

# Řídící struktury

**Řídící struktura** (též strukturovaný příkaz, anglicky control flow statement) je v informatice konstrukce pro zápis počítačového programu. Řídící struktury jsou používány ve vyšších programovacích a skriptovacích jazycích. Rozhodují o dalším provádění programu: větví jeho běh, vytváří cykly nebo jinak mění běh programu.

## Větvení programu:

Větvení nastane v momentě, kdy se program liší podle splnění nebo nesplnění nějaké podmínky.

### **Větvení se skládá ze tří částí:**

1. otázka, na kterou jde odpovědět ano, nebo ne
2. první možnost pokračování programu (odpověď zní ano)
3. druhá možnost pokračování programu (odpověď zní ne)

Vícenásobné větvení je pak případ, kdy je nutné mít více možností pro rozhodování – větev je pak vybrána podle hodnoty podmínky (provede se ta větev, která hodnotě podmínky odpovídá).

## Cykly

♦ Cyklus se skládá z posloupnosti příkazů a podmíněného skoku, pomocí kterého se cyklus ukončuje při splnění podmínky.

### 1. Existují tyto druhy cyklů:

1. **nekonečný cyklus** – za normálních okolností není vůbec ukončen
  2. **while-do cyklus** – cyklus s podmínkou na začátku posloupnosti příkazů
  3. **do-while cyklus** – cyklus s podmínkou na konci posloupnosti příkazů
  4. **cyklus s testem podmínky uprostřed posloupnosti příkazů** – řídce užívaná varianta
  5. **for cyklus** – speciální případ cyklu s podmínkou na začátku, obvykle užívaný pro výčet prvků z množiny prvků (např. interval celých čísel  $<1,10>$ ). V některých jazycích je počet opakování vyhodnocen jednou na začátku a další změna této podmínky nemá na počet opakování vliv. V ostatních jazycích je for cyklus vlastně zvláštním případem while-do cyklu:
- Hodně jazyků umožňuje násilné přerušení cyklu (vyskočení ven z cyklu, v

Pascalu a v C to je **break**), je tedy možné vytvořit například nekonečný cyklus, uvnitř tohoto cyklu testovat nějakou podmínku a ve vhodné situaci cyklus přerušit.

- Cykly se ukončí na základě podmínky. Cykly s **While** probíhají, pokud je podmínka splněna, cykly s **Until** probíhají až do okamžiku splnění podmínky.
- Cykly s podmínkou na začátku nemusí proběhnout vůbec, cykly s podmínkou nakonci proběhnou aslepoň jednou.
- Cyklus s **While** na začátku cyklu je téměř totožný s předcházejícím cyklem **While**. Jediný rozdíl je, že tyto čtyři cykly mohou být ukončeny v jejich průběhu příkazem **Exit Do**.

### Prostředky větvení

- **IF** = podmínka
  - V programování se "if" používá k provádění určitých akcí na základě hodnoty logického výrazu (pravdivosti) nebo podmínky. Je to základní prvek v mnoha programovacích jazycích, který umožňuje větvit kód na základě určitých podmínek.
  - If-else konstrukce jsou základem pro řízení toku programu a umožňují programátorům definovat různé akce na základě různých situací. Tyto konstrukce jsou nezbytné pro vytváření adaptabilních a dynamických programů.
- **SWITCH** = hledá jestli je case == a
  - Ve světě programování je "switch" (přepínač) struktura, která umožňuje rozhodovat mezi různými částmi kódu na základě hodnoty určité proměnné nebo výrazu. Switch se používá, když máte řadu možných hodnot, které chcete porovnat s konkrétní proměnnou a provést různé akce podle toho, jaká hodnota se shoduje.
  - Když se vyhodnotí switch, program zjistí hodnotu výrazu a porovná ji s hodnotami uvedenými v case případech. Pokud se nějaký případ shoduje, provede se kód uvnitř tohoto případu. Pokud se žádný případ neodpovídá hodnotě výrazu, provede se kód uvnitř bloku "default", pokud je uveden.
  - Použití switch může být efektivní pro situace, kdy máte mnoho možných větví podmínek závislých na jedné proměnné. Switch může často udělat kód přehlednějším a snadnějším na čtení než velké

množství vnořených if-else podmínek.

