



Dateisystem

# Ein- und Ausgabe und Dateien

---

## **IO**

- Gemeinsame Methoden zur Ein- und Ausgabe verschiedener Medien, z.B. Konsole, Dateien, Netzwerk

## **File**

- Entspricht einer geöffneten Datei
- Erbt von IO

## **FileUtils**

- Modul mit nützlichen Operationen auf dem Dateisystem
- Gehört zur Standardbibliothek

# Dateien lesen

- Datei zum Lesen öffnen:

```
file = File.new "numbers.txt", "r"
file.read      # => "1 2 3"

# Zum Anfang der Datei zurückgehen
file.rewind
file.read(3)   # => "1 2"

# Datei schließen
file.close
```

# Dateien lesen

---

- Kürzer mit Block

```
File.open "numbers.txt", "r" do |file|  
  file.read # => "1 2 3"  
end  
# Datei wird automatisch geschlossen
```

- Kompletten Inhalt lesen

```
File.read "numbers.txt" # => "1 2 3"
```

# Dateien lesen

---

- Zeilen für Zeile lesen:

```
File.open "wisdom.txt", "r" do |file|  
  file.readlines  
  # => ["Money makes the world go round\n",  
  #     "An apple a day keeps the doctor away\n"]  
end
```

# Dateien lesen

- Aufzählbar wie eine Liste

```
File.open "wisdom.txt", "r" do |file|  
  file.each do |line|  
    puts line  
  end  
end  
# Money makes the world go round  
# An apple a day keeps the doctor away
```

- IO inkludiert auch Enumerable:  
#map, #select, ...
- ⇒ <https://ruby-doc.org/core-2.5.1/IO.html>

# Dateien schreiben

- Datei zum Schreiben öffnen:

```
File.open "numbers.txt", "w" do |file|  
  file.write "4 5 6"  
end  
  
# Zeile für Zeile schreiben  
File.open "numbers.txt", "w" do |file|  
  file.puts "4", "5"  
  file.puts "6"  
end
```

# Intermezzo: puts?

---

## **IO#puts**

- Schreibt Argumente auf je eine Zeile

## **STDOUT.puts**

- Standardausgabe ist ein Objekt von IO

Äquivalente Beispiele:

```
puts "Hello Ruby"
```

```
# Hello Ruby
```

```
# => nil
```

```
STDOUT.puts "Hello Ruby"
```

```
# Hello Ruby
```

```
# => nil
```



# Dateien schreiben

---

- Modus gibt an wie eine Datei geöffnet werden soll

```
File::new(filename, mode="r" [, opt]) → file
```

⇒ <https://ruby-doc.org/core-2.5.1/File.html#method-c-new>

**r** : Nur lesend

**w** : Überschreibt ganze Datei

**a** : Schreib ans Ende der Datei und hängt an

# Dateien schreiben

---

- Einfach kompletten Inhalt schreiben

```
File.write "numbers.txt", "1 2 3 4"
```

# File im Detail

---

```
file = File.new "numbers.txt", "r"
file.read # => "4 5 6"

# Datum der letzten Änderung
file.mtime # => 2018-09-13 09:19:49 +0200

# Dateigröße in Bytes
file.size # => 5
```

⇒ <https://ruby-doc.org/core-2.5.1/File.html>



Dateisystem