

- Python je dynamicky typovaný jazyk, což znamená, že interpret je schopen automaticky určit datový typ proměnné na základě přiřazené hodnoty.
- V Pythonu se datové typy automaticky přizpůsobují hodnotám, které jsou jim přiřazeny, a umožňují dynamickou manipulaci s daty.

Základní datové typy:

1. **Integer (celé číslo):**
 - celá čísla
 - např. 5, -10
2. **Float (desetinné číslo):**
 - Reálná čísla s plovoucí desetinnou čárkou
 - např. 3.14, -0.001
3. **Boolean (logická hodnota):**
 - Reprezentuje pravdivostní hodnoty
 - např. True a False.
4. **String (řetězec znaků):**
 - Sekvence znaků
 - Imutabilní (neměnné)
 - např. "hello world" nebo 'Python je skvělý'.
 - *f string* - `f"calculation: {10 + 2 * 20}"`
5. **List (seznam):**
 - Uspořádaná, změnitelná (mutabilní) sekvence prvků.
 - Může obsahovat různé typy dat.
 - např. [1, 2, 3] nebo ['jablko', 'hruška', 'banán'].
6. **Tuple (n-tice):**
 - Podobný seznamu, ale je neměnný (immutable).
 - Může obsahovat různé typy dat.
 - např. (1, 2, 3).
7. **Range:**
 - Reprezentuje sekvenci čísel generovanou na základě počáteční, koncové hodnoty a kroku.
 - Používá se často v cyklech `for`.
8. **Dictionary (slovník):**
 - Neuspořádaná kolekce párů klíč-hodnota.
 - např. {'jmeno': 'Jan', 'vek': 30}.
9. **Set (množina):**
 - Neuspořádaná kolekce unikátních prvků.
 - např. {1, 2, 3}.
10. **NoneType:**
 - Speciální typ reprezentující absenci hodnoty.
 - Má jedinou hodnotu `None`

Vlastnosti datových typů v Pythonu:

1. **Dynamické přiřazení typů:** Python automaticky určuje datový typ proměnných na základě hodnoty, která je do nich přiřazena. Například:

```
x = 5          # x je nyní typu int
x = "hello"    # x je nyní typu str
```

2. **Silná typová kontrola:** Python provádí silnou typovou kontrolu, což znamená, že nelze provádět operace s datovými typy, které nejsou kompatibilní. Například nelze sčítat číslo a řetězec, pokud není provedena konverze.
3. **Dynamické rozšiřování:** Python umožňuje dynamicky rozšiřovat datové struktury, jako jsou seznamy a slovníky, přidáváním nových prvků nebo klíčů. Například:

```
list = [1, 2, 3] list.append(4) # Přidává prvek 4 na konec seznamu
```

4. **Metody a operátory:** Každý datový typ v Pythonu má své vlastní metody a operátory pro manipulaci s daty. Například metoda `append()` pro seznamy, nebo operátor `+` pro spojování řetězců.
5. **Dynamická alokace paměti:** Python spravuje paměť dynamicky, což znamená, že se stará o alokaci a dealokaci paměti pro proměnné a datové struktury.

Mutabilita

[!info] **mutabilní objekt** ... po jeho vytvoření umožňuje měnit jeho vnitřní stav **imutabilní objekt** ... po vytvoření NENÍ možné měnit vnitřní stav

- Proměnná v Pythonu je pouze odkazem na objekt, až samotný objekt je umístěn někde v paměti `####` Hodnota (value)
- Konkrétní kus dat, která jsou v objektu obsažena `####` Identity
- Unikátní identifikátor objektu (místo v paměti, kde "žije")

```
>>> id(42)
4343440904
```

Typ

- Typ nám říká z jaké třídy se objekt odvozuje

```
>>> type(42)
<class 'int'>
```

Imutabilní objekty

- `int`

- float
- complex
- str
- bytes
- tuple
- bool (odvozený i int)

Mutabilní objekty

- list
- dictionary
- set

Data Type	Built-in Class	Mutable
Numbers	int, float, complex	
Strings	str	
Tuples	tuple	
Bytes	bytes	
Booleans	bool	
Frozen sets	frozenset	
Lists	list	
Dictionaries	dict	
Sets	set	
Byte arrays	bytearray	