

Aufgabe 4 - (Teil 1 von SCHIFFE VERSENKEN)

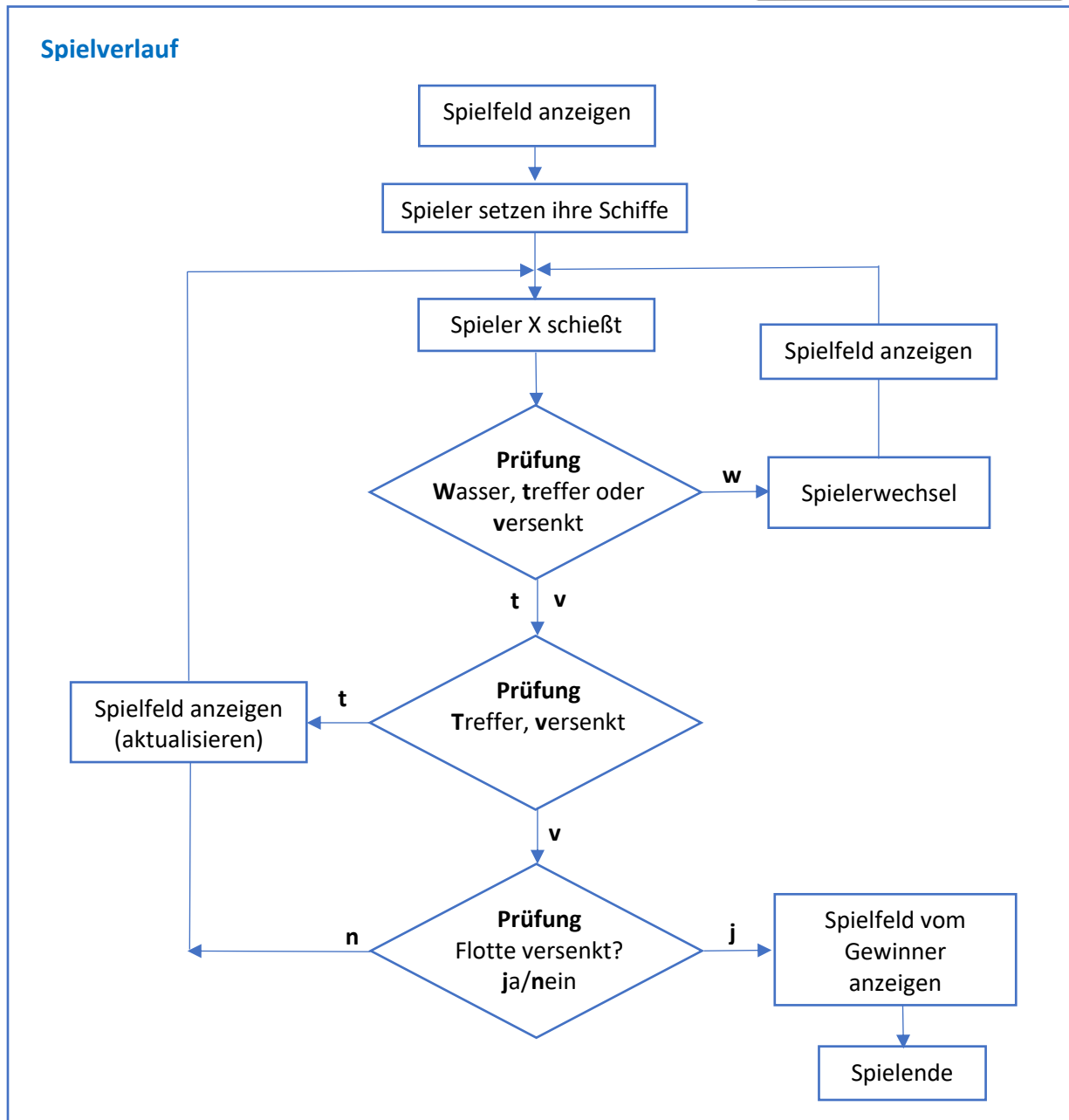
Situation: Laut einer aktuellen VuMA-Umfrage spielen mehr als 40 Prozent der Deutschen Computer- und Videospiele. In der Altersgruppe der 14- bis 29-Jährigen beträgt der Anteil der Videospiele sogar über 70 Prozent. Die meisten Gamer interessieren sich für Action-Spiele/ Ego-Shooter, Abenteuer-Spiele und Geschicklichkeitsspiele.



