

**Aufgabe 9.1**

```
s = '''Das ist  
ein Text'''  
print(s)
```

Das ist  
ein Text

**Aufgabe 9.2**

```
s = 'Das ist \nWahnsinn'  
print(s)
```

Das ist  
Wahnsinn

**Aufgabe 9.3**

```
s = 'C\'est la vie!'  
print(s)
```

C'est la vie!

**Aufgabe 9.4**

```
s = 'A\\B'  
print(s)
```

A\\B

Da der Backslash zum Maskieren der Stringbegrenzungszeichen ("..." und '...') sowie für die Bildung von Steuerzeichen (`\n`) verwendet wird, kann er nicht direkt in einer Zeichenkette auftreten und muss seinerseits durch einen zweiten Backslash maskiert werden.

**Aufgabe 9.5**

```
s = 'Hund'  
print(s[1:])
```

und

**Aufgabe 9.6**

```
print('A' + 2*'B' + 'A')
```

ABBA

### Aufgabe 9.7

```
s = 'Was ist das?'  
print(len(s))
```

12

Jedes Zeichen (auch Leerzeichen, Zeichenschaltungen und Tabuloren) wird gezählt.

### Aufgabe 9.8

```
s = "hallo"  
print(list(s))
```

```
['h', 'a', 'l', 'l', 'o']
```

Die `list()`-Methode zerlegt eine Zeichenkette in eine Liste von Einzelzeichen (*characters*).

### Aufgabe 9.9

```
print(ord('A'))
```

65

An Prüfungen zu diesem Thema steht eine ASCII-Tabelle zur Verfügung, so dass der Wert (die „Ordnungszahl“ des Zeichens) dort abgelesen werden kann.

### Aufgabe 9.10

```
print(chr(66))
```

B

### Aufgabe 9.11

```
s = "Ananas"  
print(s.count('a'))
```

2

Die String-Methode `str.count(<zeichenkette>)` zählt, wie oft `<zeichenkette>` in `str` vorkommt. Man beachte, dass Gross- und Kleinschreibung unterschieden wird.

### Aufgabe 9.12

```
s = 'Hallo'
s = s.lower()
print(s)
```

hallo

### Aufgabe 9.13

```
s = 'ch'
s.upper()
print(s)
```

ch

*Achtung:* Da Zeichenketten unveränderlich (*immutable*) sind, können die nicht durch die String-Methoden verändert werden. Dafür liefern die Methoden einen Rückgabewert und es liegt in der Verantwortung des Programmierers, diesen Rückgabewert in einer Variablen zu speichern.

### Aufgabe 9.14

```
s = 'abcde'
s = s.replace('a', 'b')
print(s)
```

bbcde

Die Methode `str.replace()` ersetzt die Zeichenkette im ersten Parameter durch die Zeichenkette im zweiten Parameter. Man kann die Anzahl der Ersetzungen durch einen dritten Parameter beschränken aber das ist kein Prüfungstoff.

### Aufgabe 9.15

```
L = ['x', 'y', 'z']
s = '+'.join(L)
print(s)
```

x+y+z

### Aufgabe 9.16

```
s = 'teller'
print(s.split('e'))
```

['t', 'll', 'r']

### Aufgabe 9.17

```
s = 'abcdxyz'  
s = s.strip('abyz')  
print(s)
```

cdx

`str.strip(<zeichenkette>)` entfernt links und rechts von `str` die in `<zeichenkette>` vorkommenden Zeichen. Die Reihenfolge ist dabei unwichtig. Fehlt der Parameter, so werden automatisch alle Formen von „Whitspaces“ (Leerzeichen, Tabulatoren, Zeilenschaltungen) entfernt.

`str.lstrip(<zeichenkette>)` und `str.rstrip(<zeichenkette>)` funktionieren analog, nur dass sie auf jeweils einer Seite (*left*, *right*) wirken.

### Aufgabe 9.18

```
s = 'abcxyz'  
s = s.lstrip('abyz')  
print(s)
```

cxyz

### Aufgabe 9.19

```
s = 'abcxyz'  
s = s.rstrip('abyz')  
print(s)
```

abcx

### Aufgabe 9.20

```
s = 'a{}pb{}m'.format(3, 'e')  
print(s)
```

a3pbem

### Aufgabe 9.21

```
s = 't{1}k{0}w{1}'.format(3, 'e')  
print(s)
```

tek3we

### Aufgabe 9.22

```
print(int("15" + "4") + 3)
```

157

## Aufgabe 9.23

18.0

## Aufgabe 9.24

2