

10 Ein- und Ausgaben

Lernziele In diesem Abschnitt lernst du:

- ▷ Mit beliebigen Texteingaben zu arbeiten.
- ▷ Resultate und Mitteilungen in einem Fensterchen auszugeben.

Einführung Du kennst bereits `inputInt` und `inputFloat`, um Zahlen einzugeben und kannst mit `print` Resultate auf den Bildschirm ausgeben. Neu kommen `inputString` und `msgDlg` hinzu, um auch Text eingeben und mit einem Fensterchen ausgeben zu können.

Magst du dich erinnern, dass ein Text (in Gänsefüßchen) im Programmieren «string» heisst? Ist es da nicht logisch, dass die Eingabefunktion für solchen Text `inputString()` genannt wird? Während du also bei `inputInt` und `inputFloat` nur Zahlen eingeben kannst, kannst du bei `inputString` jeden möglichen Text eingeben, z. B. deinen Namen:

```
name = inputString("Wie heisst du?")
print "Hallo", name
```

Wäre dieses Programm aber nicht noch besser, wenn auch die Ausgabe in einem kleinen Fensterchen erfolgen würde? Dafür gibt es `msgDlg()`. Es funktioniert (fast) genau gleich wie `print`, allerdings mit einem wichtigen Unterschied: Bei `print` setztst du *nie* Klammern, bei `msgDlg` dafür *immer*.

```
name = inputString("Wie heisst du?")
msgDlg("Hallo", name)
```

Das Programm Mit den neuen Ein- und Ausgaben können wir ein kleines Quiz schreiben, das auch gleich zählt, wie viele richtige und falsche Antworten gegeben wurden. Beachte, wie wir in Zeile 11 auch die Zahl 169 in Gänsefüßchen setzen müssen. `inputString` gibt uns nämlich auch dann einen String/Text zurück, wenn die Eingabe eigentlich eine Zahl wäre.

```
1 richtig = 0    # Anzahl richtige Antworten
2 falsch = 0    # Anzahl falsche Antworten
3
4 eingabel = inputString("Hauptstadt von Frankreich?")
5 if eingabel == "Paris":
6     richtig += 1
7 else:
8     falsch += 1
```

```
9
10 eingabe2 = inputString("Quadrat von 13?")
11 if eingabe2 == "169":
12     richtig += 1
13 else:
14     falsch += 1
15
16 msgDlg("Du hattest", richtig, "richtige und",
17        falsch, "falsche Antworten.")
```

Die wichtigsten Punkte Je nachdem, ob du als Eingabe eine ganze Zahl, eine beliebige Zahl oder einen String/Text erwartest, verwendest du eine andere Funktion: `inputInt()`, `inputFloat()`, `inputString()`.

Mit `msgDlg()` kannst du beliebigen Text, Variablen und Zahlen in einem eigenen kleinen Fensterchen ausgeben. Auch hier kannst du mit der Kombination `\n` in einem String einen Zeilenumbruch erzeugen.

AUFGABEN

39. Ändere die Zeile 10 im Beispielprogramm oben ab und verwende `inputInt` anstelle von `inputString`. Was musst du dann noch ändern, damit das Programm korrekt funktioniert?

40. Wenn jemand eine falsche Antwort gibt, dann lass dein Programm die korrekte Antwort ausgeben: Falsch. Richtig wäre «Paris».

41. Erweitere dein Quiz um eigene Fragen. In den Einstellungen von TigerJython kannst du auch das Editorfenster während der Programmausführung ausblenden lassen. Dann sieht man während des Quizes nicht gleich die Antworten im Code.

42. Schreibe das Programm aus dem Abschnitt über Fallunterscheidung (Seite 48) so um, dass du über `inputInt` eine Zahl eingeben kannst, die dann daraufhin überprüft wird, ob sie eine Quadratzahl ist.

43.* Schreibe ein Programm, das dein Sternzeichen herausfindet. Dazu muss man zuerst den Tag und den Monat seiner Geburt eingeben. Das Programm antwortet dann z. B. mit «Du bist Schütze.» Tipp: Berechne aus dem Datum zuerst den Tag im Jahr. Der «3. Februar» wäre demnach der 34. Tag.