1. (3 Punkte) Die Hilfe zu `str.replace` lautet:

```
replace(self, old, new, count=-1, /)
    Return a copy with all occurrences of substring old replaced by new.
      count
        Maximum number of occurrences to replace.
        -1 (the default value) means replace all occurrences.
```

Was erscheint auf der Konsole?

```
s = 'banane'
print(s.replace('an', 'xy', 1))
print(s.replace('a', ''))
print(s)
```

Lösung:

```
bxyane
bnne
banane
```

2. (3 Punkte) Die Hilfe zu zwei String-Methoden lautet:

```
center(self, width, fillchar=' ', /)
    Return a centered string of length width.
    Padding is done using the specified fill character (default is a space).

zfill(self, width, /)
    Pad a numeric string with zeros on the left, to fill a field of the given width.
    The string is never truncated.
```

Was erscheint auf der Konsole?

```
print('Hi'.center(6, '*'))
print('7'.zfill(3))
print('1234'.zfill(3))
```

Lösung:

```
**Hi**
007
1234
```

3. (4 Punkte) Die Hilfe zu `sorted` beginnt so:

```
sorted(iterable, /, *, key=None, reverse=False)
    Return a new list containing all items from the iterable in ascending order.
```

Welche der folgenden Aufrufe sind für die Liste `a = [3, -1, 2]` erlaubt? Begründe die Fälle, die einen Fehler ergeben.

```
sorted(a, reverse=True)      # (1)
sorted(a, True)              # (2)
sorted(iterable=a)           # (3)
sorted(a, key=abs)           # (4)
```

Lösung: Erlaubt sind (1) und (4).
(2): `reverse` steht hinter dem `*` und muss mit Namen übergeben werden.
(3): `iterable` steht vor dem `/` und darf nur über die Position übergeben werden.

4. (3 Punkte) Die Hilfe zu `print` beginnt so:

```
print(*args, sep=' ', end='\n', file=None, flush=False)
    Prints the values to a stream, or to sys.stdout by default.
    sep
      string inserted between values, default a space.
    end
      string appended after the last value, default a newline.
```

   (a) (2 Punkte) Was erscheint auf der Konsole?

```
print(1, 2, 3, sep='-', end='!')
print('x')
```

   (b) (1 Punkt) Schreibe einen Aufruf von `print` mit den drei Argumenten `'a'`, `'b'`, `'c'`, der `a:b:c` ausgibt.

Lösung:

```
a) 1-2-3!x
b) print('a', 'b', 'c', sep=':')
```

5. (3 Punkte) Die Hilfe zu `int` enthält den folgenden Text:

```
int(x, base=10) -> integer
Convert a number or string to an integer, or return 0 if no arguments
are given.  If x is a number, return x.__int__().  For floating-point
numbers, this truncates towards zero.
```

Was erscheint auf der Konsole?

```
print(int('101', 2))
print(int('ff', 16))
print(int(3.9), int(-3.9))
```

Lösung:

```
5
255
3 -3
```

6. (4 Punkte) Die Hilfe zu zwei Methoden lautet:

```
sort(self, /, *, key=None, reverse=False)
    Sort the list in ascending order and return None.

shuffle(x)
    Shuffle list x in place, and return None.
```

   (a) (2 Punkte) Was erscheint auf der Konsole?

```
a = [3, 1, 2]
b = a.sort()
print(a, b)
```

   (b) (1 Punkt) Was bedeutet `in place`?

   (c) (1 Punkt) Eine Liste `karten` soll gemischt werden. Warum ist `karten = random.shuffle(karten)` falsch?

Lösung: a) `[1, 2, 3] None`
b) Die Liste selbst wird verändert, es entsteht keine neue Liste.
c) `shuffle` gibt `None` zurück, `karten` wäre danach `None`. Richtig ist `random.shuffle(karten)`.

7. (3 Punkte) Die Hilfe zu `random.sample` beginnt so:

```
sample(population, k, *, counts=None)
    Chooses k unique random elements from a population sequence.
    Returns a new list containing elements from the population while
    leaving the original population unchanged.
```

Schreibe ein Programm, das sechs verschiedene Lottozahlen von 1 bis 49 aufsteigend sortiert ausgibt.

Lösung:

```
import random

zahlen = random.sample(range(1, 50), 6)
print(sorted(zahlen))
```

8. (3 Punkte) Die Hilfe zum Modul `statistics` enthält:

```
median(data)
    Return the median (middle value) of numeric data.
    When the number of data points is odd, return the middle data point.
    When the number of data points is even, the median is interpolated by
    taking the average of the two middle values:

mode(data)
    Return the most common data point from discrete or nominal data.
```

Was erscheint auf der Konsole?

```
import statistics

print(statistics.median([5, 1, 3]))
print(statistics.median([1, 2, 2, 7]))
print(statistics.mode(['a', 'b', 'b', 'c']))
```

Lösung:

```
3
2.0
b
```

9. (2 Punkte) Die Hilfe zu `math.comb` beginnt so:

```
comb(n, k, /)
    Number of ways to choose k items from n items without repetition and without order.
    Evaluates to n! / (k! * (n - k)!) when k <= n and evaluates
    to zero when k > n.
```

   (a) (1 Punkt) Welche Werte haben `math.comb(5, 2)` und `math.comb(3, 5)`?

