

Objetivos

- Trabalhar com o for, while, if em Python¹
- Desenvolver bom estilo de programação

Ficheiros necessários

- LAB4.zip

Exercícios

Quanto é isso em anos de cão?

Todos sabemos que os nossos melhores amigos envelhecem a um ritmo diferente dos humanos. Escreva um programa que pede ao utilizador uma idade humana (expressa num número inteiro) e exibe a idade equivalente de um cão, usando a regra simples que um ano humano corresponde a sete anos dos cães. Considere definir a constante `ANOS_CAO_POR_ANO_HUMANO`. Pode assumir que o utilizador introduz uma idade inteira, mas que o inteiro não é necessariamente positivo. Se não for, exiba uma mensagem de erro.

O seu programa deve pedir continuamente idades humanas ao utilizador até que o mesmo introduza 0, altura em que o programa deve terminar.

Aqui fica um exemplo de uma possível execução do seu programa (input do utilizador em *italico*):

```
Introduza uma idade em anos humanos: -12
Lamentamos, por favor introduza um número positivo ou 0 para sair
Introduza uma idade em anos humanos: 13
A idade equivalente nos cães é 91
Introduza uma idade em anos humanos: 0
```

Encontrar os divisores

Escreva um programa que pede ao utilizador para introduzir um inteiro, e depois exibe todos os divisores do número introduzido, um a um. A sua função deve verificar que o número introduzido é superior a 0. O programa deve pedir números continuamente até que o número 0 seja introduzido.

Aqui fica um exemplo de uma possível execução do seu programa (input do utilizador em *italico*):

```
0 seu número: -10
Por favor introduza um número positivo
0 seu número: 42
1
2
3
6
7
```

¹ exercícios adaptados de: Mehran Sahami, Chris Piech, Brahm Capoor, Juliette Woodrow, Parth Sarin, Kara Eng & Tori Qiu, Programming Methodologies. Stanford University.

```
14
21
42
0 seu número: 53
1
53
0 seu número: 0
```

Pedra, Papel, Tesoura

Em 1997, um computador chamado Deep Blue venceu o campeão mundial de xadrez Gary Kasparov num jogo de xadrez. Em 2023, a IBM finalmente ganhou confiança para expandir seu repertório de jogos de humanos vs. computador e contratou-o para escrever um programa em que um jogador humano e o computador se defrontam num jogo de pedra, papel e tesoura.

Cada jogo consiste em 5 turnos de Pedra, Papel e Tesoura. Em cada turno o utilizador escolhe se quer jogar Pedra, Papel ou Tesoura, e o computador faz o mesmo. Neste jogo, o utilizador digitará 1 se quiser jogar Pedra, 2 se quiser jogar Papel e 3 se quiser jogar Tesoura.

Em cada turno, siga os seguintes passos:

1. Peça ao utilizador para introduzir a sua jogada (pode assumir que ele digita ‘1’, ‘2,’ ou ‘3’)
2. Escolha aleatoriamente pedra, papel ou tesoura como o movimento do computador usando o módulo random (lembre-se, chamar ‘random.randint(a, b)’ devolve-lhe um número aleatório entre a e b , inclusive)
3. Determine quem vence o turno. Lembre-se que a pedra vence a tesoura, a tesoura vence o papel, e o papel (de alguma forma) vence a pedra.
4. Exiba uma mensagem a relatar se o jogador humano ganhou ou perdeu, e qual foi a jogada de cada jogador.

No fim do programa, mostre uma mensagem a dizer quantos turnos é que o jogador ganhou.

Aqui fica um exemplo de uma possível execução do seu programa (input do utilizador em *íntico*):

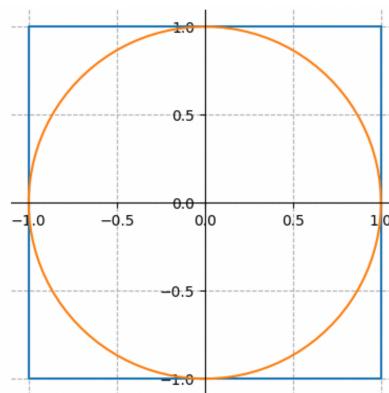
```
A sua jogada (1, 2, ou 3): 1
Empate!
A sua jogada (1, 2, ou 3): 3
Ganhou! A tesoura corta o papel
A sua jogada (1, 2, ou 3): 2
Perdeu! A tesoura corta o papel
A sua jogada (1, 2, ou 3): 2
Empate!
A sua jogada (1, 2, ou 3): 1
Ganhou! A pedra esmaga a tesoura
Ganhou 2 jogadas!
```

Estimando π

Escreva uma função, `estimar_pi`, que estima o valor de π usando o Python!

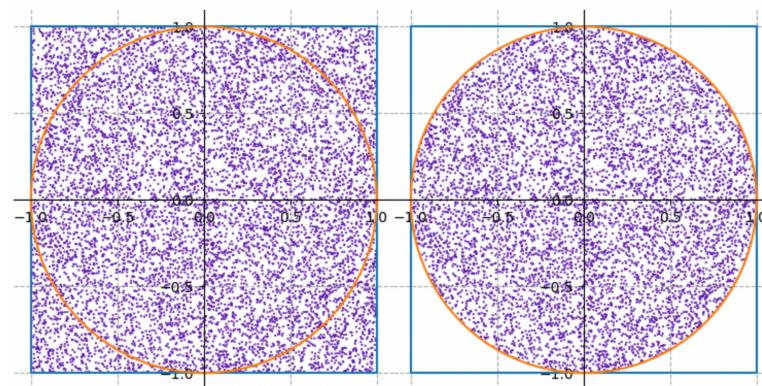
π é uma constante matemática associada às propriedades geométricas de um círculo. Especificamente, um círculo com raio r terá uma área πr^2 e um perímetro $2\pi r$. O valor de π é cerca de 3,14159... e neste problema vamo-nos aproximar bastante desse valor (vamos tentar uma precisão de 2 casas decimais)!

Uma forma interessante de estimar o valor de π é com o módulo `random`. Imagine um quadrado 2×2 desenhado num plano 2D, com centro na origem. Podemos desenhar um círculo dentro deste quadrado, com raio 1, centrado na origem. O nosso desenho será algo como isto:



A área do quadrado será $2 \times 2 = 4$ e a área do círculo será $\pi \times 1^2 = \pi$. O círculo ocupa uma grande parte da área do quadrado: para ser exato, o rácio da área do círculo para a área do quadrado é $\frac{\pi}{4}$.

Uma forma de pensar nisto é que se escolher um monte de pontos ao acaso dentro do quadrado, a fração de pontos que vão estar dentro do círculo é $\frac{\pi}{4}$.



A sua função, `estimar_pi`, deverá escolher aleatoriamente 100000 pontos dentro do quadrado (lembre-se que é bom estilo de programação usar constantes onde apropriado; considere definir uma constante `NUM_PONTOS = 100000`).

Pode escolher um ponto aleatório no quadrado escolhendo uma coordenada aleatória x e uma coordenada aleatória y (ambos os quais devem ser números reais entre -1 e 1). Conte o número de pontos que estão dentro do círculo e use esse número para calcular a fração de pontos que caíram dentro do círculo. Esse número é aproximadamente $\frac{\pi}{4}$, por isso multiplique o seu rácio por 4 e exiba esse valor como a nossa aproximação de π . Quão perto chegou? O que acontece quando altera `NUM_PONTOS`?

Dica: Um ponto (x, y) está dentro do círculo de raio 1 centrado na origem, se $x^2 + y^2 \leq 1$.