

Geração de Números Pseudoaleatórios

CIÊNCIA DA
COMPUTAÇÃO

ATITUS
EDUCAÇÃO



ATITUS
EDUCAÇÃO



- Aula 07 -
Pensamento Computacional

Prof. Me. Lucas R. C. Pessutto

Na aula de hoje...

01

Números Pseudoaleatórios

O que são?

Biblioteca random

Números Pseudoaleatórios

- * São muito utilizados em programação para gerar comportamento aleatório em programas
- * Exemplos de programas que usam esses números são jogos ou simuladores
- * Existem algoritmos para geração de números que são aparentemente aleatórios
- * Na verdade são gerados a partir de funções matemáticas



Números Pseudoaleatórios em Python

- * O módulo random inclui várias funções úteis para a geração de números pseudoaleatórios
- * Por exemplo, para gerar um número no intervalo $[a, b]$ podemos utilizar a função randint:

```
rnd = random.randint(a, b)
```

- * A variável rnd recebe um valor inteiro dentro do intervalo delimitado pelos valores inteiros a e b, inclusive, que são passados para a função

Números Pseudoaleatórios em Python

* Exemplos:

```
import random
```

```
print("Os números sorteados são:")
```

```
rnd = random.randint(1, 10)
```

```
print(rnd)
```

```
rnd = random.randint(0, 100)
```

```
print(rnd)
```

```
rnd = random.randint(-100, 100)
```

```
print(rnd)
```

```
Os números sorteados são:
```

```
5
```

```
13
```

```
18
```

```
Os números sorteados são:
```

```
8
```

```
48
```

```
23
```

```
Os números sorteados são:
```

```
1
```

```
84
```

```
-12
```

Números Pseudoaleatórios em Python

- * Às vezes, queremos repetir execuções com os mesmos números pseudoaleatórios, e para isso podemos utilizar

```
random.seed(a)
```

- * A variável a é o que chamamos de semente
- * Para uma mesma semente, têm-se um mesmo conjunto de números pseudoaleatórios sorteados em execuções diferentes
- * O valor padrão de a é None, o que faz com que não haja controle na seleção dos números pseudoaleatórios

Números Pseudoaleatórios em Python

* Exemplos:

```
import random
```

```
print("Os números sorteados são:")
```

```
# Configurando a semente aleatória  
random.seed(100)
```

```
rnd = random.randint(1, 10)  
print(rnd)
```

```
rnd = random.randint(0, 100)  
print(rnd)
```

```
rnd = random.randint(-100, 100)  
print(rnd)
```

```
Os números sorteados são:  
3  
58  
16
```

```
Os números sorteados são:  
3  
58  
16
```

```
Os números sorteados são:  
3  
58  
16
```

Números Pseudoaleatórios em Python

- * Há uma série de outras funções úteis, como:

`random.uniform`

`random.choice`

`random.sample`

`random.shuffle`

Números Pseudoaleatórios em Python

```
import random

print("Função uniform")

rnd = random.uniform(1, 10)
print(rnd)

rnd = random.uniform(-10, 10)
print(rnd)
```

```
Função uniform
4.989056580299216
-3.356665064167526
```

Números Pseudoaleatórios em Python

```
cores = ['amarelo', 'azul', 'vermelho', 'preto', 'verde', 'branco']
```

```
print("Função choice")
```

```
cor = random.choice(cores)  
print(f"Cor escolhida: {cor}")
```

```
cor = random.choice(cores)  
print(f"Cor escolhida: {cor}")
```

```
Função choice  
Cor escolhida: azul  
Cor escolhida: vermelho
```

Números Pseudoaleatórios em Python

```
print("Função sample")
```

```
print(f"Amostra 1 = {random.sample(cores, 3)}")  
print(f"Amostra 2 = {random.sample(cores, 2)}")  
print(f"Amostra 3 = {random.sample(cores, 5)}")
```

Função sample

Amostra 1 = ['preto', 'amarelo', 'branco']

Amostra 2 = ['vermelho', 'amarelo']

Amostra 3 = ['branco', 'verde', 'amarelo', 'preto', 'azul']

Números Pseudoaleatórios em Python

```
print("Função shuffle")

random.shuffle(cores)
print(f"Embaralhando 1: {cores}")

random.shuffle(cores)
print(f"Embaralhando 2: {cores}")

random.shuffle(cores)
print(f"Embaralhando 3: {cores}")
```

Função shuffle

```
Embaralhando 1: ['preto', 'vermelho', 'amarelo', 'branco', 'verde', 'azul']
Embaralhando 2: ['preto', 'azul', 'vermelho', 'verde', 'branco', 'amarelo']
Embaralhando 3: ['verde', 'amarelo', 'branco', 'preto', 'vermelho', 'azul']
```

Problema 03: Jogo do par ou ímpar



Enunciado de um problema:

- * Implemente um jogo de par ou ímpar
- * O programa solicita ao usuário que digite um valor inteiro positivo e um caractere 'p' para a opção par e 'i' para opção ímpar
- * O computador deve fazer a geração de um número inteiro entre 1 e 10 para sua jogada e calcular o resultado

Problema 03: Jogo do par ou ímpar



Análise do Problema

Entradas: caractere 'p' ou 'i' e um número inteiro

Saída: resultado (par ou ímpar) e vencedor

Processamento: geração de um valor inteiro e verificação se a soma das jogadas é par ou ímpar

```
[P]ar ou [I]mpar? p  
Escolha um número: 6
```

```
O computador escolheu 3.  
A soma é ímpar! Eu ganhei :)
```

```
[P]ar ou [I]mpar? i  
Escolha um número: 1
```

```
O computador escolheu 4.  
A soma é ímpar! Você ganhou :(
```

Problema 03: Jogo do par ou ímpar



Implementação do Programa

```
import random

# Lê a opção
opcao = ''
while opcao != 'p' and opcao != 'i':
    opcao = input("[P]ar ou [I]mpar? ")
    opcao = opcao.lower()
    if opcao != 'p' and opcao != 'i':
        print("Opção inválida! ", end="")

# Lê a jogada e escolhe o número aleatório
humano = int(input("Escolha um número: "))
computador = random.randint(0, 10)
print(f"O computador escolheu: {computador}")
```

Problema 03: Jogo do par ou ímpar



Implementação do
Programa

```
total = humano + computador
# Verifica se a soma é par ou ímpar
if total % 2 == 0:
    print("A soma é par! ", end="")
    resultado = 'p'
else:
    print("A soma é ímpar! ", end="")
    resultado = 'i'

if resultado == opcao:
    print("Você ganhou :(")
else:
    print("Eu ganhei! :)")
```