

# Proměnné

- Je to místo v úložišti identifikované adresou a asociované se symbolickým jménem
- Obsahuje nějaká data neboli hodnotu
- Tato hodnota může být přečtena či přepsána
- Jméno (na rozdíl od proměnných v matematice) může být dlouhé, aby vystihovalo užití

## Datové typy

- Různé jazyky mají různé datové typy

## Interpretované jazyky

- Mívají datový typ přidělovaný dynamicky podle potřeby
  - Například se z čísla může stát text
- Rozsah hodnot nebývá omezen
- Javascript:
  - Číslo – celá čísla, desetinná čísla, znaménko
  - Text (string) – ukládá textový řetězec
  - Boolean – True/False
- Python rozděluje čísla na celá a desetinná

## Kompilované jazyky

- Datový typ je pevně přiřazen proměnné
- Délka (v bytech) je také pevná
- Velikost typu určuje kompilér a může se lišit
- Číselné typy
  - byte – využito v Arduinu, celá čísla, 1 byte, bez znaménka, zkratka pro unsigned char
  - int – celá čísla, 4 byty, se znaménkem
  - float – desetinná čísla (vědecký tvar), 4byty
  - double – desetinná čísla, 8 bytů
  - Tyto základní typy mohou být modifikovány:
    - signed – se znaménkem
    - unsigned – bez znaménka
    - short – zmenšuje velikost
    - long – zvětšuje velikost
- Textové typy
  - char – ukládá jeden ASCII znak, 1 byte
    - Při potřebě stringu v C++ vytvoříme pole typu char
  - String – není datový typ, ale objekt
- Boolean
  - bool – ukládá True/False, 1 byte

## Objekt

- Samostatná datová entita
- Soubor proměnných a metod
- Jsou základem pro OOP (objektově orientované programování)

### Třídy a prototypy

- Objekt musíme nějak definovat
- Třída je definice formátu dat a procedur (objekt je poté instance třídy)
  - Třída se napíše pouze jednou
  - Objekt se vytvoří jako proměnná, kde datový typ je název třídy
  - Využívána spíše u kompilovaných jazyků (C++, C#)
- Prototyp je další objekt, se kterým je hlavní objekt spojen
  - Nové objekty se tvoří na základě existujících prototypů
  - Tyto dva objekty pak sdílí jeden prototyp
  - Využíván spíše u interpretovaných jazyků (JavaScript)

### Enkapsulace (modifikátory přístupu)

- Koncept OOP, který svazuje data a metody a chrání je od vnějšku objektu
- Data a metody uvnitř objektu nemusí být přístupná vnějšku
  - Nastaveno modifikátorem *private* (opak *public*)
  - Nepodporováno všemi jazyky (Python), tam se přístup může „upravovat“ jmennými konvencemi (například když název proměnné začíná podtržítkem)

### Dědičnost

- Třída (nebo objekt) dědí vlastnosti jiné třídy (nebo objektu)
- Specifikujeme při vytváření
- Vlastnosti dědicí třídy (nebo objektu) se dají rozšiřovat nebo upravovat