Kontrollstrukturen

Ludwig Ettner

16. Juli 2024

Übersicht

Einführung

- Was sind Datentypen?
- Warum sind sie wichtig?
- überblick über die Datentypen in Python

Variablen

- Variablen speichern Datenwerte
- Beispiel:

```
1    name = "Max"
2    alter = 30
3    liste = [1, 2, 3]
```

Variablennamen müssen mit Buchstaben oder Unterstrichen beginnen

Keine Leerzeichen im Namen

Primitive Datentypen

- **Strings:** Textdarstellung in Python

```
1 text = "Hallo, Welt!"
```

  - `len()` - Gibt die Länge des Strings zurück.
  - `upper()`, `lower()` - Wandelt den String in Groß- bzw. Kleinbuchstaben um.
  - `strip()` - Entfernt Leerzeichen am Anfang und Ende des Strings.

- **Integer:** Ganze Zahlen

```
1 zahl = 42
```

- **Float:** Kommazahlen

```
1 zahl = 3.14
```

- **Boolean:** Wahrheitswerte

```
1 wahr = True
2 falsch = False
```

Komplexe Datentypen

- **Listen:** Sequenzen von Elementen

```
1 liste = [1, 2, 3, "vier", 5.0]
```

  - `append()` - Fügt ein Element am Ende der Liste hinzu.
  - `extend()` - Erweitert die Liste um die Elemente einer anderen Liste.
  - `insert()` - Fügt ein Element an einer bestimmten Position ein.

- **Tuples:** Unveränderliche Sequenzen

```
1 tupel = (1, 2, 3)
```

- **Dictionaries:** Schlüssel-Wert-Paare

```
1 woerterbuch = {"Name": "Max", "Alter": 30}
```

  - `keys()` - Gibt die Schlüssel des Dictionaries zurück.
  - `values()` - Gibt die Werte des Dictionaries zurück.
  - `items()` - Gibt Schlüssel-Wert-Paare des Dictionaries zurück.

Zugriff auf Elemente

- **Strings:**

```python
text = "Python"
print(text[0])  # Ausgabe: P
```

- **Listen:**

```python
liste = [1, 2, 3]
print(liste[0])  # Ausgabe: 1
```

  - Indexierung von hinten: `liste[-1]` gibt das letzte Element zurück.

- **Dictionaries:**

```python
woerterbuch = {"Name": "Max", "Alter": 30}
print(woerterbuch["Name"])  # Ausgabe: Max
```

Spezielle Funktionen

- **Strings:**

```
1    text = "Python"
2    print(len(text))   # Ausgabe: 6
3    print(text.upper())   # Ausgabe: PYTHON
4    print(text.lower())   # Ausgabe: python
5    print(text.strip())   # Entfernt Leerzeichen
```

Spezielle Funktionen

- **Listen:**

```
liste = [1, 2, 3]
liste.append(4)   # Fuegt 4 am Ende hinzu
liste.extend([5, 6])   # Erweitert um 5 und 6
liste.insert(1, "neu")   # Fuegt "neu" an Index 1 ein
```

Spezielle Funktionen

- **Dictionaries:**

```
1 woerterbuch = {"Name": "Max", "Alter": 30}
2 print(woerterbuch.keys())   # Ausgabe:
      dict_keys(['Name', 'Alter'])
3 print(woerterbuch.values())   # Ausgabe:
      dict_values(['Max', 30])
4 print(woerterbuch.items())   # Ausgabe:
      dict_items([('Name', 'Max'), ('Alter',
      30)])
```

Besonderheiten

- **2D-Listen:**

```
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
print(matrix[1][2])  # Zugriff auf Element in der 2. Liste, 3. Element -> Ausgabe: 6
```

- **Listen mit verschiedenen Datentypen:**

```
mischliste = [1, "zwei", 3.0, True]
```

- **Tuples als Schlüssel in Dictionaries:**

```
woerterbuch = {(1, 2): "Tupel als Schluessel"}
print(woerterbuch[(1, 2)])  # Ausgabe: Tupel als Schluessel
```

Beispiele

- Beispiel für Berechnungen mit Variablen

```
x = 5
y = 3
summe = x + y
print(summe)   # Ausgabe: 8
```

- Beispiel für Schleifen (nicht in dieser Präsentation behandelt)

```
for zahl in liste:
    print(zahl)
```

Zusammenfassung

- Datentypen in Python sind wichtig für die Speicherung und Manipulation von Daten.
- Primitive Datentypen umfassen Strings, Integer, Floats und Booleans.
- Komplexe Datentypen sind Listen, Tuples und Dictionaries.
- Variablen speichern Werte, die im Programm verwendet werden können.
- Python bietet viele eingebaute Funktionen für die Arbeit mit verschiedenen Datentypen.
- Besonderheiten wie 2D-Listen und Listen mit unterschiedlichen Datentypen erweitern die Funktionalität.

Fragen

Vielen Dank!
Fragen?