

A cartoon illustration of a green snake with a yellow underbelly and a friendly expression. The snake is positioned on the left side of the slide, with its head and upper body visible. A large, white, oval-shaped speech bubble with a black outline extends from the snake's mouth towards the center of the slide. The background of the slide is a photograph of a London street scene, featuring the Big Ben clock tower on the right, a red double-decker bus in the foreground, and various historic buildings and trees. The bottom of the slide has a solid red overlay.

Programmieren
lernen mit Python

Listen (1/3)

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

Schnitzeljagd

Taschenlampe

Regenjacke

Bleistift

Schaufel

Schere



Listen

```
1 schnitzeljagd = ["Stift", "Regenjacke", "Lampe"]  
2 print(schnitzeljagd)
```

```
["Stift", "Regenjacke", "Lampe"]
```

Listen

- Ermöglichen es, mehrere Daten zusammen zu speichern
- Enthalten Elemente, die in eckige Klammern [] geschrieben und durch Kommata , getrennt werden
- Gleiche Elemente können mehrfach vorkommen

Listen

```
1 schnitzeljagd = ["Stift", "Regenjacke", "Lampe", "Lampe"]  
2 print(schnitzeljagd)
```

```
["Stift", "Regenjacke", "Lampe", "Lampe"]
```

Listen

- Ermöglichen es, mehrere Daten zusammen zu speichern
- Enthalten Elemente, die in eckige Klammern [] geschrieben und durch Kommata , getrennt werden
- Gleiche Elemente können mehrfach vorkommen

Länge der Liste

```
1 schnitzeljagd = ["Stift", "Regenjacke", "Lampe"]
2 anzahl_elemente = len(schnitzeljagd)
3 print(anzahl_elemente)
```

3

len()

- Gibt die Anzahl an Elementen der Liste zurück
- Funktioniert wie bei Strings:

```
1 text = "Schnitzeljagd"
2 anzahl_zeichen = len(text)
3 print(anzahl_zeichen)
```

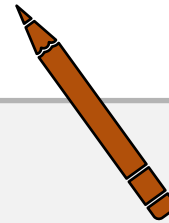
13

Zugriff auf Elemente

```
1 schnitzeljagd = ["Stift", "Regenjacke", "Lampe"]
2 print(schnitzeljagd[0])
```

0 1 2

Stift



Zugriff

- Index = Position eines Elements in der Liste
- Zugriff auf Elemente der Liste funktioniert über Index
 - Index beginnt bei 0 und endet bei `len(Liste) - 1`
 - Nur Integer können als Index verwendet werden

Zugriff auf Elemente

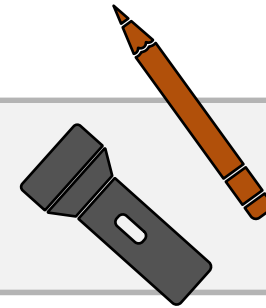
```

1 schnitzeljagd = ["Stift", "Regenjacke", "Lampe"]
2 print(schnitzeljagd[0])
3 print(schnitzeljagd[-1])

```

-3
-2
-1

Stift
Lampe



Zugriff

- Index = Position eines Elements in der Liste
- Zugriff auf Elemente der Liste funktioniert über Index
 - Index beginnt bei 0 und endet bei `len(Liste) - 1`
 - Nur Integer können als Index verwendet werden
 - "Negative Indexierung" auch möglich

Modifikation von Elementen

```
1 schnitzeljagd = ["Stift", "Regenjacke", "Lampe"]
2                 0           1           2
3 schnitzeljagd[1] = "Schere"
4 print(schnitzeljagd)
```

["Stift", "Schere", "Lampe"]



Modifikation

- Elemente einer Liste können überschrieben werden, nachdem über den Index auf sie zugegriffen wurde

Zugriff auf Zeichen

```
           01234
1 schnitzeljagd = ["Stift", "Schere", "Lampe"]
2 print(schnitzeljagd[0][1])
```

t

Zugriff

- Wenn Elemente einer Liste Strings sind, kann auf Zeichen der Elemente über Index zugegriffen werden

Zugriff auf Zeichen

```
                                012345
1 schnitzeljagd = ["Stift", "Schere", "Lampe"]
2 print(schnitzeljagd[0][1])
3 print(schnitzeljagd[-2][-2])
```

t
r

Zugriff

- Wenn Elemente einer Liste Strings sind, kann auf Zeichen der Elemente über Index zugegriffen werden

Zugriff auf Zeichen

```
1 schnitzeljagd = "Stift"  
2 print(schnitzeljagd[1])
```

t

Zugriff

- Strings werden als Liste von Zeichen interpretiert

Konkatenation & Multiplikation

```
1 bei_schnitzeljagd = ["Simon", "Leonie"]  
2 in_schule = ["Stella"]  
3 print(bei_schnitzeljagd + in_schule)
```

```
["Simon", "Leonie", "Stella"]
```

```
1 bei_schnitzeljagd = ["Simon", "Leonie"]  
2 print(bei_schnitzeljagd * 2)
```

```
["Simon", "Leonie", "Simon", "Leonie"]
```

- Listen fassen eine Menge an Daten zusammen
- Über den Index kann man...
 - ... auf Elemente zugreifen
 - ... Elemente verändern
 - ... auf einzelne Zeichen eines Strings zugreifen
- Listen können mit `+` kombiniert und mit `*` wiederholt werden

