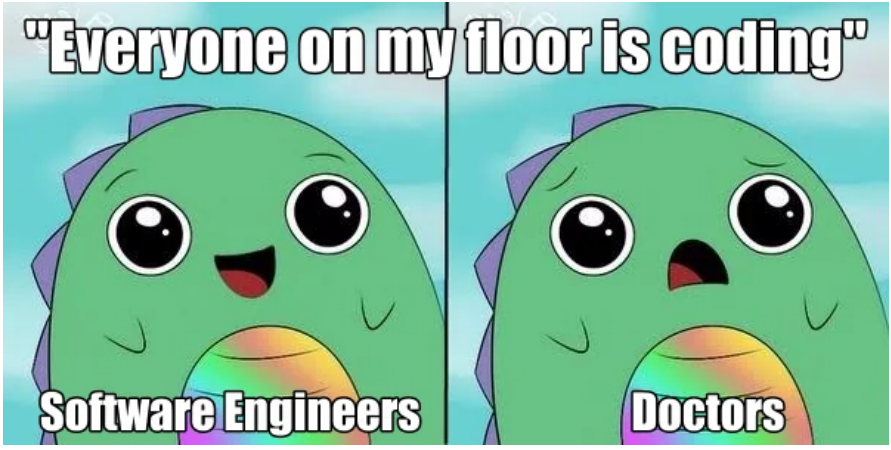
Výpočetní technika a lékařská informatika

Programování

Mgr. Markéta Trnečková, Ph.D.

Programování

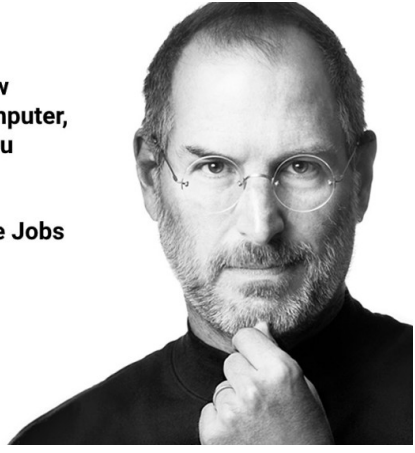


Proč programování pro mediky?

- Moderní medicína využívá data – analýza, vizualizace, automatizace
- Výzkum: bioinformatika, statistika, modelování
- Klinická praxe: skriptování, zpracování dat z přístrojů
- Osobní rozvoj: logické myšlení, strukturované řešení problémů

"Everyone should know how to program a computer, because it teaches you how to think."

Steve Jobs



Co je programování?

- Způsob, jak říct počítači, co má dělat
- Řešení problému krok za krokem
- Přemýšlení v malých logických blocích
- Zápis pomocí programovacího jazyka (např. Python)

Už jsme programovali

- V Excelu jsme psali funkce jako `=IF()`, `=SUM()`, `=VLOOKUP()`
- Používali jsme logiku: `IF BMI > 25 THEN "Nadváha"`
- Řešili jsme problémy krok za krokem
- To všechno jsou základy programování!

Co je Python?

- Python je moderní programovací jazyk
- Používá se ve vědě, medicíně, výzkumu, automatizaci i vývoji aplikací
- Je jednoduchý na naučení – čitelný, přehledný
- Má obrovskou komunitu a množství knihoven (např. pro práci s daty, grafikou, AI)

Jak vypadá kód v Pythonu?

