|  |  |
| --- | --- |
| **Operátory podmínek** | |
| bool | True/False |
| == | Rovnost |
| != | Nerovnost |
| > | Větší než |
| < | Menší než |
| >= | Větší nebo rovno |
| <= | Menší nebo rovno |
| ! (not) | Logická negace |
| || (or) | Logický součin |
| && (and) | Logický součet |

25. Podmínky a cykly: while, for, if v různých jazycích, pass, continue, break

**Větvení programu** = běh programu **různými cestami** (podmínkami)

**Podmínka** = vyhodnocení na základě logického výrazu bool (True/False)

**Blok příkazů** = co všechno se má udělat při splnění podmínky

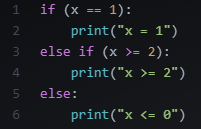
**V pythonu**:

* **za podmínky dvojtečka**
* blok příkazů je odsazen **mezerou**
* za jednotlivé příkazy se **nepíše nic**

**V C**:

* **za podmínky závorky**
* blok příkazůje ukončen další **závorkou,**
* za jednotlivé příkazy se píše **středník**

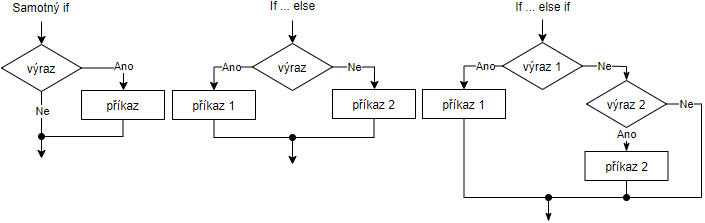
**if:**

* **True**: provede se příkaz
* **False**: program pokračuje dále

**if … else if:**

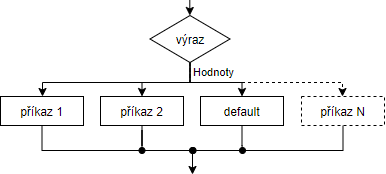
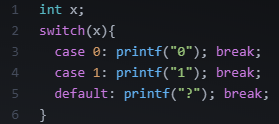
* **False**: další větvení podmínkou if
* V pythonu se může zapsat jako **elif**

**if … else:**

* **False**: provede se defaultní příkaz

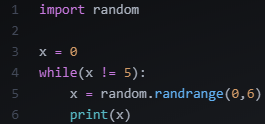
**switch:**

* **Není v pythonu**, přesto se dá vytvořit podmínkami if … else if … else if … else
* Vstup musí být stejného typu jako porovnávací hodnoty (string <--> string, int <--> int)
* Kontroluje **jestli vstup == hodnota**, neumí porovnávat intervaly
* Zápis v C:

****

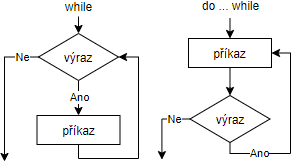
**Cyklus** = opakování běhu části programu

**Tělo cyklu** = příkazy uvnitř cyklu

**while**:

* Cyklus s **podmínkou na začátku**, **False před startem** -> cyklus nezačne
* **Nevíme kolikrát se příkaz udělá**
* **True:** provede příkaz
* **False:** ukončí cyklus

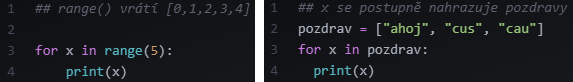
**do … while:**

* Cyklus s **podmínkou na konci**, **False před startem** -> příkaz se udělá jednou
* **Není v pythonu**, dal by se (neefektivně) vytvořit pomocí while … if … break
* **True:** provede příkaz
* **False:** ukončí cyklus

**for:**

* Cyklus s určeným počtem opakování
* Cyklus se **ukončí po vyčerpání všech prvků**

**Zápis v Pythonu:**

* **Funguje jinak** než v jiných jazycích, zapisuje se **for … in …**
* **Proměnná se nahrazuje** dalším číslem/písmenem posloupnosti
* Kolikrát se příkaz provede určujeme pomocí **range()** (vrací posloupnost jako array čísel)
* Pokud přiřadíme string/array, budou se příkazy opakovat dokud žádný nezbude

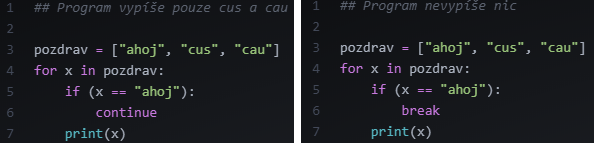
**Zápis v C:**

* 3 výrazy na začátku:
  + 1 -> **příkaz** **před prvním** provedením cyklu
  + 2 -> **podmínka** **před** **každým** provedením cyklu
  + 3 -> **příkaz po** provedení cyklu

**continue:**

* Přeskočí jeden krok cyklu

**break:**

* Okamžitě ukončí celý cyklus

**pass**:

* **Pouze v pythonu** (nikde jinde)
* Nic nedělá, jiný zápis **return Null**
* Když chceme vytvořit funkci/objekt, kterému **zatím nechceme nic přiřazovat**

