

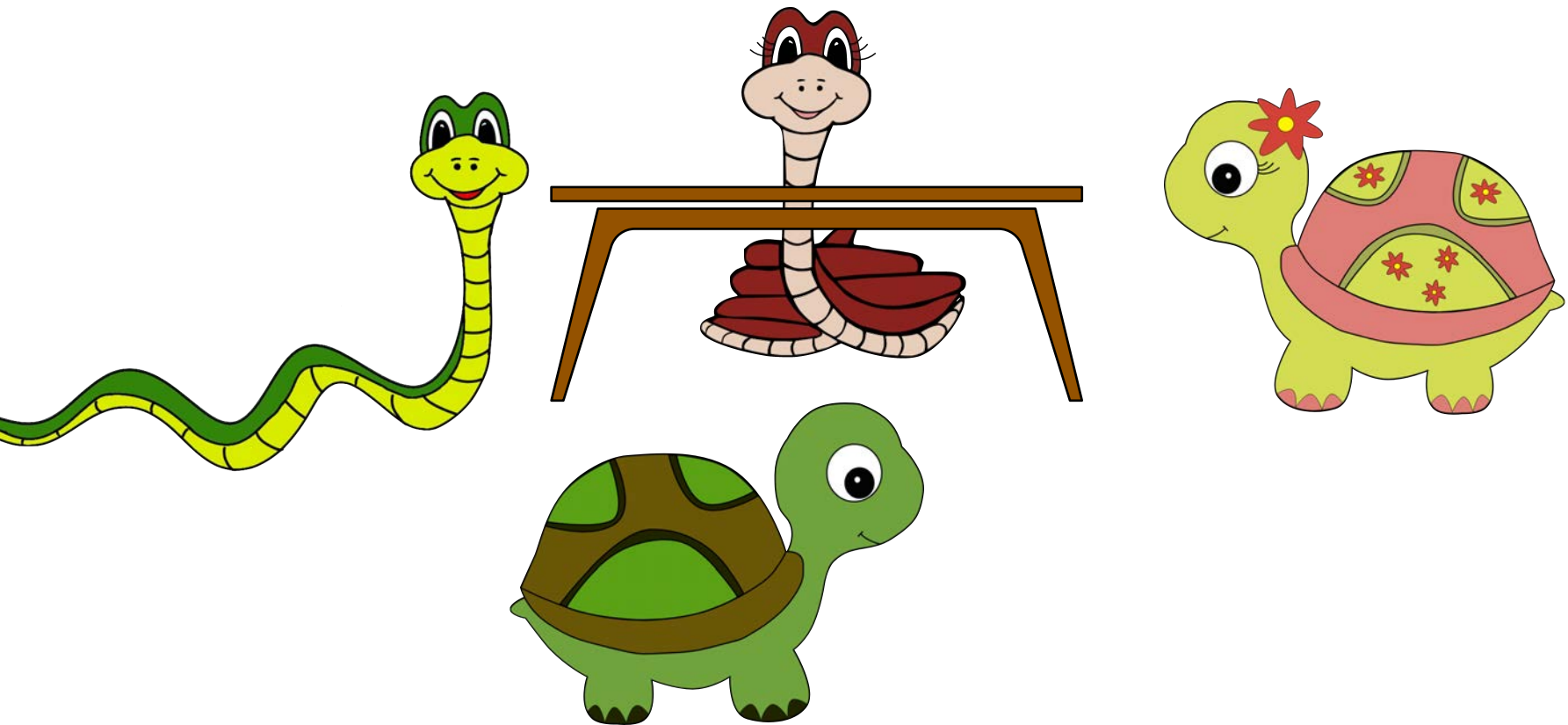


Programmieren
lernen mit Python

Verzweigungen (2/2)

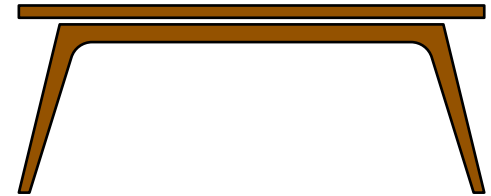
Kira Grammel, Nina Ihde, Sebastian Serth & Selina Reinhard
Hasso-Plattner-Institut
Universität Potsdam

Plätze finden im Speisesaal



Mehrfachverzweigungen

```
1 if sitze == 4:
2     print("Der Tisch ist perfekt!")
3 elif sitze < 4:
4     print("Der Tisch ist zu klein.")
5 else:
6     print("Der Tisch ist zu groß.")
```