- **Proměnná:**

```
x = 5
```

- **Podmínka:**

```
if x > 3:
    print("Vetsi nez 3")
```

- **Cyklus:**

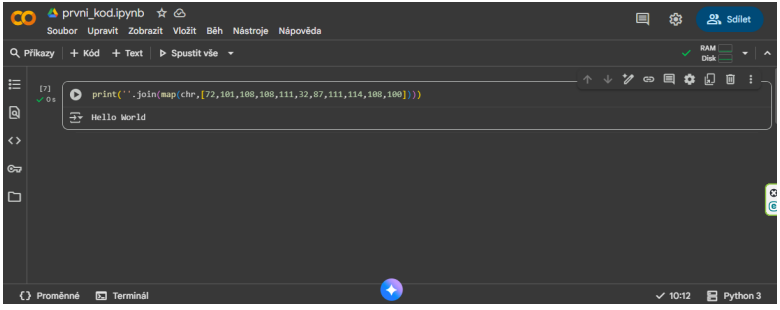
```
for i in range(5):
    print(i)
```

- **Funkce:**

```
def bmi(vaha, vyska):
    return vaha / (vyska ** 2)
```

Google Colab – jak začít

- Otevřete stránku: `https://colab.research.google.com`
- Přihlaste se pomocí svého Google účtu
- Klikněte na **"New Notebook"** (Nový sešit)



První krok v Colabu

Součet

- Do buňky napiš následující kód:

```
1+6
```

- Vyhodnoť buňku pomocí ▶ (Stiskni **Shift + Enter**)
- Výsledek se zobrazí pod buňkou

- Je možno přidávat více buněk a každou vyhodnocovat zvlášť, nebo `Spustit vše`
- Budeme každý úkol psát do nové buňky.

Proměnné a hodnoty

- Proměnná je pojmenované místo, kam uložíme hodnotu
- Hodnota může být číslo, text, logická hodnota, atd.
- Proměnnou vytvoříme jednoduše:

```
a = 1
b = 6
```

- Proměnnou můžeme kdykoliv použít při dalších výpočtech:

```
c = a + b
```

- Nebo vypsat pomocí `print()`:

```
print(a)
```

Aritmetické operace a základní funkce

- **Základní operace:**

```
a = 10
b = 3

soucet = a + b          # scitani
rozdil = a - b          # odcitani
soucin = a * b          # nasobeni
podil = a / b           # deleni (vysledek je desetinny)
zbytek = a % b          # zbytek po deleni
mocnina = a ** b        # umocneni
```

- **Základní funkce:**

```
round(3.14159)          # zaokrouhleni
abs(-5)                 # absolutni hodnota
max(4, 7, 2)            # nejvetsi hodnota
min(4, 7, 2)            # nejmensi hodnota
```

Text za # je komentář a nevyhodnocuje se.

Úkol

Příklad

Vytvořte proměnnou `jmeno`.
Napište kód, který vypočítá BMI (`bmi`) z proměnných `vaha` a `vyska`.
Vypište výsledek:

```
print("BMI pro", jmeno, "je", bmi)
```

Vyzkoušejte si měnit hodnoty proměnných.

Podmínky (větvení)

- Pomocí podmínky můžeme rozhodnout, co se má stát
- Používáme klíčové slovo `if`, případně `else`
- Příklad:

```
bmi = 27

if bmi > 25:
    print("Nadvaha")
else:
    print("BMI v norme")
```

- Můžeme přidat i další větvení pomocí `elif`:

```
if bmi < 18.5:
    print("Podvaha")
elif bmi < 25:
    print("Normalni vaha")
else:
    print("Nadvaha")
```

Všimněte si, jak se zapisuje tělo jednotlivých větví (pomoc odsazení).

Úkol

Příklad

Doplňte předchozí kód o výpis zda má člověk podváhu, normální váhu, nadváhu nebo obezitu. Vyzkoušejte si měnit hodnoty proměnný a sledujte, jak se mění výpis.

Složené podmínky – logické spojky

- Můžeme spojovat více podmínek pomocí:
  - `and` – obě podmínky musí být pravdivé
  - `or` – stačí, aby jedna byla pravdivá
  - `not` – negace (nepravda)
- Příklad:

```
vek = 65
tlak = 145

if vek > 60 and tlak > 140:
    print("Rizikovy pacient")
```

- Podmínky můžeme skládat do závorek pro přehlednost.

