

# 4 Jednoduché datové typy

---

Slouží k určení typu hodnot uložených v proměnné. Podle typu hodnoty jdou na proměnnou aplikovat různé funkce nebo operátory.

## 1. Ordinální/Neordinální

- U ordinálních datových typů lze jednoznačně určit předchozí i následující hodnotu

### 1. Logická hodnota (`boolean`)

- Může být buď `True` nebo `False`

### 2. Celé číslo (`integer`)

- Kladná i záporná celá čísla
- Velikost čísla je omezená počtem bitů, které může zabírat v paměti

### 3. Znak (`char`)

- Jeden znak

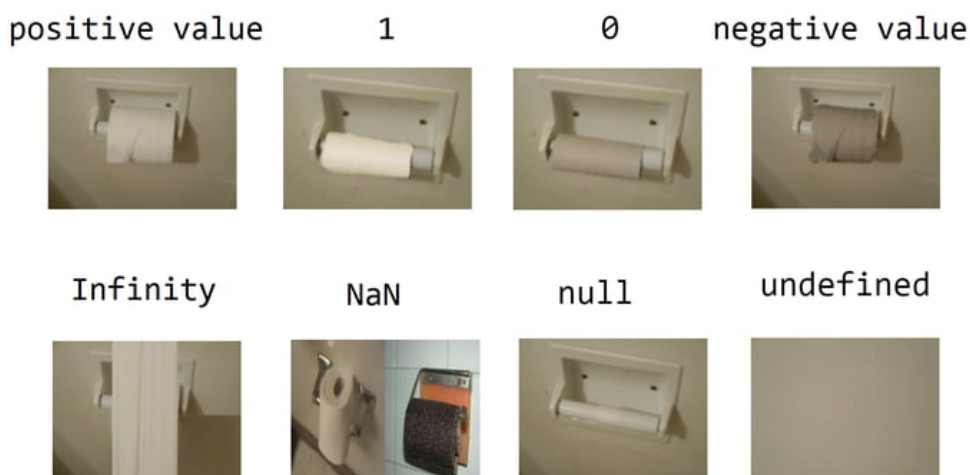
- U neordinálních čísel nelze jednoznačně určit předchozí a následující hodnotu

### 1. Reálná čísla (`float` nebo `double`)

- čísla s pohyblivou desetinnou čárkou (tečkou)
- Oproti celým číslům nejsou tak přesná u větších hodnot, ale jsou schopná uchovat mnohem větší číslo
- `double` je větší `float` (více bitů alokovaných v paměti)

### 2. Prázdné datové typy

- `void` -> většinou vyjadřuje, že funkce nevrací žádnou hodnotu
- `null` -> proměnná nemá (smysluplnou) hodnotu
- `none` -> většinou objektové programování, určuje, že objekt není



## 3. Pointer

- Ukazuje na místo v paměti, kde se nachází hodnota proměnné
- Například v linked listu ukazuje na následující hodnotu

#### 4. Pole

- Slouží k ukládání většího množství souvisejících hodnot
- Může mít více dimenzí
  - 1D list
    - `[ "a", "b", "c" ]`
    - Stačí jeden for loop
  - 2D list
    - `[ [ "a", "b" ], [ "c", "d" ] ]`
    - Jsou potřeba 2 for loop(y)
  - 3D list
    - `[ [ [ "a", "b" ], [ "c", "d" ] ], [ [ "e", "f" ], [ "g", "h" ] ] ]`
    - Jsou potřeba 3 for loop(y)

#### 2. Homogenní/Nehomogenní

##### 1. Homogenní

- Obsahují jednu nebo více hodnot stejného typu `string`

##### 2. Nehomogenní (heterogenní)

- Obsahují více hodnot různých typů `[ 1, "ahoj" ]`

#### 3. Typovaný/Netypovaný jazyk

##### 1. Netypovaný jazyk

- Často zaměňován s dynamicky typovanými jazyky
- Nemá žádné typy - všechno jsou jen bity

##### 2. Typovaný jazyk

###### 1. Dynamicky typovaný jazyk

- Při vytváření proměnné se neurčuje datový typ
- Datový typ se odvozuje podle hodnoty proměnné, proto jde například do původně `int` nastavit `string` atd...
- Například **python**

###### 2. Staticky typovaný jazyk

- Při vytváření proměnné jasně určíme datový typ
- Do proměnné se nedá vložit hodnota s jiným datovým typem
- Například **C**

```
# Vytváření staticky typovaných proměnných v pythonu
```

```
# 1. Logická hodnota
```

```
x = bool(True)
```

```
print(type(x))
```

```
# 2. Celé číslo
```

```
x = int(20)
```

```
print(type(x))
```

```
# 3. Python nemá char
```

```
# 4. Desetinné číslo
```

```
x = float(20.5)
```

```
print(type(x))
```

```
# 5. String
```

```
x = str("Hello World")
```

```
print(type(x))
```

```
# 6. Pole
```

```
x = list(("apple", "banana", "cherry"))
```

```
print(type(x))
```

```
#Python je dynamicky typovaný jazyk, proto mu nevadí to, že měníme hodnotu  
a datový typ proměnné
```