If-elif-else

- Wir können mit `elif` mehrere Fallunterscheidungen kombinieren
- Hinter `elif` folgt eine weitere Bedingung
- Können mehrere `elif` Zweige haben

Vergleiche

```
1 print("a" < "b")  
2 print("a" > "b")
```

```
1 print(6 == 7)  
2 print(6 != 7)
```

```
1 print(6 <= 7)  
2 print(6 >= 6)
```

True
False

False
True

True
True

Vergleiche

- Können Variablen und/oder Vergleichswerte miteinander vergleichen
- Vergleiche liefern einen Boolean
- `>` heißt "größer als"
- `==` heißt "ist gleich"
- `!=` heißt "ist ungleich"

Verknüpfung von Bedingungen and

```
1 sitze = 4
2 ort = "Fenster"
3 if sitze == 4 and ort == "Fenster":
4     print("Der Tisch hat 4 Sitze und ist am Fenster.")
```

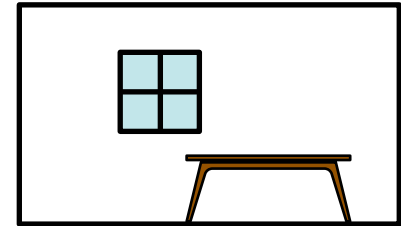
Der Tisch hat 4 Sitze und ist am Fenster.

Boolesche Aussagenlogik

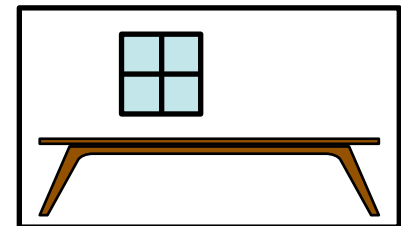
- Können Booleans verknüpfen
- and
 - Heißt "und"
 - wahr, wenn beide wahr sind

Verknüpfung von Bedingungen and

$4 == 4$
"Fenster" == "Fenster"
and
True



$8 == 4$
"Fenster" == "Fenster"
and
False

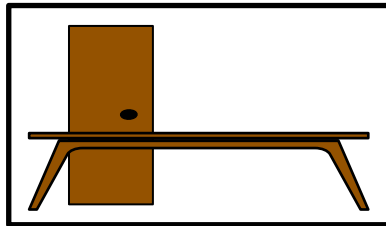
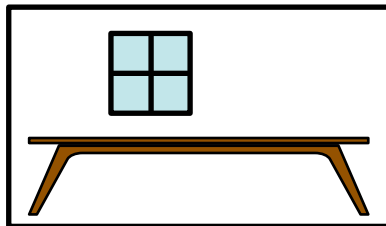
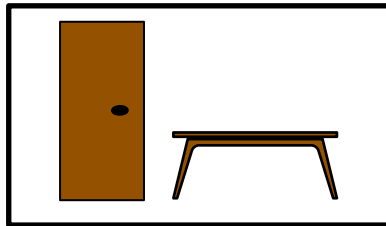
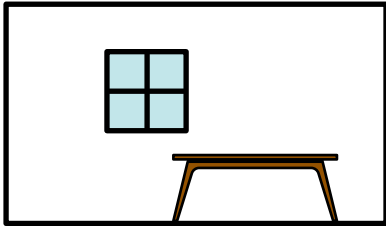


Verknüpfung von Bedingungen and

4 Sitze

Am Fenster

Tisch hat 4 Sitze **und** ist am Fenster



Verknüpfung von Bedingungen or

```
1 ort = "Fenster"  
2 if ort == "Fenster" or ort == "Tür":  
3     print("Der Tisch ist am Fenster oder an der Tür.")
```

Der Tisch ist am Fenster oder an der Tür.

Boolesche Aussagenlogik

■ or

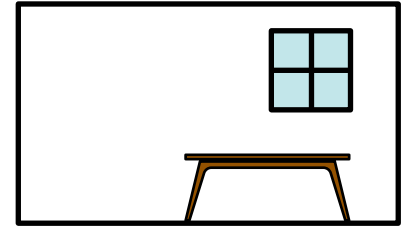
- Heißt "oder"
- wahr, wenn mindestens eins wahr ist