Úkol

Příklad

Vytvořte tři proměnné `a`, `b` a `c`.
Napište kód, který vypíše nejmenší hodnotu z těchto proměnných.
Změnou hodnot proměnných vyzkoušejte, zda kód funguje správně.

Zkuste přidat čtvrté číslo a upravit podmínky.

Ternární operátor – zkrácená podmínka

- Ternární operátor umožňuje zapsat jednoduchou podmínku na jeden řádek
- Syntaxe:

```
vysledek = hodnota1 if podminka else hodnota2
```

- Příklad:

```
vek = 17
status = "dospely" if vek >= 18 else "nezletily"
print(status)
```

- Výsledek: `nezletily`

Příklad

Jak by vypadala zkrácená podmínka pro rozhodování, které ze dvou čísel je menší?

Cykly – opakování příkazů

- Cykly umožňují opakovat určitou činnost vícekrát
- **For cyklus** – opakuje přes daný rozsah:

```
for i in range(5):
    print("i je:", i)
```

- `range(n)` vytvoří čísla od 0 do `n - 1`
- Můžeš zadat i začátek a krok `range(zacatek, konec, krok)`
- Vytváří posloupnost čísel od `zacatek` do `konec - 1` s krokem `krok`

Příklad

Napište kód který vypíše všechna sudá čísla menší než 20.

Příklad

Jak bychom vypsali čísla 10, 9, 8, . . . 1 ?

Cykly – opakování příkazů

- **While cyklus** – opakuje, dokud platí podmínka:

```
i = 0
while i < 5:
    print("i je:", i)
    i = i + 1
```

Příklad

Napište kód který vypíše všechna sudá čísla menší než 20, pomocí cyklu `while`.

Cykly

Příklad

Vytvořte proměnnou `n`, které přiřaďte nějaké kladné číslo. Spočítejte průměr hodnot od 1 do `n`.

Funkce – opakovaně použitelné bloky kódu

- Funkce umožňuje zabalit logiku do jednoho pojmenovaného bloku
- Můžeme ji opakovaně volat s různými vstupy
- Syntaxe:

```
def jmeno_funkce(vstup1, vstup2):
    # telo funkce
    return vysledek
```

- Příklad – výpočet BMI:

```
def bmi(vaha, vyska):
    return vaha / (vyska ** 2)

print(bmi(70, 1.75))
```

Funkce

Příklad

Vytvořte funkci, která bude brát jako vstup hodnotu `n` a bude počítat průměr hodnot od 1 do `n`.

Příklad

Vytvořte funkci, která bude brát jako vstup hodnotu `n` a bude počítat faktoriál čísla `n`.

Příklad

Jaké můžeme zvolit přístupy k řešení předchozího úkolu?

Seznamy – práce s více hodnotami

- Seznam je struktura, která uchovává více hodnot v jednom objektu
- Hodnoty mohou být čísla, texty, nebo jiné objekty
- Vytvoření seznamu:

```
cisla = [3, 7, 2, 9]
jmena = ["Anna", "Petr", "Eva"]
```

- Přístup k prvkům pomocí indexu (od nuly):

```
print(cisla[0])   # Vystup: 3
print(jmena[2])   # Vystup: Eva
```

- Iterace přes seznam:

```
for jmeno in jmena:
    print(jmeno)
```

Seznamy

Příklad

Vytvořte funkci, která bude brát jako vstup seznam hodnot `seznam` a bude počítat průměr hodnot z tohoto seznamu.

Další úkoly k procvičení

Příklad

Napište funkci `pozdrav(jmeno)`, která vypíše "Ahoj, jmeno!".

Příklad

Vytvořte seznam čísel a vypište jeho délku pomocí funkce `len()`. (Zjistěte sami, jak se tato funkce používá)

Příklad

Vytvořte funkci, která pro zadané číslo rozhodne, zda je sudé, liché nebo 0.

Příklad

Napište funkci `je_prvocislo(n)`, která vrátí `True`, pokud je číslo prvočíslem, jinak `False`.