

Modulárny program

Poznámky k téme

Väčší program sa nepíše ako jeden dlhý blok príkazov. Rozdeľuje sa na menšie časti – **funkcie** a **moduly** –, z ktorých každá rieši jednu úlohu. Python navyše obsahuje veľa hotových modulov (*štandardnú knižnicu*): na matematiku, náhodné čísla, prácu s časom a mnoho ďalšieho. V tejto téme sa naučíš moduly používať, vytvoriť si vlastný modul a rozložiť program na prehľadné časti.

Čo sa naučíš

- importovať modul a používať jeho funkcie (`import`, `from ... import`),
- pracovať s modulmi `math`, `random` a `time`,
- vytvoriť vlastný modul a rozumieť zápisu `if __name__ == "__main__":`,
- rozložiť program na funkcie s jasnými úlohami.

1 Import modulu

Modul je súbor s funkciami a hodnotami, ktoré môžeme použiť v našom programe. Najprv ho musíme **importovať** – zvyčajne na začiatku programu. K jeho obsahu sa potom dostaneme cez meno modulu a bodku:

```
import math

print(math.sqrt(16))
print(math.pi)
print(math.floor(3.7), math.ceil(3.2))
```

Výstup:

```
4.0
3.141592653589793
3 4
```

1.1 Spôsoby importu

Zápis	Použitie v programe
<code>import math</code>	<code>math.sqrt(2)</code> , <code>math.pi</code>
<code>from math import sqrt, pi</code>	<code>sqrt(2)</code> , <code>pi</code> – bez mena modulu
<code>import math as m</code>	<code>m.sqrt(2)</code> – modul pod kratším menom
<code>from math import *</code>	všetko bez mena modulu – neodporúča sa

```
from math import sqrt, pi
import random as r

print(round(sqrt(2), 4), round(pi, 4))
r.seed(1)
print(r.randint(1, 6))
```

Výstup:

```
1.4142 3.1416
2
```

Prvý spôsob (`import math`) je najprehľadnejší: pri každom použití vidno, odkiaľ funkcia pochádza. Zápis `from math import *` prinesie do programu desiatky mien naraz a môže nechtiac prepísať tvoje vlastné premenné.

Tip: Čo modul obsahuje, zistíš v interaktívnom režime príkazom `dir(math)`; popis funkcie ukáže `help(math.sqrt)`.

2 Modul math

Funkcia / hodnota	Význam	Príklad	Výsledok
<code>math.sqrt(x)</code>	druhá odmocnina	<code>math.sqrt(25)</code>	5.0
<code>math.pi</code> , <code>math.e</code>	konštanty π a e	<code>math.pi</code>	3.141592653589793
<code>math.floor(x)</code>	zaokrúhlenie nadol	<code>math.floor(-2.5)</code>	-3
<code>math.ceil(x)</code>	zaokrúhlenie nahor	<code>math.ceil(2.1)</code>	3
<code>math.gcd(a, b)</code>	najväčší spoločný deliteľ	<code>math.gcd(48, 18)</code>	6
<code>math.factorial(n)</code>	faktoriál	<code>math.factorial(5)</code>	120
<code>math.sin(x)</code> , <code>math.cos(x)</code>	goniometrické funkcie (uhol v radiánoch)	<code>math.sin(math.pi / 2)</code>	1.0
<code>math.radians(st)</code>	stupne na radiány	<code>math.radians(180)</code>	3.141592653589793
<code>math.hypot(a, b)</code>	prepona: $\sqrt{a^2 + b^2}$	<code>math.hypot(3, 4)</code>	5.0

```
>>> import math
>>> math.sqrt(25)
5.0
>>> math.floor(-2.5)
-3
>>> math.factorial(5)
120
>>> round(math.sin(math.radians(30)), 4)
0.5
>>> math.hypot(3, 4)
5.0
```

Goniometrické funkcie pracujú s uhlami v **radiánoch**: uhol v stupňoch najprv preved' funkciou `math.radians`.

3 Modul random

Modul `random` vytvára (pseudo)náhodné čísla – hodí sa na hry, simulácie a generovanie testovacích údajov.

