

# CodeBytes - Flussüberfahrt

## Themen:

- Eingabe über die Konsole. Siehe dazu das Tutorial XY
- `if`-Abfragen. Siehe dazu das Tutorial XY
- `while`-Schleifen. Siehe dazu das Tutorial XY
- Texte zusammenfügen. Siehe dazu XY.

## Aufgabe:

Schreibe ein Programm, indem du folgendes Problem lösen musst: > Du, deine Ziege, ein Wolf und ein Sack Futter stehen an der linken Seite eines Flusses.

Du hast ein kleines Ruderboot, indem du nur eine Sache auf die andere Seite bringen kannst.

Allerdings darf der Wolf **nicht** mit der Ziege oder die Ziege mit Futter alleine auf einer Seite sein, da sonst die Ziege oder das Futter gefressen wird!

Wie kannst du alles auf die rechts Seite bringen?

## Darstellung

Um die Situation darzustellen, gib ein kleines Bild auf der Konsole aus:

```
-F-W-Z-Du-~~~~~<=>~~~~~-
```

Der Anfang: Hier sieht man, dass das Futter (F), der Wolf (W), die Ziege (Z) und Du auf der linken Seite des Flusses stehen.

```
-F-Z-~~~~~<=>~~~~~-Du-W-
```

Nach einer Überfahrt ändert sich die Situation: Du und der Wolf sind nun auf der rechten Seite. Allerdings ist die Ziege mit dem Futter alleine auf der linken Seite => Die Ziege frisst das Futter!

## Darstellung erzeugen

Am Besten erzeugst du die Ausgabe, indem du einen Text mit dem Mittelteil, dem Fluss hast. Dann fügst du jede nach Position der einzelnen Figuren diese entweder am Anfang des Textes, also auf der linken Flusseite, oder am Ende, auf der rechten Seite, an.

## Programm-Gerüst

1. du benötigst 4 Variablen, um zu speichern, ob die Figuren links oder rechts des Flusses stehen. Verwenden dafür die `True` (auf der linken Seite), `False` (auf der rechten Seite)
2. lasse das Programm in einer Endlosschleife laufen
3. Gib die aktuelle Situation aus
4. Lies die Eingabe ein:
  - Überprüfe ob das Programm beendet werden soll, über die Eingabe `ende`
  - Falls die Eingabe eine der vier Möglichkeiten: `nichts`, `wolf`, `ziege` oder `futter` ist, überprüfe ob diese Überfahrt möglich ist und ändere entsprechend die Zustandsvariablen der beteiligten Figuren
  - Bei einer ungültigen Eingabe, gib die Situation erneut aus und lies eine neue Eingabe ein
5. Überprüfe die aktuelle Situation, kann der Wolf die Ziege fressen? Kann die Ziege das Futter fressen? Ist alles auf der rechten Seite? Falls ja, beende das Programm mit einer entsprechenden Meldung

## Beispielablauf:

### Ungültige Überfahrt

```
-F-W-Z-Du-~~~~~<=>~~~~~
```

```
Was willst du mit auf die andere Seite nehmen: nichts, wolf, ziege, futter?
wolf
Die Ziege hat das Futter gefressen!
```

### Gültige Überfahrt

```
-F-W-Z-Du-~~~~~<=>~~~~~
```

```
Was willst du mit auf die andere Seite nehmen: nichts, wolf, ziege, futter?
ziege
```

```
-F-W-~~~~~<=>~~~~~Du-Z-
```

```
Was willst du mit auf die andere Seite nehmen: nichts, wolf, ziege, futter?
...
```

### Alles auf der rechten Seite => Ende

```
...
-Z-Du-~~~~~<=>~~~~~W-F-
```

Was willst du mit auf die andere Seite nehmen: nichts, wolf, ziege, futter?  
ziege  
Geschafft! Du hast alles auf die rechte Seite gebracht!

## Ungültige Eingabe

Die Eingabe fisch ist keine gültige Eingabe.

```
-F-W-Z-Du-~~~~~<=/=>~~~~~-
```

Was willst du mit auf die andere Seite nehmen: nichts, wolf, ziege, futter?  
fisch  
Ungültige Eingabe  
-F-W-Z-Du-~~~~~<=/=>~~~~~-  
...

Die Eingabe futter ist keine gültige Eingabe, da Du nicht auf der gleichen Seite wie das Futter bist!

Was willst du mit auf die andere Seite nehmen: nichts, wolf, ziege, futter?  
ziege  
-F-W-~~~~~<=/=>~~~~~Du-Z-

Was willst du mit auf die andere Seite nehmen: nichts, wolf, ziege, futter?  
futter  
Ungültige Eingabe  
-F-W-~~~~~<=/=>~~~~~Du-Z-  
...

**Hinweis:** Die ... bedeuten, dass hier noch weitere Eingaben stattfinden.

## Tipps:

Man kann eine Schleife mit `break` sofort verlassen, und mit `continue` an den Schleifenanfang springen und den nächsten Durchlauf beginnen. Der Befehl `pass` tut überhaupt nichts und kann als Platzhalter verwendet werden, wenn nichts passieren soll.

```
# Endlosschleife
while True:
    # ...
    if zu_ende:
        break
    if nachster_durchlauf:
        continue
```

```
if tue_nichts:  
    pass
```

- Denke daran, dass du jedes die Seite mit wechselt, egal welchen Gegenstand Du mitnimmst