

## Skriptovací tutorial

V tomto tutoriálu si ukážeme jak zautomatizovat nějaký opakující se proces, v tomto případě tvorbu jednotlivých obrázků animace. Pro tento účel použijeme programovací jazyk Python. Tento programovací jazyk je možné použít buď tak, že napíšeme v příkazovém řádku Linuxu příkaz `python`. Tím se nám zobrazí příkazová řádka Pythonu, do které můžeme zadávat jednotlivé příkazy. Druhou možností je napsat program do souboru s názvem například `program.py` a ten pak spustit pomocí příkazu `python program.py`. Třetí možností si ukážeme nakonec.

Soubor `program.py` si můžeme vytvořit pomocí příkazů:

```
cat > program.py
print 1+1
Ctrl+D
```

Tím získáme soubor `program.py` s jediným řádkem `print 1+1`. Ten spustíme pomocí příkazu `python program.py`. Program sečte `1+1` a výsledek vytiskne. Program pak můžeme editovat, tento řádek smazat a místo něj napsat:

```
for i in range(10):
    print i
```

Je důležité, aby na druhém řádku jsou (nejlépe dvě) mezery před příkaze `print`, čímž je vymezen cyklus. Příkaz `range(10)` vrátí 10 čísel od 0 do 9. Řádek `for i in range(10):` provede desetkrát za sebou vše, co za ním následuje a je odsazeno určitým počtem mezer. V prvním cyklu bude hodnota `i` rovna nule, ve druhém bude 1 a tak dále až 9. Zkuste to pustit. Když nahradíte příkaz `print i` za `print i+i`, pak program vytiskne hodnoty 0, 2 až 18, tedy sečte nuly, jedničky, dvojky a další jako čísla. Pokud bychom je chtěli sečíst jako řetězce, pak bychom je museli převést na řetězce funkcí `str`, tedy nahradit `print i+i` za `print str(i)+str(i)`. Program sečte dvě nuly ve formě řetězců jako `00`, jedničky jako `11` a tak dále. Podobně můžeme sčítat "0" a `i`, čímž získáme `00`, `01` až `09`.

My využijeme sčítání řetězců k tomu, abychom získali v každém cyklu `000`, `001` až `009` a v případě více cyklů pak `010` až `999`, tedy vždy tři cifry. Proto si převedeme `i` na řetězec a uložíme si ho v proměnné `si` (příkazem `si = str(i)`). Dále k němu budeme přidávat na začátek nuly dokud bude platit, že je délka řetězce menší než tři. Celé to bude vypadat takto:

```
for i in range(10):
    si = str(i)
    while len(si)<3:
        si = "0"+si
    print si
```

Všimněte si, že před `si = "0"+si` jsou čtyři mezery, je to tedy cyklus v cyklu `for`.

Nyní uděláme odbočku k programu Pymol (<https://www.pymol.org/>), který slouží k vizualizaci struktur molekul. Výsledné obrázky je možné uložit ve formátu `png` a je možné uložit i `video`. My si ale ukážeme jiný způsob jak udělat krásné `video`. Program Pymol umožňuje vytvořit vstupní soubor pro všeobecný vizualizační program `PoVRay`. Tento soubor obsahuje scénu jako soubor kuliček, trubek a jiných geometrických útvarů spolu s instrukcemi o barvách, texturách, mlze, poloze kamery a světla a podobně. Stáhněte si z `PDB` na svou pracovní stanici strukturu s kódem `1EJG`. Otevřete `PDB` soubor na své pracovní stanici v Pymolu, změňte zobrazení na `spheres` a výsledek uložte jako soubor pro `PoVRay`. Soubor můžete přkopírovat na výpočetní stanici a tam renderovat pomocí příkazu:

```
povray +H400 +W600 +A0.3 test.pov
```

(případně změňte jméno souboru). Měli byste získat soubor `test.png`, který si můžete přetáhnout na pracovní stanici a tam se na něj kouknout. Volby značí výšku, šířku a `antialiasing` obrázku v jednotkách `px`. Dále můžete `pov` soubor editovat a zkusit si s něčím pohrát.