Verknüpfung von Bedingungen or

"Fenster" == "Fenster"
"Fenster" == "Tür"

or

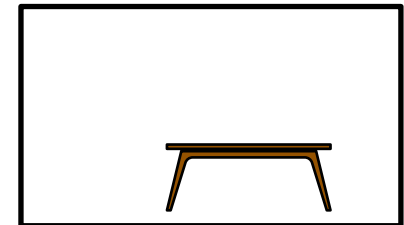
True



"Wand" == "Fenster"
"Wand" == "Tür"

or

False

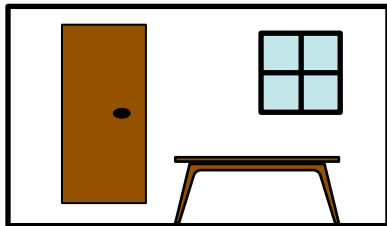


Verknüpfung von Bedingungen or

Am Fenster

An der Tür

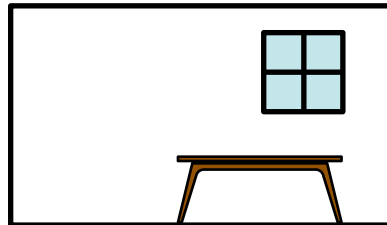
Tisch ist am Fenster **oder** an der Tür



✓

✓

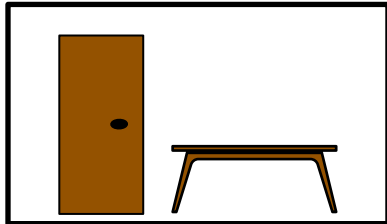
✓



✓

✗

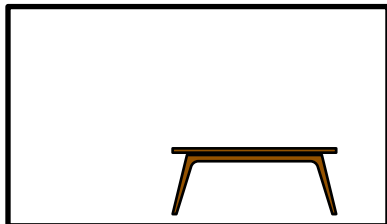
✓



✗

✓

✓



✗

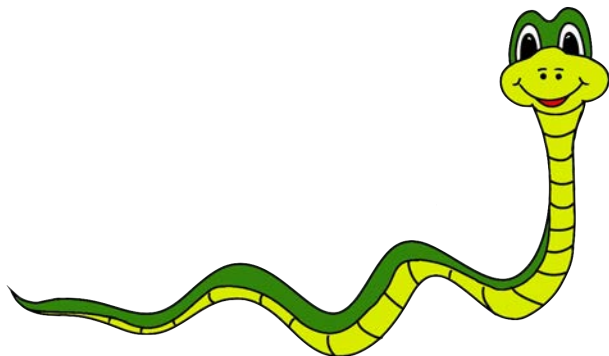
✗

✗

Plätze finden im Speisesaal

```
1 sitze = 4
2 ort = "Tür"
3 if sitze == 4 and (ort == "Fenster" or ort == "Tür"):
4     print("Juhu, den Tisch nehmen wir.")
```

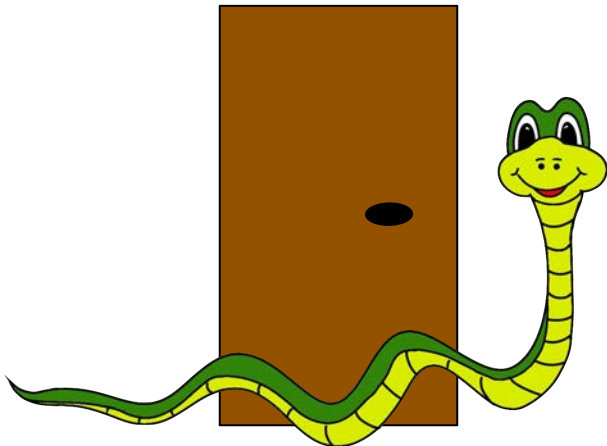
- Simon, Leonie und zwei weitere Freunde suchen einen Tisch, an den sie alle 4 passen und der am Fenster oder an der Tür steht.
- Nutzen dafür die Variable `sitze` (ein Integer) und `ort` (ein String)



Plätze finden im Speisesaal

```
1 sitze = 4
2 ort = "Tür"
3 if sitze == 4 and (ort == "Fenster" or ort == "Tür"):
4     print("Juhu, den Tisch nehmen wir.")
```

