



# Informatik für Schüler, Foliensatz 7

## Dateien und Textverarbeitung

Prof. G. Kemnitz

Institut für Informatik, Technische Universität Clausthal  
3. Februar 2010



## Datei lesen

- Erzeugen eines Dateiobjekts:

```
infile = open('date.txt', 'r')
```

'r' steht für lesen; alternativ 'rb' um auch nicht druckbare Zeichen lesen zu können

- gesamte Datei lesen und in einer Zeichenkette speichern

```
z = infile.read()
```

z – Zeichenkette

- Nach Benutzung ist eine Datei wieder zu schließen:

```
infile.close()
```



## Experiment

- Mit dem Editor eine Textdatei test.txt mit mehreren Textzeilen erzeugen; im Interpreter-Modus erzeugte Textdatei öffnen, lesen, anzeigen und ausgeben:

```
f = open('test.txt', 'r')
z = f.read()
f.close()
z
print z
```



## Weitere Leseoperationen

- maximal  $n$  Zeichen lesen

```
z = infile.read(n)
```

- ein Zeile lesen

```
z = infile.readline()
```

- Datei als Liste von Zeilen lesen

```
z = infile.readlines()
```



## Experiment

Datei als Liste von Zeilen lesen und in einer Schleife zeilenweise mit vorangestellter Zeilennummer ausgeben (Programm mit dem Editor schreiben und testen):

```
f = open('test.txt', 'r')
z = f.readlines()
f.close()
nr=1
for l in z:
    print 'Nr.: ', nr, ': ', l
    nr = nr +1
```

## Nicht druckbare Zeichen

ASCII-Zeichensatz (Standardzeichensatz, den jeder Rechner unterstützt)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SPC	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	:	;	<	=	>	?
4	@	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>	<b>I</b>	<b>J</b>	<b>K</b>	<b>L</b>	<b>M</b>	<b>N</b>	<b>O</b>
5	<b>P</b>	<b>Q</b>	<b>R</b>	<b>S</b>	<b>T</b>	<b>U</b>	<b>V</b>	<b>W</b>	<b>X</b>	<b>Y</b>	<b>Z</b>	[	\	]	^	_
6	`	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>f</b>	<b>g</b>	<b>h</b>	<b>i</b>	<b>j</b>	<b>k</b>	<b>l</b>	<b>m</b>	<b>n</b>	<b>o</b>
7	<b>p</b>	<b>q</b>	<b>r</b>	<b>s</b>	<b>t</b>	<b>u</b>	<b>v</b>	<b>w</b>	<b>x</b>	<b>y</b>	<b>z</b>	{		}	~	DEL



Nicht druckbare Zeichen werden in einer Zeichenkette durch Escape-Codes dargestellt:

- `\\` Backslash
- `\ddd` Darstellung durch eine Dezimalzahl im Bereich von 0 bis 255
- `\xhh` Darstellung durch zwei Hex.-Ziffern
- `\n` Zeilenumbruch
- `\t` Tabulator
- `\r` Wagenrücklauf
- `\v` Zeilevorschub
- `\a` Glocke (Peep-Ton)



## Experimente

- Ausprobieren der Escape-Codes `'\n'`, `'\t'`, `'\r'`, `'\v'` und `'\a'`:

```
print 'vor\d...nach'
```

- Welche Zeichen werden mit den hexadezimalen Zeichencodes `x7f` und `x85` dargestellt? Welchen dezimalen Werte haben diese beiden Zeichencodes?

```
print '\x...'
```

- 
- Mit den Editor eine Textdatei `test.txt` mit Zeilenumbrüchen und Tabulatoren erzeugen; im Interpreter-Modus mit `'rb'` öffnen, lesen, anzeigen, ausgeben:

```
f = open('test.txt', 'rb')
z = f.read()
f.close()
print z
```





## In eine Datei schreiben

- Erzeugen des Dateiobjekts:

```
outfile = open('date.txt', 'w')
```

'w' steht für schreiben; alternativ 'wb' schreiben binärerer, d.h. auch nicht druckbare Zeichen

- Ausgabe einer Zeichenkette

```
outfile.write(z)
```

z – Zeichenkette

- Nach Benutzung ist eine Datei wieder zu schließen:

```
outfile.close()
```



## Aufgabe 7.1: Zeichen und Zeilen einer Datei zählen

Schreiben Sie ein Programm, das

- zur Eingabe des Namens einer Textdatei auffordert
- die Textdatei einliest
- die Anzahl der Zeichen und die Anzahl der Zeilen (bzw. Zeilenumbrüche) zählt und
- beide Werte mit »print« ausgibt.

