

Simulacije

- Pogosto računalnik uporabljamo za simulacijo določenih procesov.
 - Radi bi zbrali vse slikice Smrkcev. Koliko evrov moramo zapraviti v Mercatorju, da bomo dobili vse (je 24 različnih slikic, za vsakih 10€ dobimo eno).
 - Prikaži dogajanje na sistemu k blagajn, če veš, da se v povprečju vsakih n minut pojavi kupec in da posameznega kupca blagajničarka odpravi v m minutah. Vsak kupec se vedno postavi v najbolj prazno vrsto. Vsakih t minut pa pride kupec, katerega „obravnavava“ zahteva p minut.
 - Napiši program, ki bo nadomestil delilca kart pri pokru.
 - Kako igrati človek ne jezi se, če smo ostali brez kocke.

Simulacija meta kocke

- Napišimo funkcijo, ki bo simulirala kocko
 - Funkcija vrne neko naključno število med 1 in 6

```
def kocka():
```

```
    return ? # naključno število med 1 in 6
```

- In kaj napisati namesto '?' ?

Naključna števila

- `import random`
- `random.random()`
- Dobimo decimalno število tipa `double` z intervala `[0.0, 1.0)`
- `6 * (random.random())` dobimo dec. število z intervala `[0,6)`
- `int(6 * (random.random()))` – dobimo število iz množice `{0, 1, 2, 3, 4, 5}`
- Torej

```
return 1 + int(6 * (random.random()))
```

Naključna števila

- In potem, ko dobro premislimo, kako dobiti naključno število med a in b

```
int((b - a + 1)*random.random()) + a
```

- Nam iskanje po spletu
 - pokaže preprostejšo (manj tipkanja) rešitev
 - `random.randint(a, b)`

random.randint(a, b)

- Dobimo naključno celo število med **a** in **b**. Meji sta **vključeni**, torej lahko dobimo tako **a** kot tudi **b**.

- Kocka:

- `random.randint(1, 6)`



- Ruleta (francoska):

- `random.randint(0, 36)`



- Poker:

- Barva:

- `random.randint(1, 4)`

- Vrednost:

- `random.randint(2, 14)`



random.shuffle() in random.choice()

- `random.shuffle(sez)` naključno premeša elemente v seznamu `sez`
 - `>>> sez = [1, 2, 3, 4, 5]`
 - `>>> random.shuffle(sez)`
 - `>>> sez`
 - `[3, 4, 5, 1, 2]`
 - `>>>`
- `random.choice(sez)` vrne naključni element iz seznama `sez`
 - Uporabimo lahko tudi niz in dobimo naključni znak
 - `>>> random.choice(sez)`
 - `2`
 - `>>> random.choice('matija')`
 - `'j'`
 - `>>> random.choice(['matija'])`
 - `'matija'`
 - `>>>`

Kako pa dobiti anagram

- Kaj je [anagram](#)?
 - 'Eleven plus two' => 'Twelve plus one'
 - 'sliva' => 'Silva' => 'Slavi' => 'svila'
 - 'iavls', 'avasl', ...
- `random.shuffle('matija')`
 - Traceback (most recent call last):
 - File "<pyshell#1>", line 1, in <module> random.shuffle('matija')
 - File "C:\Python32\lib\random.py", line 266, in shuffle
 - `x[i], x[j] = x[j], x[i]`
 - TypeError: 'str' object does not support item assignment
- `>>> sez = list('matija')`
- `>>> random.shuffle(sez)`
- `>>> ''.join(sez)`
- `'iajmta'`