

Funções (continuação). Abstração e reutilização.**Exercícios**

1. No programa `intersection.py`, escreva uma função `intersection(a, b, c, d)` que determine a intersecção dos intervalos $[a, b]$ e $[c, d]$. A função deve devolver um par de números (e, f) que representam o intervalo resultado $[e, f]$. Se o resultado for um intervalo vazio, deve devolver $(0, 0)$. Note que a função só funciona se $a \leq b \wedge c \leq d$. Essas pré-condições são verificadas com instruções `assert`. Execute o programa para testar a função. Acrescente outros casos de teste à função principal (pode partilhar casos de teste com outros colegas).
2. Corrija e complete cada uma das funções pedidas [nesta tarefa CodeCheck](#). Repare que é uma extensão do programa `dates.py` do exercício 9 da aula03. Procure decompor as funções mais complexas em tarefas mais simples que possam ser expressas por chamadas a funções mais simples.
3. O programa `houses1.py` usa funções do módulo `turtle` para desenhar retângulos numa janela gráfica sempre que o utilizador clica no botão esquerdo do rato. Experimente e analise o programa.
 - a. Complete a função `house` para desenhar uma casa com telhado, porta e janela. Pode e deve reutilizar as funções `rect` e `triang`, já fornecidas.
 - b. Modifique a função `main` para desenhar uma casa sempre que o utilizador clicar no botão direito do rato.

Este é um exemplo de [programação orientada a eventos](#). O programa aguarda continuamente por eventos e quando algum ocorre (como um clique no rato), então chama uma função previamente registada para lidar com esse evento. Chama-se *ciclo de eventos* (*event loop*) ao processo de esperar e lidar com eventos repetitivamente, e chama-se *callback functions* às funções que são registadas para lidar com os eventos.

4. Crie um programa para calcular o preço total que uma família de 4 pessoas paga ao comprar bilhetes num museu. A tabela de preços é esta:

Idade	Preço
Inferior a 6 anos	Isento
Entre 6 e 12 anos	2.50 €
Entre 13 e 65 anos	5.00 €
Mais do que 65 anos	2.50 €

Crie duas versões do programa. Numa faça um programa monolítico, sem definir funções. Na outra defina uma ou mais funções que permitam decompor o problema em partes mais simples e reutilizáveis. Não pode recorrer a instruções de repetição (`for` ou `while`).