Testen Sie das Programm mit den beiden System-Dateien:

- /etc/fstab (Konfigurationsdatei für das Dateisystem)
- /etc/X11/xorg.conf (Konfigurationsdatei für Tastatur, Mause, Bildschirme etc.).



## Aufgabe 7.2: Liedtext generieren

Laden Sie sich die Textdatei »Kontrabass.txt« mit dem Text der ersten Strophe des Scherz-Liedes »Drei Chinesen mit 'nem Kontrabass, ...« herunter. Schreiben Sie ein Programm, das daraus die nächsten fünf Strophen generiert:

- Strophe 2: Ersetzen aller Selbstlaute durch »a«
- Strophe 3: Ersetzen aller Selbstlaute durch »e«
- Strophe 4: Ersetzen aller Selbstlaute durch »i«
- Strophe 5: Ersetzen aller Selbstlaute durch »o«
- Strophe 6: Ersetzen aller Selbstlaute durch »u«

Das Programm soll alle sechs Strophen hintereinander getrennt durch eine Leerzeile in eine Datei »KontrabassAll.txt« schreiben.



## Aufgabe 7.3: Wort in einer Datei suchen

Schreiben Sie ein Programm, das die Datei »/etc/fstab« des Rechners einliest und testet, ob sie die Zeichenkette »cdrom« enthält. Wenn ja, sollen alle Zeilen mit der vorangestellten Zeilennummer ausgegeben werden, die diese Zeichenkette enthalten.

### Lösungshinweise

- Datei zeilenweise lesen und Zeilennummer mitzählen.
- Für jede Zeile für jede Position  $i$  von Null bis vier Zeichen vor dem Ende den Teilbereich  $z[i:i+5]$  mit dem Suchbegriff »cdrom« vergleichen.
- Bei mehrfach enthaltenem Suchbegriff Zeile nur einmal ausgeben.



## Aufgabe 7.4: Textformatierung

Schreiben Sie ein Programm, das

- eine Textdatei (z.B. ein altes Programm) in eine Zeichenkettenvariable einliest
- die vorhandenen Zeilenumbrüche und Tabulatoren durch Leerzeichen ersetzt
- Folgen aus mehreren Leerzeichen durch ein einzelnes Leerzeichen ersetzt
- nach jeweils 20 darstellbaren Zeichen das nächste Leerzeichen durch einen Zeilenumbruch ersetzt
- das Ergebnis auf den Bildschirm anzeigt und
- in eine Datei »Ausgabe.txt« ausgibt.

Kontrollieren Sie die Ausgabedatei mit dem Editor.



## Hinweise

- Eingabedatei im 'rb'-Modos öffnen; der Name der zu öffnenden Datei kann vorher in einer Eingabeanweisung erfragt werden
- Verarbeitungsprinzip:

```
Ergebnis = leere_Zeichenkette
```

```
for Zeichen in Zeichenkette:
```

```
    if Ersetzungsbedingung:
```

```
        Ergebnis = Ergebnis + Ersatzzeichen
```

```
    else:
```

```
        Ergebnis = Ergebnis + Zeichen
```



## Aufgabe 7.5: Akustische Fehlermeldung

Schreiben Sie ein Programm, das nacheinander fünf Zahlen im Bereich vom null bis zehn anfordert und die eingegebenen Zahlen getrennt durch Tabulatoren in eine neue Datei »Zahlen.txt« schreibt. Bei einem unzulässigen Eingabewert ist eine akustische Fehlermeldung (Beep) auszugeben und eine neue Eingabe anzufordern.

### Hinweise:

- Die Anforderung der fünf Zahlen erfordert eine Wiederholschleife.
- »bis korrekte Eingabe« ist mit einer Abbruchschleife zu beschreiben.
- Die akustische Fehlermeldung ist die nicht druckbare Ausgabe:

```
print '\a'
```