmethoden mit
def self.name

Aufruf von Klassenmethoden
mit self.class.name



Attribute? Methoden!