

Unit 11

Dr. Günter Kolousek

21. Juli 2015

Lege wiederum ein Verzeichnis an. Nennes es 16_unit11! In diesem Verzeichnis sollen alle Dateien der jeweiligen Einheit abgelegt werden.

1 Schulübungen

1. Programmiere das "Stein, Papier und Schere" Spiel.

Folgende Regeln gelten:

- Der Stein bricht die Schere.
- Die Schere schneidet das Papier.
- Das Papier umwickelt den Stein.

Es spielt ein Mensch gegen den Computer. Der Computer wählt zufällig einen der drei Gegenstände. Das Spiel beendet sich sobald der Mensch das Spiel verlässt oder ein vorher definierter Punktestand (entweder Mensch oder Computer) erreicht wird.

Zuerst soll dieser Punktestand eingegeben werden. Wir vorerst davon ausgehen, dass der Mensch nur richtige Eingaben tätigt. Das Spiel läuft dann folgendermaßen ab:

Willkommen zu Stein, Papier und Schere!

Wieviele Punkte sind zum Sieg nötig? 3

```
Wähle:  (1)Stein   (2)Papier   (3)Schere   1
Mensch: Stein      Computer:  Papier      => Computer gewinnt!
Punkte: Mensch 0   Computer 1
```

```
Wähle:  (1)Stein   (2)Papier   (3)Schere   1
```

Mensch: Stein Computer: Schere => Mensch gewinnt!
Punkte: Mensch 1 Computer 1

Wähle: (1)Stein (2)Papier (3)Schere 2
Mensch: Papier Computer: Papier => Unentschieden!
Punkte: Mensch 1 Computer 1

Wähle: (1)Stein (2)Papier (3)Schere 3
Mensch: Schere Computer: Papier => Mensch gewinnt!
Punkte: Mensch 2 Computer 1

Wähle: (1)Stein (2)Papier (3)Schere 1
Mensch: Stein Computer: Schere => Mensch gewinnt!
Punkte: Mensch 3 Computer 1

Mensch hat gewonnen!!

Verwende die Möglichkeiten der `format` Methode der Strings um die Ausgabe zu erzielen! Die `format` funktioniert folgendermaßen:

```
>>> "Die offene Rechnung beträgt {} Euro".format(300)
'Die offene Rechnung beträgt 300 Euro'
>>> "Die offene Rechnung beträgt {} Euro und {} Cents".format(300, 50)
'Die offene Rechnung beträgt 300 Euro und 50 Cents'
>>> "Die offene Rechnung beträgt {1} Euro und {0} Cents".format(50, 300)
'Die offene Rechnung beträgt 300 Euro und 50 Cents'
>>> "Die offene Rechnung beträgt {euro} Euro und {cents} Cents".format(
... cents=50, euro=300)
'Die offene Rechnung beträgt 300 Euro und 50 Cents'
>>> "Die offene Rechnung beträgt {euro:5} Euro und {cents:2} Cents".\
... format(cents=50, euro=300)
'Die Rechnung beträgt    300 Euro und 50 Cents'
>>> "Die offene Rechnung beträgt {euro:<5} Euro und {cents:2} Cents".\
... format(cents=5, euro=300)
'Die Rechnung beträgt 300    Euro und   5 Cents'
>>> "Die offene Rechnung beträgt {euro:^5} Euro und {cents:>2} Cents".\
... format(cents=5, euro=300)
'Die Rechnung beträgt   300    Euro und   5 Cents'
>>> "Die offene Rechnung beträgt {euro:=^5} Euro und {cents:>2} Cents".\
... format(cents=5, euro=300)
'Die Rechnung beträgt =300= Euro und   5 Cents'
>>> "Die Rechnung beträgt {euro:b} Euro und {cents:b} Cents".\
... format(cents=5, euro=300)
'Die Rechnung beträgt 100101100 Euro und 101 Cents'
```

Die genaue Beschreibung kann der Dokumentation auf der Python-Homepage entnommen werden. Aber hier gilt auch: "Probieren geht über studieren!"

Zufallszahlen können mit der Funktion `randint` aus dem Modul `random` ermittelt werden. Die genaue Funktion kann wieder mittels der `help()` Funktion eruiert werden.

2. Erweitere nun das "Stein, Papier und Schere" Spiel, sodass jetzt der Name von dem Spieler abgefragt wird und außerdem auch jetzt mit Fehleingaben des Benutzers korrekt umgegangen wird.
 - a) Dazu muss die Eingabe des Spielernamen programmiert werden. Dieser Spielernamen darf maximal 5 Zeichen haben (damit dieser schön in die Ausgabe passt).
 - b) Weiters muss die Ausgabe adaptiert werden. Statt "Mensch" muss jeweils der Spielername eingesetzt werden. Die Ausgabe soll aber weiter bündig formatiert bleiben.
 - c) Gibt der Benutzer "falsche" Eingaben ein, dann soll jeweils eine Meldung ausgegeben werden und der Benutzer muss solange erneut eingeben bis die Eingabe korrekt ist.
3. Erweitere das Spiel "Stein, Papier und Schere" Spiel nochmals, sodass jetzt mehrere Benutzer hintereinander das Spiel spielen können und das Spiel sich die Spielstände der Benutzer merkt und am Ende die Punktestände aller Spieler ausgibt und den Sieger ermittelt.

Verwende dazu den Datentyp `dict`!

2 Hausübung

Kapitel 11 lesen!