

# 4 Informàtica (Teoría)

## Python: Examen Lab I, Preparación

Joaquim Gabarro  
gabarro@cs.upc.edu

Computer Science  
Universitat Politècnica de Catalunya

Ejemplos de análisis por casos

Ejemplos de recorridos

## Ejemplos de análisis por casos

## Calculadora: `calculadora(op,v1,v2)`

Se pide diseñar una calculadora de una operación.

- ▶ Diseña la función `calculadora(op,v1,v2)` que reciba como parámetros una cadena que corresponde a un operador ('+', '\*', '-', '//', '%', '\*\*') y dos valores enteros `v1` y `v2` y realice el cálculo de los dos valores con el operador escogido.
- ▶ El resultado también ha de ser entero.
- ▶ En caso de error, devolver 0. Es decir si `v2 == 0` el cociente y el resto han de ser 0
- ▶ Hacer también un doctest para verificar la función.

Comentar las soluciones `calculadora_elif.py` y `calculadora_if.py`.

## Plurales: mis\_plurales(w)

- ▶ Las palabras terminadas en **-d** forman el plural con **-des**

ciudad → ciudades

- ▶ Las palabras terminadas en **-z** forman el plural con **-ces**.

disfraz → disfraces

Ejemplos de palabras terminas en **-d** y **-z**: edad, codorniz, feliz, veloz, amistad, verdad, arroz, cicatriz, virtud, capaz, comunidad.

## Ejemplo de aplicación:

```
>>> mis_plurales("edad")
'edades'
>>> mis_plurales("codorniz")
'codornices'
>>> mis_plurales("patata")
'no aplica'
```

Para diseñar el programa aconsejamos mirar el significado de `w[-1]` y `w[:-1]`. Simplemente mirad que pasa cuando `w=patata`.

## Ticket parking: `aparca(minutos)`

Los precios del parking de un Centro Comercial se calculan:

- ▶ Hasta 1 hora (60 minutos) el precio es 50 céntimos
- ▶ Pasados los 60 minutos, el precio es de 50 céntimos más 3 céntimos por minuto que sobrepase los 60 minutos.
- ▶ Para un tiempo superior a 10 h, el precio es de 13 euros.

Diseñar una función `aparca(minutos)` que devuelva el precio en euros y centimos. Utilizar solo variables enteras.

Fixero `aparca(minutos)`

## Ejemplos de recorridos

## Codi seguretat codi(cognom, num)

Una empresa de seguretat genera codis per entrar a les plantes d'un edifici d'oficines a partir del primer cognom de l'empleat i el número de la planta intercalant aquest número entre les lletres del cognom. Els espais en blanc del cognom s'elimen.

Programar la funció `codi(cognom, num)` que, donats un `cognom` i un `número` de planta `num`, retorna el `string` del codi associat a l'empleat.

```
>>> codi("Roig", 7)
'R7o7i7g7'
>>> codi(" de la Roca", 1)
'd1e1l1a1R1o1c1a1'
```

# Comentari

Atenció amb els tipus: no es el mateix un `int` que un `str`:

```
>>> nombre =1
>>> type(nombre)
<class 'int'>
>>> cadena = str(nombre)
>>> type(cadena)
<class 'str'>
```

Fitxer `codi.py`.

## Codi producte `codi_prod(nom, pref)`

Un fabricant associa un codi als seus productes a partir d'un prefix i el nom del producte.

En el nom es substitueix la **e** per **1**, la **i** per **3**. Els espais en blanc del nom es substitueixen per **0**. En el **nom** només apareixen lletres minúscules o espais en blanc.

Cal programar la funció `codi_prod(nom, pref)` que donat el **nom** del producte y una paraula amb el prefixe **pref**, retorni el codi associat al producte.

## Example:

```
>>> codi_prod("diode", "X")
'Xd3od1'
>>> codi_prod("diode zener", "E")
'Ed3od10z1n1r'
>>> codi_prod("diode", " ")
' d3od1'
>>> codi_prod("", "X")
'X'
>>> codi_prod("", "")
'',
```

## Encriptar `encripta(w, c_origen, c_subs)`

Es demana una funció `encripta` tal que entrant-li un string qualsevol, `w` un caràcter a substituir `c_origen` i un altre pel qual serà substituït `c_subs`, aquesta implementi totes les substitucions i retorni:

- ▶ Un string fruit de la concatenació del caràcter `'.'` (punt decimal) i el string fruit de la substitució anterior
- ▶ Un string format per tants asteriscs `'*'` com caràcters formin el string substituït

```
>>> encripta('informatica', 'i', '1')  
      ('.1nformat1ca', '*****')
```

Fitxer `encripta.py`