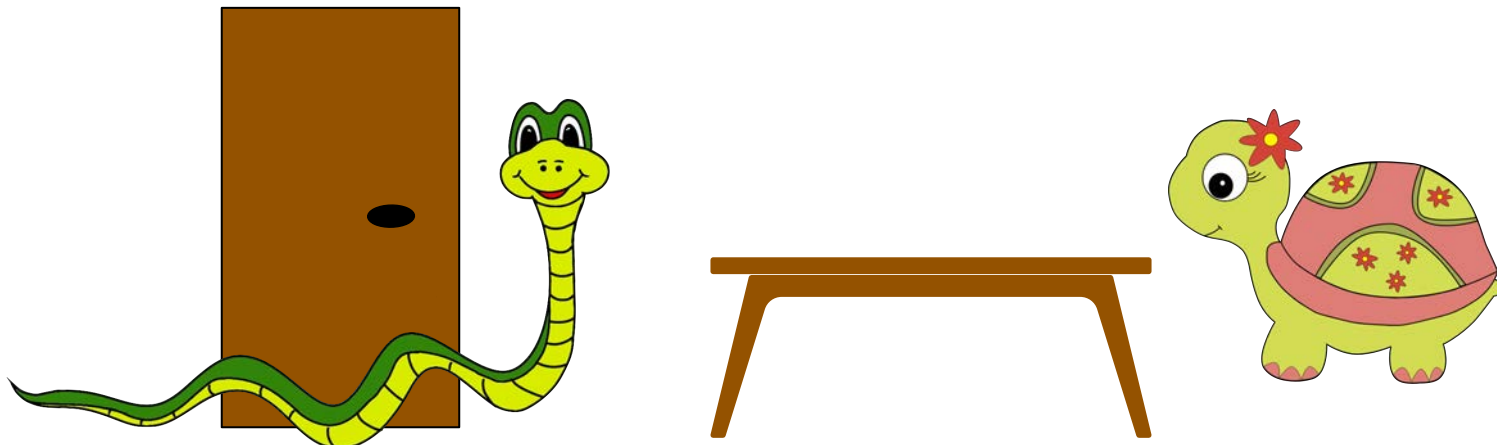
Juhu, den Tisch nehmen wir.



Plätze finden im Speisesaal

```
1 sitze = 8
2 ort = "Tür"
3 if sitze == 4 and (ort == "Fenster" or ort == "Tür"):
4     print("Juhu, den Tisch nehmen wir.")
5 elif sitze != 4:
6     print("Wir brauchen einen Tisch mit genau 4 Sitzen.")
```

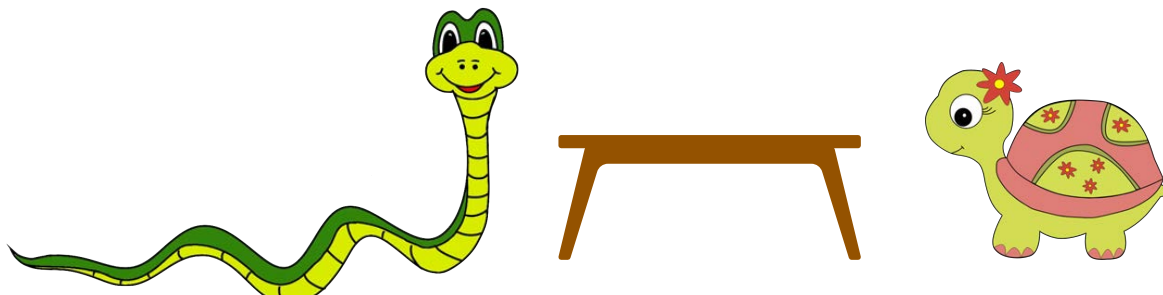
Wir brauchen einen Tisch mit genau 4 Sitzen.



Plätze finden im Speisesaal

```
1 sitze = 4
2 ort = "Wand"
3 if sitze == 4 and (ort == "Fenster" or ort == "Tür"):
4     print("Juhu, den Tisch nehmen wir.")
5 elif sitze != 4:
6     print("Wir brauchen einen Tisch mit genau 4 Sitzen.")
7 else:
8     print("Der Tisch hat zwar die richtige Größe, ist aber
          weder am Fenster noch an der Tür.")
```

Der Tisch hat zwar die richtige Größe, ist aber weder am Fenster noch an der Tür.



Zusammenfassung

```
1 if Bedingung1:
2     Aktion1
3 elif Bedingung2:
4     Aktion2.1
5     Aktion2.2
6 elif Bedingung3:
7     ...
8 else:
9     Aktion
```

- Bedingungen können Vergleiche und Boolesche Aussagenlogik enthalten
- `and` wahr, wenn beide wahr sind
- `or` wahr, wenn mindestens eins wahr ist

