

Podmínky a cykly

- Upravují tok programu
- Podmínky nám větvi program
- Cykly vykonávají určitou část programu několikrát

Logický výraz

- Je pravdivý nebo nepravdivý
- Vždy vrátí jednu hodnotu, True nebo False
- Například: „Mají proměnné A a B stejnou hodnotu?“
- Program daný výraz vyhodnotí a vrátí True nebo False
- Operátory:
 - == → rovná se
 - > / >= → větší než / větší nebo rovno
 - < / <= → menší než / menší nebo rovno
 - !=, <> → nerovná se
 - &&, and → logické „a“ (logické násobení)
 - ||, or → logické „nebo“ (logické sčítání)
 - !, not → negace
 - in / not in → nachází se v / není v (využito v Pythonu, například “ah” in “ahoj”)
 - is / is not → je / není, testování identity objektu (například v Pythonu)
- Pokud je logický výraz složen pouze z jedné proměnné, pravdivost se určuje podle její hodnoty (pro různé jazyky odlišné)
 - NULL, undefined, atd. → False
 - Číslo 0 / jiné než 0 → False / True
 - Prázdný string / string s 1+ znakem → False / True
 - True / False → True / False

Podmínky

- Větvení programu na základě logického výrazu
- **If** – pokud je logický výraz pravda, provede se blok kódu spojený s podmínkou
- **Else** – vždy následuje po příkazu *if*; kód za *else* se vykoná, pokud podmínka u *if* je nepravdivá
- **Else if, elif, ...** - vyskytuje se po příkazu *if* a jeho kód se vykoná pouze pokud je předchozí podmínka nepravdivá a jeho podmínka je pravdivá
- **Switch** – větvení programu na základě hodnoty dané proměnné
- **Case** – vyskytuje se uvnitř switche a má přiřazenou hodnotu, kterou porovnává s proměnnou u switche
- **Default** – pokud hodnota proměnné není ani u jednoho case, vykoná se jeho kód

Cykly

- Opakují danou část kódu, dokud je podmínka pravdivá
- **For** – provede určitý počet iterací kódu
 - Tradiční – **for** *i* = *first* **to** *last* **do** *statement*
 - C++ → for (int i = 0; i < 10; i++)
 - Python → for i in range(10)

- S iterátorem – ***for item in iterable_object***
 - Někdy nazýván **Foreach**
 - C# → `foreach (string name in names)`
- While – dokud bude podmínka platit, bude vykonávat daný kód
 - Nejprve se vyhodnotí podmínka a poté se provede kód
- Do ... While - dokud bude podmínka platit, bude vykonávat daný kód
 - Nejprve se provede kód a poté se vyhodnotí podmínka

Úprava cyklů

- Pomocí klíčových slov můžeme cyklus ukončit předčasně nebo přeskočit na další opakování
- **Break** – „rozbije“ cyklus neboli ho ukončí
- **Continue** – ukončí aktuální opakování cyklu a započne následující
- **Pass** – využito v Pythonu
 - kvůli jeho syntaxi (blok kódu není oddělen závorkami, ale odsazením) nemůžeme vytvořit prázdnou funkci, podmínku, cyklus nebo třídu pro pozdější doplnění
 - při potřebě prázdného bloku kódu využijeme klíčové slovo *pass*