

# Dynamische Typisierung in Python

---

## Was bedeutet dynamische Typisierung?

In Python ist die **Typisierung dynamisch**, das heißt:

- Der **Datentyp einer Variable** wird **erst zur Laufzeit** festgelegt.
- Eine Variable kann **beliebige Datentypen** annehmen, ohne vorherige Deklaration.

Beispiel:

```
x = 10          # x ist vom Typ int
x = "Hallo"     # jetzt ist x vom Typ str
```

Python erkennt automatisch den Typ basierend auf dem Wert. Es ist **keine explizite Typ-Deklaration nötig**, wie in manchen anderen Programmiersprachen.

---

## Vergleich mit anderen Programmiersprachen

Sprache	Typisierung	Beispiel
Python	Dynamisch, stark	<code>x = 5</code> → <code>x = "Hallo"</code> ist erlaubt
Java	Statisch, stark	<code>int x = 5;</code> → <code>x = "Hallo";</code> ✗ Fehler
C	Statisch, schwach	<code>int x = 5;</code> → Typumwandlungen möglich
JavaScript	Dynamisch, schwach	<code>let x = 5;</code> <code>x = "Hallo";</code> erlaubt

---

## Vorteile der dynamischen Typisierung

- Kürzerer, flexibler Code
- Einfach für Anfänger
- Schnelles Prototyping möglich

## Nachteile

- Fehler durch unerwartete Typwechsel
  - Schwerer lesbarer Code bei komplexen Projekten
  - Keine Kompilierzeit-Prüfung auf Typfehler
- 

## Variablen und Datentypen in Python

Variable Name	Beispielwert	Datentyp (type)	Beschreibung
---------------	--------------	-----------------	--------------

---

Variable Name	Beispielwert	Datentyp (type)	Beschreibung
v_integer	5	int	Ganze Zahl
v_float	5.5	float	Kommazahl
v_string	"Python"	str	Zeichenkette
v_boolean	True	bool	Wahrheitswert
v_list	[1, 2, 3]	list	Liste von Elementen
v_tuple	(1, 2)	tuple	Unveränderliche Liste
v_dict	{"a": 1}	dict	Schlüssel-Wert-Paar Struktur
v_set	{1, 2, 3}	set	Menge mit einzigartigen Werten
v_none	None	NoneType	Kein Wert

---

## Hinweis zu Variablenbenennung

- Variablennamen sollten **aussagekräftig** sein
- Groß-/Kleinschreibung ist wichtig (Name ≠ name)
- Konstanten werden üblicherweise in **GROßBUCHSTABEN** geschrieben