

1. (4 Punkte) Was erscheint auf der Konsole? Schreibe error für einen Fehler und ein Minuszeichen für eine Leerzeile.

```
s = 'abcaab'
print(s.capitalize())
print(s.index('b'))
print(s.find('s'))
print(s.count('a'))
```

Lösung:

```
Abcaab
1
-1
3
```

2. (4 Punkte) Was erscheint auf der Konsole? Schreibe error für einen Fehler und ein Minuszeichen für eine Leerzeile.

```
s = 'cDAaBAA'
print(s.swapcase())
print(s.index('b'))
print(s.find('A'))
print(s.count('a'))
```

Lösung:

```
CdaDAbaa
error
2
1
```

3. (2 Punkte) Was erscheint auf der Konsole?

```
s = 'acBBa'
a = s.replace('a','c').capitalize().swapcase().replace('C','1')
print(a)
```

Lösung:

```
c1BB1
```

4. (2 Punkte) Was erscheint auf der Konsole?

```
s = 'dEbbA'
a = s.capitalize().swapcase().replace('b','c')
print(a)
```

Lösung:

```
dEBBA
```

5. (2 Punkte) Was erscheint auf der Konsole? Schreibe error für einen Fehler.

```
a = '12ab'
print(a.isalpha() or a.isdigit())
b = '1234'
print(b.isdigit() and b.isalnum())
```

Lösung:

```
False
True
```

6. (2 Punkte) Was erscheint auf der Konsole? Schreibe error für einen Fehler.

```
a = 125
print(a.isalpha() or a.isdigit())
b = 'abc'
print(b.isdigit() or b.isalnum())
```

Lösung:

```
error
True
```

7. (2 Punkte) Ergänze den fehlenden Format-String

```
s1, x1 = 'Alice', 5.7876
s2, x2 = 'Bob', 14.4421
f =
print(f.format(s1,x1))
print(f.format(s2,x2))
```

Ausgabe: (blanks sind fürs Abzählen durch Punkte ersetzt)
Da ist .Alice ... mit .der .Zahl ...5.79
Da ist .Bob mit .der .Zahl ..14.44

Lösung:

```
f = 'Da ist {:7} mit der Zahl {:.2f}'
```

8. (2 Punkte) Ergänze den fehlenden Format-String

```
s1, x1 = 'Alice', 5.7876
s2, x2 = 'Bob', 14.4421
f =
print(f.format(s1,x1))
print(f.format(s2,x2))
```

Ausgabe: (blanks sind fürs Abzählen durch Punkte ersetzt)
!..... Alice!.5.788...!
!..... Bob!.14.442..!

Lösung:

```
f = '! {:>10} ! {:<7.3f} !'
```

9. (3 Punkte) Implementiere die Funktion func.

```
def func(s, teil):
    """
    s, teil: Strings
    returns True, wenn s mit teil beginnt oder endet und die Länge
    von s nicht größer ist als die doppelte Länge von teil

    Beispiele:
    >>> func('abc', 'ab')
    True
    >>> func('abc', 'abcccc')
    False
    """
```

Lösung:

```
def func(s, teil):  
    return (s.startswith(teil) or s.endswith(teil)) and not len(s) > 2 * len(teil)
```

10. (3 Punkte) Implementiere die Funktion func.

```
def func(s):  
    '''  
    s: String  
    returns True, wenn s mindestens die Länge 5 hat und die ersten beiden  
        Zeichen alphanumerisch und die letzten beiden Zeichen Ziffern sind.  
  
    Beispiele:  
>>> func('R2be-09')  
    True  
>>> func('A?2b09')  
    False  
    '''
```

Lösung:

```
def func(s):  
    return len(s) >= 5 and s[:2].isalnum() and s[-2:].isdigit()
```