

O jogo "Pedra Papel Tesoura"

Em português, "Pedra Papel Tesoura" é um jogo simples que duas ou mais pessoas podem jogar para decidir, por exemplo, quem começa um jogo. Simultaneamente cada jogador lança Pedra (mão fechada), Papel (mão aberta) ou Tesoura (indicador e dedo do meio em V). As regras são simples:

- Pedra vence contra Tesoura;
- Tesoura vence contra Papel;
- Papel vence contra Pedra.

1. Vamos lá começar...

Tal como o jogo da moeda (Tutorial [T5A](#)), a primeira parte do programa consiste em seleccionar aleatoriamente a jogada do computador e ler (input) a jogada do jogador.

O algoritmo deste jogo é o seguinte:

1. Seleccionar aleatoriamente a jogada do computador.
2. Ler a jogada do jogador (input)
3. Mostrar a jogada do computador (output).
4. Verificar se o jogador ganha, perde ou empata e visualizar esse resultado (output).

Assim, o jogador deverá pensar numa opção (pedra, papel ou tesoura) e depois executar o programa, verificando se acertou ou não.

Em Python, o código que implementa os 3 primeiros passos do algoritmo é o seguinte:

```
import random

# 1. escolhe objeto 'a sorte
escolha = random.choice(["pedra", "papel", "tesoura"])

# 2. le jogada do jogador (input)
jogada = input("pedra, papel ou tesoura? ")

# 3.
print ("Eu joguei", escolha)
```

Através da função [random.choice\(\)](#), cria-se aleatoriamente o valor de "pedra", "papel" ou "tesoura" que é atribuído à variável `escolha`.

Em seguida lê-se a jogada do jogador para a variável `jogada` e visualiza-se o resultado da escolha do computador com a função [print](#).



Agora experimente esta primeira versão...

2. Quem vence o jogo?

Este jogo tem várias possibilidades, podendo obter-se a derrota, a vitória ou o empate. Isto depende das regras do jogo, também denominadas "mecânicas de jogo".

Quantas possibilidades haverá?

Com três símbolos temos $3^2 = 9$ possibilidades (computador vs jogador):

- "pedra" vs "tesoura": Perdeu!
- "tesoura" vs "papel": "Perdeu!"
- "papel" vs "pedra": "Perdeu!"
- "tesoura" vs "pedra": "Ganhou!"
- "papel" vs "tesoura": "Ganhou!"
- "pedra" vs "papel": "Ganhou!"

As 3 restantes possibilidades dão o "Empate..."

Teremos assim que fazer uma decisão entre múltiplas possibilidades. Precisamos utilizar uma estrutura de decisão para fazer uma seleção múltipla. Em Python esta estrutura é o [if...elif...else](#).

O código que implementa as regras do jogo definidas acima é o seguinte:

```
# 3. verifica se ganhou ou perdeu
if escolha == "pedra" and jogada == "tesoura":
    print ("perdeu! ")
elif escolha == "tesoura" and jogada == "papel":
    print ("perdeu! ")
elif escolha == "papel" and jogada == "pedra":
    print ("perdeu! ")
elif escolha == "tesoura" and jogada == "pedra":
    print ("Ganhou! ")
elif escolha == "papel" and jogada == "tesoura":
    print ("Ganhou! ")
elif escolha == "pedra" and jogada == "papel":
    print ("Ganhou! ")
else:
    print ("Empate!")
```

Tanto o primeiro "if" como os sucessivos "elif" têm expressões condicionais para avaliar cada uma das jogadas. A primeira expressão condicional a ser avaliada como verdadeira é executada e o resto das expressões já não são avaliadas. Se nenhuma destas expressões condicionais for verdadeira então aplica-se o código presente no "else", caso tenha sido definido.

Repare que para avaliar cada uma das jogadas é necessário assegurar que ambos os jogadores apresentam um par de valores específico, simultaneamente. Para tal usa-se uma operação de conjunção, definida pelo operador lógico [and](#). Assim, só quando ambas as expressões são verdadeiras é que a expressão condicional é verdadeira. Nos restantes casos é falsa.

3. Desafio final

As mecânicas de jogo podem ser reutilizadas em diferentes jogos, com temas e/ou conceitos diferentes.

Desenvolva uma variação deste jogo, selecionando outro tema de forma a substituir as palavras "pedra", "papel" e "tesoura" por outras três que funcionem com as mesmas regras.



Partilhe o seu jogo!