Generalitat de Catalunya
Departament d'Educació
Institut Thos i Codina

Com programar: de l'enunciat al codi

Domini Públic (CC0)

Índex

Enunciat del problema **1**

Resolució **2**
Fase 1: Entendre què demana 2
Pas 1. Llegeix l'enunciat i marca tres coses 2
Pas 2. Posa nom a la funció i als paràmetres 2
Pas 3. Mira l'exemple i comprova que l'entens 2
Fase 2: Pensa la solució (sense escriure codi encara) 3
Pas 4. Resol (de cap o amb paper) una frase de l'enunciat 3
Pas 5. Escriu els passos en català normal (pseudocodi) 3
Fase 3: Escriu el codi 3
Pas 6. Escriu l'esquelet (la "caixa" buida) 3
Pas 7. Tradueix cada línia del pseudocodi a Python, una a una 3
Fase 4: Comprova que funciona 4
Pas 8. Executa amb l'exemple de l'enunciat 4
Pas 9. Compara el que surt amb el que diu l'enunciat 4
Pas 10. Prova un cas diferent 4
Fase 5. Revisa que no has comès cap dels errors habituals 4
Versions d'aquest document 4

Programar és pensar, no escriure codi

Programar no consisteix a escriure codi, sinó a pensar un procediment que algú altre, l'ordinador, pugui seguir sense entendre res. El codi és només la manera d'escriure aquest pensament en un idioma que l'ordinador accepta.

Descompondre

Aprendre a programar és, sobretot, aprendre a descompondre: un problema gran que no sé resoldre l'he de dividir en trossos petits que sí que sé fer. Si no soc capaç d'explicar amb les meves paraules com es resol, tampoc no li ho podré explicar a l'ordinador.

El llenguatge és secundari

Qui sap pensar bé el procediment el pot escriure en qualsevol llenguatge. Python és només un llenguatge més: la sintaxi canvia d'un a altre, però la lògica és la mateixa.

Un mapa del tresor

Imaginem un mapa del tresor que diu: «Camina cap al nord, gira a l'esquerra, camina fins a la roca i excava». El mapa no és el tresor, sinó una seqüència d'indicacions ordenades que algú ha de poder seguir sense conèixer el terreny. Però si diu «camina cap al nord» sense dir quants passos ni la mida d'aquests, qui el segueix no sap quan parar: pot aturar-se massa aviat o passar-se de llarg. Passa el mateix amb «gira a l'esquerra» sense dir quants graus, o amb «fins a la roca» si hi ha tres roques. Programar és dibuixar mapes tan precisos que els pugui seguir algú que no coneix el terreny ni sap res del que busquem.

Enunciat del problema

```
Enunciat:

Fes una funció anomenada esborra que, a partir d'una llista de frases
 ↪  i un caràcter,
esborri el caràcter indicat de totes les frases. Retorna la llista
 ↪  modificada i la
quantitat de caràcters esborrats.

Exemple:

Llista de frases: ["El talento, gana partidos, pero el trabajo en
 ↪  equipo y la
inteligencia, gana campeonatos.", "A veces, cuando, se, pierde, se,
 ↪  gana.",
"No se puede ganar, una guerra como, tampoco se, puede, ganar un
 ↪  terremoto.",
"La pérdida, de, unos, es la, ganancia, de otros."]
Caràcter: ','

Resultat:

["El talento gana partidos pero el trabajo en equipo y la inteligencia
 ↪  gana
```

```
campeonatos.", "A veces cuando se pierde se gana.", "No se puede ganar
 ↪  una guerra
como tampoco se puede ganar un terremoto.", "La pérdida de unos es la
 ↪  ganancia de
otros."]
17
```

Resolució

Aquests passos serveixen per a qualsevol exercici de funcions. Es fan sempre en aquest ordre i no te n'has de saltar cap.

Fase 1: Entendre què demana

Pas 1. Llegeix l'enunciat i marca tres coses

Pregunta	Resposta en aquest exercici
Què **rep** la funció? (entrades)	Una llista de frases i un caràcter
Què **ha de fer**?	Esborrar el caràcter de totes les frases
Què **retorna**? (sortides)	La llista modificada i un número (quants caràcters s'han esborrat)

Pas 2. Posa nom a la funció i als paràmetres

- L'enunciat diu que la funció es diu `esborra`.
- Les entrades seran `frases` i `caracter`.

Pas 3. Mira l'exemple i comprova que l'entens

- Entrada: `["A veces, cuando, se, pierde, se, gana."]` i `','`
- Sortida: `["A veces cuando se pierde se gana."]` i `5`
- Comprovació: a la frase hi ha 5 comes. Ho he entès.

Fase 2: Pensa la solució (sense escriure codi encara)

Pas 4. Resol (de cap o amb paper) una frase de l'enunciat

Agafo "`A veces, cuando, se`" i vaig lletra per lletra:

- Si la lletra és la coma → no la guardo i sumo 1 al comptador.
- Si no és la coma → la guardo.

Pas 5. Escriu els passos en català normal (pseudocodi)

```
Començar amb comptador = 0 i una llista nova buida
Per cada frase de la llista:
    Fer una frase nova buida
    Per cada lletra de la frase:
        Si la lletra és el caràcter → comptador + 1
        Si no → afegir la lletra a la frase nova
    Afegir la frase nova a la llista nova
Retornar la llista nova i el comptador
```

Fase 3: Escriu el codi

Pas 6. Escriu l'esquelet (la "caixa" buida)

```
def esborra(frases, caracter):
    ...
    return nova_llista, comptador
```

Pas 7. Tradueix cada línia del pseudocodi a Python, una a una

```
def esborra(frases, caracter):
    comptador = 0
    nova_llista = []
    for frase in frases:
        cadena = ""
        for lletra in frase:
            if lletra == caracter:
                comptador += 1
            else:
                cadena = cadena + lletra
        nova_llista.append(cadena)
    return nova_llista, comptador
```

Fase 4: Comprova que funciona

Pas 8. Executa amb l'exemple de l'enunciat

```
llista = ["El talento, gana partidos, pero el trabajo en equipo y la
 ↪  inteligencia, gana campeonatos.",
          "A veces, cuando, se, pierde, se, gana.",
          "No se puede ganar, una guerra como, tampoco se, puede,
           ↪  ganar un terremoto.",
          "La pérdida, de, unos, es la, ganancia, de otros."]

resultat, quantitat = esborra(llista, ",")
print(resultat)
print(quantitat)
```

Pas 9. Compara el que surt amb el que diu l'enunciat

- La llista és igual a la de l'enunciat? Sí / No
- El número és `17`? Sí / No
- Si alguna resposta és "No", torno al Pas 4 i reviso el raonament.

Pas 10. Prova un cas diferent

- Llista buida `[]` → ha de retornar `([], 0)`.
- Un caràcter que no hi és, per exemple `'z'` → la llista queda igual i el comptador és `0`.

Fase 5. Revisa que no has comès cap dels errors habituals

- ☐ He escrit `return` **dins** de la funció, però **fora** dels bucles?
- ☐ Retorno **dues** coses (llista i número)?
- ☐ He posat el caràcter entre cometes quan crido la funció (`","`)?
- ☐ He fet servir el mateix nom de variable a tot arreu (`caracter`, no `caracters`)?

Versions d'aquest document

- HTML
- PDF
- ODT
- MD

Domini Públic (CC0)