PYTHON GRUNDLAGEN

1. VARIABLEN

Variablen speichern Werte dynamisch ohne explizite Typdeklaration.
Ein Wert wird über das Gleichheitszeichen zugewiesen.



- Zuweisung: x = 10 oder name = 'Alice'
- Erlaubt: Buchstaben (a-z, A-Z), Ziffern und Unterstriche (_)
- Verboten: Ziffer am Anfang (z.B. 1x ist ungültig) sowie reservierte Schlüsselwörter
- Groß-/Kleinschreibung wird unterschieden (Case-Sensitive: port != Port)

2. DATENTYPEN

Python erkennt elementare Datentypen automatisch zur Laufzeit (dynamische Typisierung).



- int: Ganze Zahlen ohne Dezimalpunkt (z.B. port = 80, count = -3)
- float: Fließkommazahlen mit Punkt (z.B. version = 3.11, timeout = 0.5)
- str: Zeichenketten in einfachen oder doppelten Anführungszeichen (z.B. host = 'localhost')
- bool: Wahrheitswerte **True** oder **False** (z.B. is_admin = True)

3. BEDINGUNGEN

Entscheidungsstrukturen steuern den Programmablauf basierend auf logischen Vergleichen.



- Syntax: if bedingung: ... elif alternative: ... else: ...
- Vergleichsoperatoren: == (Gleichheit), != (Ungleichheit), <, >, <=, >=
- Logische Verknüpfungen: **and** (Und), or (Oder), not (Verneinung)
- Beispiel: **if** port == 443: status = 'HTTPS' **else**: status = 'Unbekannt'

4. SCHLEIFEN

Schleifen iterieren über Datenmengen oder laufen, solange eine Bedingung wahr ist.



- for-Schleife: for i in range(5): läuft von 0 bis 4
- while-Schleife: while count > 0: wiederholt, bis Bedingung False wird
- range(start, stop, step): Erzeugt Zahlenfolgen effizient
- Abbruch: **break** beendet Schleife sofort; **continue** springt zum nächsten Durchlauf

5. LISTEN

Listen sind veränderbare, geordnete Sammlungen von Elementen beliebigen Typs.



- Definition: ports = [22, 80, 443]
- Indexierung: Nullbasiert; ports[0] liefert 22, ports[-1] liefert 443
- Slicing: Teillisten extrahieren mit ports[0:2] (Elemente bei Index 0 und 1)
- Methoden: ports.append(8080) fügt an; ports.pop() entfernt das letzte Element

6. FUNKTIONEN

Funktionen kapseln Teilaufgaben, nehmen Parameter entgegen und geben Werte zurück.



- Definition mit Schlüsselwort **def**: def scan(target, timeout=1):
- Rückgabewert: return status liefert das Ergebnis an den Aufrufer
- Gültigkeitsbereich (Scope): Variablen innerhalb einer Funktion sind lokal
- Aufruf: result = scan('192.168.1.1')

7. DICTIONARIES

Dictionaries speichern assoziative Schlüssel-Wert-Paare für schnellen Datenzugriff.



- Definition: service = {'port': 22, 'protocol': 'SSH', 'open': True}
- Zugriff: service['protocol'] oder service.get('protocol', 'Unbekannt')
- Methoden: service.keys() (Schlüssel), service.values() (Werte)
- Paare iterieren: for key, val in service.items():

8. MODULE

Module bieten vorgefertigte Bibliotheken für mathematische Operationen, Systemschnittstellen und Netzwerkzugriffe.



- Importieren: **import** math, sys oder **from** time **import** sleep
- Standardbibliothek: math für Berechnungen, sys für CLI-Argumente (sys.argv)
- Paketmanager **pip**: pip install paketname zur Installation externer Werkzeuge
- Namensraum: Zugriff auf Modulfunktionen über Punktnotation (z.B. math.sqrt(16))

PYTHON GRUNDLAGEN: ÜBERSICHT

Diese Mindmap fasst die 8 essenziellen Bausteine der Programmiersprache Python zusammen. Sie dient als kompakte Referenz für Einsteiger in die Cybersecurity und Skript-Entwicklung, um Syntax, Datenstrukturen und Kontrollflüsse auf einen Blick nachzuschlagen.

- Zentraler Einstiegspunkt für Skripterstellung und Tool-Verständnis
- 8 modulare Themenfelder mit Code-Snippets und Kernregeln
- Keine Vorkenntnisse vorausgesetzt; Fokus auf Lesbarkeit und praktische Syntax

Hinweis zur Python-Syntax

- Python nutzt Einrückungen (Standard: 4 Leerzeichen) zur Blockdefinition anstelle von geschweiften Klammern. Achte stets auf einheitliche Einrückungsebenen.