Funkcia	Čo vráti
<code>random.randint(a, b)</code>	náhodné celé číslo od <code>a</code> po <code>b</code> vrátane
<code>random.random()</code>	náhodné desatinné číslo od 0 (vrátane) po 1 (bez 1)
<code>random.uniform(a, b)</code>	náhodné desatinné číslo medzi <code>a</code> a <code>b</code>
<code>random.choice(z)</code>	náhodný prvok zoznamu (alebo znak reťazca)
<code>random.shuffle(z)</code>	náhodne premieša zoznam (mení ho)
<code>random.sample(z, k)</code>	<code>k</code> rôznych náhodných prvkov
<code>random.seed(n)</code>	nastaví „semienko“ – potom sú „náhodné“ čísla zakaždým rovnaké

```
import random

random.seed(42)
print(random.randint(1, 6), random.randint(1, 6), random.randint(1, 6))
print(round(random.random(), 3))
print(random.choice(["kameň", "papier", "nožnice"]))
karty = ["A", "K", "Q", "J", "10"]
random.shuffle(karty)
print(karty)
print(sorted(random.sample(range(1, 50), 6)))
```

Výstup:

```
6 1 1
0.742
kameň
['J', 'A', 'Q', '10', 'K']
[6, 28, 35, 38, 44, 48]
```

Počítač náhodu iba napodobňuje: čísla vznikajú výpočtom zo „semienka“ (`seed`). Bez `random.seed` sa semienko nastaví podľa času, takže každé spustenie dá iné čísla. S pevným semienkom dostaneš pri každom spustení **rovnakú** postupnosť – to sa hodí pri hľadaní chýb a testovaní (preto ho používame aj v ukážkach v týchto poznámkach).

4 Modul time

```
import time

zaciatok = time.time()
sucet = 0
for i in range(1_000_000):
    sucet += i
trvanie = time.time() - zaciatok
print("Súčet:", sucet)
print("Výpočet trval menej ako sekundu:", trvanie < 1)
```

Výstup:

```
Súčet: 499999500000
Výpočet trval menej ako sekundu: True
```

`time.time()` vráti počet sekúnd od roku 1970 – rozdiel dvoch takýchto časov je trvanie výpočtu.
`time.sleep(s)` zastaví program na `s` sekúnd (hodí sa napríklad pri jednoduchých animáciách v textovom režime).

5 Vlastný modul

Vlastný modul je obyčajný súbor `.py` s funkciami. Uložíme ho do **toho istého priečinka** ako program, ktorý ho používa:

Súbor `geometria.py`:

```
"""Výpočty pre rovinné útvary."""
PI = 3.141592653589793

def obsah_kruhu(r):
    return PI * r ** 2

def obvod_kruhu(r):
    return 2 * PI * r

def obsah_obdlznika(a, b):
    return a * b
```

Program ho importuje podľa mena súboru (bez `.py`):

```
import geometria

print(round(geometria.obsah_kruhu(2), 2))
print(round(geometria.obvod_kruhu(2), 2))
print(geometria.obsah_obdlznika(3, 4))
```

Výstup:

```
12.57
12.57
12
```

5.1 if __name__ == "__main__"

Pri importe Python **vykoná celý súbor modulu**. Ak by sme v module mali aj skúšobné príkazy (`print(obsah_kruhu(1))`), vypísali by sa pri každom importe. Riešením je podmienka:

Súbor `kocka.py`:

```
import random

def hod():
    return random.randint(1, 6)

if __name__ == "__main__":
    # vykoná sa iba pri priamom spustení súboru kocka.py
    print("Skúšobný hod:", hod())
```

Premenná `__name__` má hodnotu `"__main__"` iba vtedy, keď súbor **spúšťame priamo**. Pri importe obsahuje meno modulu (`"kocka"`), takže sa skúšobná časť nevykoná:

```
import kocka

print("Hody:", [kocka.hod() for _ in range(5)])
```

Výstup:

```
Hody: [1, 1, 1, 3, 2]
```

Rovnaký zápis používajú aj úlohy na Mente, ktoré testujú tvoje funkcie: hlavný program (načítanie vstupu a výpis) patrí pod `if __name__ == "__main__":`, aby ho testy pri volaní funkcií nespúšťali.

Pozor: Nepomenuj vlastný súbor rovnako ako štandardný modul (`random.py`, `math.py`, `time.py`). Príkaz `import random` by potom načítal tvoj súbor namiesto skutočného modulu a funkcie ako `randint` by „neexistovali“.

6 Rozklad programu na funkcie

Pri väčšej úlohe sa oplatí postupovať **zhora nadol**: najprv rozdeliť úlohu na menšie podúlohy, každú podúlohu zapísať ako funkciu a hlavný program poskladať z volaní týchto funkcií. Výhody:

- každá funkcia je krátka a dá sa samostatne vyskúšať,
- hlavný program sa číta ako obsah: čo sa robí, v akom poradí,
- chyba sa hľadá v malom kúsku kódu, nie v stovkách riadkov,
- funkciu možno použiť opakovane (aj v inom programe – cez modul).

