# Lektion 12 – Module

## Nachschlagen bei…

W3Schools: Math und weiter unten in der Module Reference unter Random Module und Math Module.

## Anweisungen

Studiere die genannten Kapitel in W3Schools und die folgenden Anweisungen. Beantworte die Fragen in diesem Dokument und kopiere den Code der Aufgaben hierher.

## Erläuterungen

Module sind nichts anderes als Codebibliotheken, in die Funktionen und Variablen ausgelagert werden, damit der eigentliche Programmcode nicht zu lang und unübersichtlich wird und damit man die Funktionen für verschiedene Programme verwenden kann. Module kann man selbst programmieren. Es gibt aber eine größere Anzahl von Modulen, die bei Python schon dabei sind (wie Math oder Random) oder von Fremdanbietern stammen. Sie werden eingebunden mit import modulname. Dann können die enthaltenen Funktionen mit modulname.funktionsname() aufgerufen werden. Mit import modulname as abk kann dem Modulnamen ein kürzerer Aliasname gegeben werden. Mit from modulname import funktionsname kann auch nur eine einzelne Funktion importiert werden, die dann direkt verwendet werden kann.

## Beispiele

### Beispiel 1 - Import des ganzen Moduls

import math
x = math.sqrt(64)
print(x) *Probier aus, was dieser Code macht*

### Beispiel 2 - Import des ganzen Moduls mit Alias

import math as m
x = m.pi
print(x) *Probier aus, was dieser Code macht*

### Beispiel 3 - Import einer einzigen Funktion aus einem Modul

from math import sin
x = sin(3.141592653589793/2)
print(x) *Probier aus, was dieser Code macht. Warum liefert er dieses Ergebnis?*

## Fragen

Mit welchem Schlüsselwort wird ein Modul eingebunden? Mit welcher Anweisung wird eine einzelne Funktion aus einem Modul eingebunden?

Welche Ausgaben erzeugen die vorherigen drei Beispiele?

## Aufgaben

1. Erstelle eine Datei sinus.py. Zuerst wird der Benutzer gefragt, von welchem Winkel (In Grad) er den Sinus berechnen will. Dann wird der Winkel in Bogenmaß umgewandelt (muss so sein bei math.sin). Ausgegeben wird dann ein Antwortsatz. (Etwa so: Der Sinus von 270.0 Grad ist -1.0). Erschwernis: das Programm soll ohne die Verwendung von math.radians auskommen!
2. Erstelle eine Datei random\_spiel.py. Erzeuge eine Zufallszahl von 1 bis 100. Die Spielerin muss (über input) eine Zahl eingeben. Hat sie falsch geraten, erfährt sie, ob die gesuchte Zahl größer oder kleiner ist und muss so lange raten, bis sie die richtige Zahl gefunden hat. Im Unterschied zu einer früheren Aufgabe soll die Überprüfung in einer Funktion (def ueberpruefung) gemacht werden, die den entsprechenden Satz („Die gesuchte Zahl ist kleiner“, „…größer“, „gewonnen“) ausgibt und als Rückgabewert „True“ oder „False“ hat. Die Schleife mit Eingabe und Überprüfung wird so lange durchgeführt, bis die Überprüfung „True“ liefert. Du brauchst das Modul „random“.