Es sollte das bekannte Spiel **SCHIFFE VERSENKEN** als Python-Programm erstellt werden.

Zuerst betrachten wir den **Spielverlauf** und überlegen uns, welche Prozesse wir als **Funktionen** umsetzen können.

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										



Folgende Funktionen leiten wir vom Spielverlauf ab:

Spielfeld anzeigen	spielfeld_anzeigen()
Spieler setzt seine Schiffe	spieler_flotte()
Spieler schießt	spieler_schiesst()
Spielerwechsel	spieler_wechsel()
Prüfung Wasser, Treffer oder versenkt	spieler_pwtv()
Prüfung Treffer, versenkt	spieler_ptv()
Prüfung Flotte versenkt	spieler_pfv()

In der heutigen Aufgabe geht es zuerst um die Ausgabe des Spielfeldes zu **SCHIFFE VERSENKEN**. Dafür werden wir den Datentyp „**Liste**“ einsetzen.

a.) Spielfeld anzeigen

Für das Spielfeld benötigen wir eine Liste mit **zehn** Zeilen und **zehn** Spalten die das Quadrat aus 100 Ziffern bilden:

	A	B	C	D	E	F	G	H	I	J
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

Hinweis:
Wir beginnen mit der 0 und nicht mit der 1, da wir die Eingabe des Benutzers auf „einstellige Eingabe“ ein Buchstabe (a – j) und eine Ziffer (0 – 9).

Am Anfang vom Programm wird als erstes das Spielfeld als **Liste** angelegt mit den Ziffern 0 bis 99 oder einem Zeichen bzw. Leerzeichen.

Für die Ausgabe des Spielfeldes werden die Inhalte der Liste mit dem Zeichen „|“ voneinander getrennt. Erstellen Sie hierzu die Funktion: **spielfeld_anzeigen()**

Datentyp Liste

Eine Liste ist eine Sequenz von Objekten. Sie kann Elemente unterschiedlichen Objekttyps enthalten. Man kann sich eine Liste als ein- oder mehrdimensionales Feld vorstellen (Array).

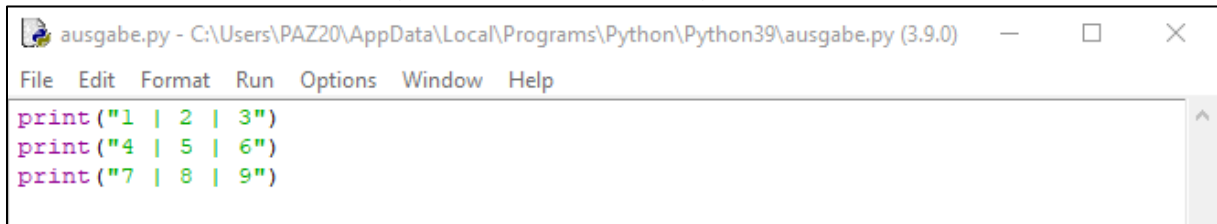
1	z = [3, 6, 12.5, -8, 5.5]	← Liste mit 5 Elementen
2	print(z)	gesamte Liste
3	print(z[0])	ein Element
4	print(z[0:3])	Ein Teilbereich (engl. Slice)

Ausgabe: [3, 6, 12.5, -8, 5.5]
3
[3, 6, 12.5]

Achten Sie darauf, der Listen **Index** beginnt immer mit **0!**

Hinweis:

Erstellen Sie zuerst eine Ausgabe mit drei **print()**-Zeilen ohne Liste und ersetzen Sie danach die Ziffern mit den Feldern der Liste (enthält die Ziffern 1 bis 9).

A screenshot of a Python IDE window titled 'ausgabe.py - C:\Users\PAZ20\AppData\Local\Programs\Python\Python39\ausgabe.py (3.9.0)'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The main text area contains three lines of Python code:

```
print("1 | 2 | 3")  
print("4 | 5 | 6")  
print("7 | 8 | 9")
```

Abb.: Python-Programm: Ausgabe mit drei **print()**-Zeilen

Ihr Programm... **afg4_spiel_feld.py**

def **spiel_feld_anzeigen()**:

print()-Ausgaben

#Hauptprogramm

spiel_feld_anzeigen()