## Prostředky cyklické

**FOR** = Cyklický prostředek "for" v programování je základní smyčka, která umožňuje opakování určitého bloku kódu určený početkrát. Typicky se používá, když znáte počet opakování předem. Syntaxe "for" smyčky se obvykle skládá z tří částí:

- **Inicializace:** Inicializuje se proměnná, která bude použita jako čítač pro sledování průběhu smyčky.
- **Podmínka:** Udává podmínku, za které má být smyčka prováděna. Pokud je tato podmínka splněna, smyčka se provádí; pokud ne, smyčka se ukončí.
- **Inkrementace/dekrementace:** Specifikuje, jak se má proměnná měnit každou iterací smyčky.
- 
- Cyklický prostředek "for" je mocný nástroj pro opakování stejného kódu po určitý početkrát a je široce používán ve všech druzích programovacích jazycích.

**WHILE** = cyklus, který nemá předem daný počet kolikrát se zopakuje

♦ Cyklický prostředek "while" je základní smyčkou (loop) v programování, která umožňuje opakovat určitý blok kódu, dokud je splněna určitá podmínka. To znamená, že kód v těle while smyčky se opakovaně vykonává, dokud je podmínka vyhodnocena jako pravdivá (true)

♦ Je důležité zajistit, aby podmínka ve smyčce "while" měla nějaký způsob, jak být splněna. Pokud se podmínka nikdy nestane nepravdivou (false), může to vést k tzv. "nekonečné smyčce", což znamená, že program bude pokračovat v nekonečné opakování stejného bloku kódu, což může vést k zablokování programu.

Řídicí struktury v programování jsou konstrukce, které určují tok řízení v programu. Umožňují programům přijímat rozhodnutí, opakovat akce a řídit pořadí, v jakém se příkazy provádějí. Existují tři hlavní typy řídicích struktur:

## 1. Podmíněné struktury (nebo podmínky):

- Umožňují provádění různých bloků kódu na základě splnění určitých podmínek.
- Nejčastěji používané konstrukce jsou:
  - `if` / `else`: Základní podmíněná struktura.
  - `switch` / `case`: Alternativa k více `if` / `else` příkazům, která může být přehlednější.

## Cykly (nebo smyčky):

- Umožňují opakované provádění bloku kódu, dokud je splněna určitá podmínka.
- Hlavní typy cyklů jsou:
  - `for`: Opakuje blok kódu pevně daný početkrát.
  - `while`: Opakuje blok kódu, dokud je splněna podmínka.
  - `do...while` (v některých jazycích, např. C nebo JavaScript): Zajišťuje, že blok kódu se provede alespoň jednou a poté se opakuje, dokud je splněna podmínka.

## Skoky:

- Umožňují přerušení běhu programu a přesun na jiné místo v kódu.
- Patří sem:
  - `break`: Ukončí nejbližší cyklus.
  - `continue`: Přeskočí zbytek aktuální iterace a pokračuje s další iterací cyklu.
  - `return`: Ukončí funkci a vrátí hodnotu.
  - `goto` (v některých jazycích, např. C): Přeskočí na označené místo v kódu. Používá se velmi zřídka, protože může vést k obtížně udržovatelnému kódu.

Řídící struktury jsou základem pro vytváření složitějších algoritmů a programů, protože umožňují dynamické rozhodování a opakování.

V Pythonu existují tři hlavní prostředky pro větvení programu, které umožňují provádět různé akce na základě podmínek: `if`, `elif`, a `else`. Jejich základní funkcí je

umožnit programu rozhodovat se a vykonávat určité bloky kódu pouze tehdy, jsou-li splněny specifické podmínky.

### `if` Příkaz

Příkaz `if` testuje podmínku a pokud je tato podmínka pravdivá, vykoná následující blok kódu.

### `elif` Příkaz

Příkaz `elif` (zkratka pro "else if") umožňuje testovat další podmínku, pokud předchozí `if` nebo `elif` podmínka nebyla pravdivá.

### `else` Příkaz

Příkaz `else` zachytí všechny ostatní případy, kdy žádná z předchozích podmínek `if` nebo `elif` nebyla splněna.

## Základní funkce podmíněných příkazů

- `if`: Zajišťuje, že určitý blok kódu je vykonán pouze tehdy, je-li podmínka pravdivá.
- `elif`: Umožňuje kontrolovat další podmínky, pokud předchozí `if` nebo `elif` podmínky nebyly pravdivé.
- `else`: Vykonává blok kódu, pokud žádná z předchozích podmínek nebyla pravdivá.

Používáním těchto prostředků můžete vytvářet složité logické větvení v Pythonu a řídit tok vašeho programu na základě různých podmínek