   (b) (1 Punkt) Beim Lotto werden 6 aus 49 Zahlen gewählt. Mit welchem Aufruf berechnet man die Anzahl der Möglichkeiten?

Lösung: a) 10 und 0
b) `math.comb(49, 6)`

10. (3 Punkte) Die Hilfe zu der Methode `most_common` der Klasse `collections.Counter` beginnt so:

```
most_common(self, n=None)
    List the n most common elements and their counts from the most
    common to the least.  If n is None, then list all element counts.
```

Schreibe ein Programm, das die zwei häufigsten Buchstaben im Wort `'abrakadabra'` mit ihrer Anzahl ausgibt.

Lösung:

```
from collections import Counter

print(Counter('abrakadabra').most_common(2))
```

Ausgabe: `[('a', 5), ('b', 2)]`

11. (3 Punkte) Die Hilfe zu der Methode `weekday` der Klasse `datetime.date` lautet:

```
weekday(...)
    Return the day of the week represented by the date.
    Monday == 0 ... Sunday == 6
```

(a) (1 Punkt) Der 30.10.2026 ist ein Freitag. Was ergibt `datetime.date(2026, 10, 30).weekday()`?

(b) (2 Punkte) Schreibe eine Funktion `ist_wochenende(d)`, die für ein Datum `d` prüft, ob es auf einen Samstag oder Sonntag fällt.

Lösung: a) 4

```
def ist_wochenende(d):
    return d.weekday() >= 5
```

12. (4 Punkte) Von der Klasse `Notenbuch` ist nur die Ausgabe von `help(Notenbuch)` bekannt:

```
class Notenbuch
 |  Notenbuch(fach)
 |  Ein Notenbuch für ein Fach.
 |
 |  hinzufuegen(self, note)
 |      note: ganze Zahl von 1 bis 6
 |
 |  schnitt(self)
 |      returns: Mittelwert der Noten, bei leerem Notenbuch None
 |
 |  beste(self)
 |      returns: die beste (kleinste) Note, bei leerem Notenbuch None
```

(a) (3 Punkte) Erzeuge ein Notenbuch für das Fach Mathe, trage die Noten 2, 3 und 1 ein und gib den Schnitt und die beste Note aus.

(b) (1 Punkt) Was ergibt `Notenbuch('Bio').schnitt()`?

Lösung:

```
n = Notenbuch('Mathe')
n.hinzufuegen(2)
n.hinzufuegen(3)
n.hinzufuegen(1)
print(n.schnitt())
print(n.beste())
```

b) `None`

13. (5 Punkte) Von einer Funktion ist nur der Kopf mit dem Docstring bekannt:

```
def kuerze(text, n, ende='...'):
    '''
    text: String
    n: ganze Zahl >= 0
    ende: String, der an einen gekürzten Text angehängt wird
    returns: text, falls er höchstens n Zeichen hat,
             sonst die ersten n Zeichen von text und danach ende
    '''
```

(a) (3 Punkte) Was ergeben `kuerze('Informatik', 4)`, `kuerze('Hallo', 10)` und `kuerze('Hallo', 2, '!')`?

(b) (2 Punkte) Implementiere die Funktion.

Lösung: a) `'Info...'`, `'Hallo'`, `'Ha!'`

```
def kuerze(text, n, ende='...'):
    if len(text) <= n:
        return text
    return text[:n] + ende
```

14. (3 Punkte) Was bedeuten die folgenden Angaben in einer Dokumentation?

(a) (1 Punkt) `round(number, ndigits=None)`: Was bedeutet `ndigits=None`?

(b) (1 Punkt) `Raises ValueError if the value is not present.`

(c) (1 Punkt) `randint(a, b): Return random integer in range [a, b], including both end points.`

Lösung: a) Das Argument `ndigits` ist optional, ohne Angabe gilt der Standardwert `None`.
b) Ist der Wert nicht vorhanden, gibt es einen `ValueError`.
c) Die Funktion gibt eine zufällige ganze Zahl von a bis b zurück, a und b sind möglich.

15. (2 Punkte) Beantworte die folgenden Fragen.

(a) (1 Punkt) Wie erhält man eine Liste aller Methoden von Strings?

(b) (1 Punkt) Warum muss man eine Klasse nicht vollständig lesen, um sie zu benutzen?

Lösung: a) Mit `dir(str)`.
b) Die Docstrings beschreiben die Schnittstelle. Wie die Klasse intern arbeitet, muss man für die Benutzung nicht wissen (Geheimnisprinzip).