Príklad: vyhodnotenie testu. Podúlohy: načítať body, vypočítať percentá a známku, vypísať správu.

```
MAXIMUM = 50

def nacitaj():
    meno = input("Meno: ")
    body = int(input("Body: "))
    return meno, body

def percenta(body):
    return body / MAXIMUM * 100

def znamka(p):
    hranice = [(90, 1), (75, 2), (50, 3), (30, 4)]
    for hranica, z in hranice:
        if p >= hranica:
            return z
    return 5

def vypis(meno, p, z):
    print(f"{meno}: {p:.1f} % – známka {z}")

def main():
    meno, body = nacitaj()
    p = percenta(body)
    vypis(meno, p, znamka(p))

if __name__ == "__main__":
    main()
```

Výstup:

```
Meno: Ema
Body: 43
Ema: 86.0 % – známka 2
```

Hlavný program je zabalený do funkcie `main` – vďaka tomu nemá globálne premenné a celý „príbeh“ programu je na jednom mieste. Funkcie `percenta` a `znamka` iba počítajú (nič nevypisujú ani nečítajú), takže sa dajú použiť aj inde.

7 Riešené úlohy

7.1 Hod mincou

Zadanie: Simuluj 1000 hodov mincou a vypíš, koľkokrát padol znak a koľkokrát hlava, aj v percentách.

```
import random

znak = 0
for _ in range(1000):
    if random.choice(["znak", "hlava"]) == "znak":
        znak += 1
hlava = 1000 - znak
print(f"znak: {znak} ({znak / 10:.1f} %)")
print(f"hlava: {hlava} ({hlava / 10:.1f} %)")
```

Výstup:

```
znak: 512 (51.2 %)
hlava: 488 (48.8 %)
```

Pri veľa hodoch sa podiel blíži k 50 %, ale takmer nikdy nevyjde presne.

7.2 Histogram hodov kockou

Zadanie: Hod' 60-krát kockou a pre každú stranu vypíš riadok s toľkými hviezdikami, koľkokrát padla.

```
import random

pocety = [0] * 7          # indexy 1 až 6 (index 0 nepoužijeme)
for _ in range(60):
    pocety[random.randint(1, 6)] += 1
for strana in range(1, 7):
    print(strana, "*" * pocety[strana], pocety[strana])
```

Výstup:

```
1 ***** 10
2 ***** 11
3 ***** 16
4 ***** 8
5 ***** 9
6 ***** 6
```

7.3 Generátor hesiel

Zadanie: Vygeneruj náhodné heslo dĺžky 10 z malých a veľkých písmen a číslíc.

```
import random
import string

znaky = string.ascii_letters + string.digits
heslo = "".join(random.choice(znaky) for _ in range(10))
print(heslo)
```

Výstup:

```
ox9yimTcfi
```

Modul `string` obsahuje hotové reťazce znakov: `string.ascii_lowercase` (a–z), `string.ascii_uppercase` (A–Z), `string.digits` (0–9). Na skutočné heslá sa používa modul `secrets`, ktorého náhoda sa nedá predpovedať.

8 Časté chyby

- **Chýbajúci import:** `NameError: name 'math' is not defined` – na začiatok patrí `import math`.
- **Chýbajúce meno modulu:** po `import math` treba písať `math.sqrt`, nie `sqrt` (alebo použiť `from math import sqrt`).
- **Vlastný súbor s menom štandardného modulu** (`random.py`) – premenuj ho.
- **Modul v inom priečinku** – vlastný modul musí byť v priečinku programu.
- **Celý program v jednom bloku** – ťažko sa číta aj opravuje; rozdeľ ho na funkcie.

Zhrnutie

- `import modul` a potom `modul.meno`; `from modul import meno` (bez predpony); `import modul as m` (alias).
- `math`: `sqrt`, `pi`, `floor`, `ceil`, `gcd`, `factorial`, goniometria v radiánoch.
- `random`: `randint` (vrátane hraníc), `random`, `choice`, `shuffle`, `sample`, `seed`.
- Vlastný modul = súbor `.py` v priečinku programu; import podľa mena súboru.
- `if __name__ == "__main__":` – kód, ktorý sa vykoná iba pri priamom spustení, nie pri importe.
- Rozklad na funkcie: jedna funkcia = jedna podúloha; hlavný program (`main`) iba volá funkcie.

Úlohy na precvičenie

1. Vypočítaj dĺžku prepony pravouhlého trojuholníka s odvesnami 5 a 12 dvoma spôsobmi (`math.sqrt` a `math.hypot`).
2. Simuluj 10 000 hodov dvoma kockami a zisti, ktorý súčet padá najčastejšie.
3. Vytvor modul `texty.py` s funkciami `pocet_samohlasok(s)` a `je_palindrom(s)` a použi ich v inom programe.
4. Napíš hru Kameň, papier, nožnice proti počítaču na 5 kôl, rozloženú na funkcie (voľba počítača, vyhodnotenie kola, výpis skóre).
5. Zmeraj, ako dlho trvá zoradenie zoznamu milióna náhodných čísel funkciou `sorted`.

Úlohy k tejto téme nájdeš na learnwithmente.com v časti Informatika → Python.