

CONDITIONALS & ITERATION

ŘÍZENÍ TOKU PROGRAMU V PYTHONU

CONDITIONALS & ITERATION

ŘÍZENÍ TOKU PROGRAMU V PYTHONU

if • match • for • while • itertools

PROČ JE TO DŮLEŽITÉ?

PROČ JE TO DŮLEŽITÉ?

- Rozhodování (podmínky)

PROČ JE TO DŮLEŽITÉ?

- Rozhodování (podmínky)
- Opakování práce (cykly)

PROČ JE TO DŮLEŽITÉ?

- Rozhodování (podmínky)
- Opakování práce (cykly)
- Čistý a čitelný kód

IF / ELIF / ELSE

```
1 age = 18
2
3 if age >= 18:
4     print("Dospělý")
5 elif age >= 15:
6     print("Téměř")
7 else:
8     print("Dítě")
```

VNOŘENÉ PODMÍNKY (NESTED IF)

```
1 if user_logged:  
2     if is_admin:  
3         print("Admin panel")
```


VNOŘENÉ PODMÍNKY (NESTED IF)

```
1 if user_logged:  
2     if is_admin:  
3         print("Admin panel")
```

👉 raději zjednodušovat a používat guard clauses

TERNÁRNÍ OPERÁTOR

```
1 status = "OK" if value > 0 else "FAIL"
```

TERNÁRNÍ OPERÁTOR

```
1 status = "OK" if value > 0 else "FAIL"
```

Používat jen pro jednoduché výrazy

PATTERN MATCHING (MATCH)

```
1 cmd = "start"
2
3 match cmd:
4     case "start":
5         run()
6     case "stop":
7         stop()
8     case _:
9         print("Unknown")
```

FOR CYKLUS

```
1 for i in range(5):  
2     print(i)
```

RANGE()

```
1 range(5)           # 0..4
2 range(1, 5)        # 1..4
3 range(0, 10, 2)     # krok 2
```

ITERACE PŘES SEKVENCÍ

```
1 for name in ["Alice", "Bob", "Eve"]:  
2     print(name)
```

ITERABLE VS ITERATOR

ITERABLE VS ITERATOR

- Iterable → lze projít (list, str, dict)

ITERABLE VS ITERATOR

- Iterable → lze projít (list, str, dict)
- Iterator → má `__next__()`

VÍCE SEKVENCÍ NAJEDNOU

```
1 names = ["A", "B"]
2 scores = [10, 20]
3
4 for n, s in zip(names, scores):
5     print(n, s)
```

WHILE CYKLUS

```
1 i = 0
2 while i < 5:
3     print(i)
4     i += 1
```

BREAK & CONTINUE

```
1 for x in data:  
2     if x == 0:  
3         continue  
4     if x < 0:  
5         break
```

ELSE U FOR/WHILE

```
1 for x in nums:
2     if x == 5:
3         break
4 else:
5     print("nenalezeno")
```

WALRUS OPERÁTOR :=

```
1 while (line := file.readline()):  
2     print(line)
```

ITERTOOLS

Andrej Pčelovodov

ITERTOOLS

- chain

ITERTOOLS

- chain
- count

ITERTOOLS

- chain
- count
- cycle

ITERTOOLS

- chain
- count
- cycle
- islice

KOMBINATORICKÉ GENERÁTORY

```
1 from itertools import permutations, combinations
2
3 permutations([1,2,3])
4 combinations([1,2,3], 2)
```

SHRNUTÍ

SHRNUTÍ

- if / match → rozhodování

SHRNUTÍ

- if / match → rozhodování
- for / while → opakování

SHRNUTÍ

- if / match → rozhodování
- for / while → opakování
- itertools → výkon a elegance