My budeme chtít se strukturou točit, proto musíme udělat následující operace. Nejprve za řádky:

```
light_source{<4000.0001,4000.0001,9930.9499>  rgb<1.0,1.0,1.0>}
plane{z , -106.1134
  pigment{color rgb<0.0000,0.0000,0.0000>}
  finish{phong 0 specular 0 diffuse 0 ambient 1.0}}
```

přidáme řádek:

```
#declare vse = union {
```

a na konec celého souboru přidáme na nový řádek složenou závorku:

```
}
```

Tím definujeme sjednocení nazvané vse (jako vše). Pokud soubor uložíte a zopakujete příkaz pro renderování, pak výsledkem bude černé plátno, protože jsme si definovali vse, ale nepoužil jsme jej. To můžeme napravit tak, že na konec souboru na zvláštní řádek dáme

```
object {vse}
```

Kdybychom chtěli tím vším rotovat, pak to můžeme udělat pomocí funkce rotate, za kterou v závorce tvořené znaménky < a > napíšeme o kolik stupňů chceme rotovat. Například rotate <0,10,0> otočí objekt o 10 stupňů kolem osy y. Když se ale podíváte na souřadnice objektů v souboru pov, pak zjistíte, že se kamera nachází v bodě 0,0,0 a protein je posunut někam do dálky podél osy z. Přibližně se z souřadnice nachází kolem -90. Kdybychom s proteinem rotovali kolem osy y, pak by nám při animaci čas od času přeletěl přes plátno. Proto nejprve posuneme protein o 90 jednotek podél osy z, otočíme jej a vrátíme zpět, tedy:

```
object {
vse
translate <0,0,90>
rotate <0,10,0>
translate <0,0,-90>
}
```

Vyzkoušejte s různými úhly.

Pro krásnou animaci nám nyní stačí vše zautomatizovat. Udělejte kopii pov souboru, například:

```
cp test.pov test_zacatek.pov
```

a smažte poslední řádky tak, aby byl řádek translate <0,0,90> jako poslední. Pak znovu upravíme skript:

```
for i in range(10):
  si = str(i)
  while len(si)<3:
    si = "0"+si
  zacatek = open("test_zacatek.pov", "r").readlines()
  novy = open("final.pov", "w")
  for line in zacatek:
    novy.write(line)
  novy.write("rotate <0,"+str(i)+"> \n")
  novy.write("translate <0,0,-90> \n")
  novy.write("}\n\n")
  novy.close()
```

Program otevře soubor funkcí open, otevře jej pro čtení (proto "r") a rozkuchá je ne jednotlivé řádky (proto readlines()). Na dalším řádku si program vytvoří soubor final.pov pro psaní (proto "w"). Smyčkou for line in zacatek: a novy.write(line) program zkopíruje soubor zacatek do souboru novy. Pak na něj přilepí řádek, který bude rotovat s atomy o i stupňů, řádek s příkazem translate a řádek s uzavřením hranatých závorek. Nakonec program soubor zavře. Když tento program pustíte, pak desetkrát vytvoří soubor final.pov, ale protože je v každém cyklu soubor přepsán, tak získáme pouze poslední soubor. Abychom získali vizualizaci, pak musíme v každém cyklu přidat příkaz pro renderování. Aby to bylo možné, je nutné použít knihovnu os. Proto přidejte na začátek skriptu řádek:

```
import os
```

Pak za řádek `novy.close()` přidejte:

```
os.system("povray +H400 +W600 +A0.3 -Of"+si+".png final.pov")
```

Pak by měl program vytvořit v prvním kroku obrázek *f000.png*, ve druhém *f001.png* a tak dále.

Můžete zkusit to samé s více cykly. Pro výslednou animaci můžete použít program `mencoder` z balíku `Mplayer`. Příkaz by mohl vypadat například takto:

```
mencoder -ovc lavc -lavcopts vcodec=mpeg4:vpas=1:vbitrate=1620000:  
mbd=2: keyint=132:v4mv:vqmin=3:vlelim=-4:vcelim=7:lumi_mask=0.07:  
dark_mask=0.10: naq:vqcomp=0.7:vqblur=0.2:mpeg_quant  
-mf type=png:fps=25 -nosound -o test.avi mf://f*.png
```