

## 8.5-Funcions2

1) *Escriviu una funció que calculi el mòdul d'un vector.*

```
def sum_square(x, y):
```

```
    return x + y ** 2
```

```
def module(v):
```

```
    from functools import reduce
```

```
    return reduce(sum_square, v, 0) ** 0.5
```

2) Una immobiliària d'una ciutat maneja una llista d'immobles com la següent:

```
[{'año': 2000, 'metros': 100, 'habitaciones': 3, 'garaje': True, 'zona': 'A'},  
{'año': 2012, 'metros': 60, 'habitaciones': 2, 'garaje': True, 'zona': 'B'},  
{'año': 1980, 'metros': 120, 'habitaciones': 4, 'garaje': False, 'zona': 'A'},  
{'año': 2005, 'metros': 75, 'habitaciones': 3, 'garaje': True, 'zona': 'B'},  
{'año': 2015, 'metros': 90, 'habitaciones': 2, 'garaje': False, 'zona': 'A'}]
```

Construir una funció que permeti fer recerca d'immobles en funció d'un pressupost donat. La funció rebrà com a entrada la llista d'immobles i un preu, i tornarà una altra llista amb els immobles el preu dels quals sigui menor o igual que el dau. Els immobles de la llista que es tornin han d'incorporar un nou parell a cada diccionari amb el preu de l'immoble, on el preu d'un immoble es calcula amb la fórmula següent en funció de la zona:

- Zona A:  $\text{preu} = (\text{metros} * 1000 + \text{habitacions} * 5000 + \text{garatge} * 15000) * (1 - \text{antiguitat}/100)$
- Zona B:  $\text{preu} = (\text{metros} * 1000 + \text{habitacions} * 5000 + \text{garatge} * 15000) * (1 - \text{antiguitat}/100) * 1.5$

```
pisos = [{'año': 2000, 'metros': 100, 'habitaciones': 3, 'garaje': True, 'zona': 'A'}, {'año': 2012, 'metros': 60, 'habitaciones': 2, 'ga  
raje': True, 'zona': 'B'}, {'año': 1980, 'metros': 120, 'habitaciones': 4, 'garaje': False, 'zona': 'A'}, {'año': 2005, 'metros': 75, 'habi  
taciones': 3, 'garaje': True, 'zona': 'B'}, {'año': 2015, 'metros': 90, 'habitaciones': 2, 'garaje': False, 'zona': 'A'}]
```

**def** *añadir\_precio*(piso):

```
precio = (piso['metros'] * 1000 + piso['habitaciones'] * 5000 + int(piso['garaje']) * 15000) * (1 - (2020 - piso['año']) / 100)  
if piso['zona'] == 'B':  
    precio *= 1.5  
piso['precio'] = precio  
return piso
```

**def** *busca\_piso*(pisos, presupuesto):

**def** *filtro*(piso):

```
return piso['precio'] <= presupuesto  
return list(filter(filtro, map(añadir_precio, pisos)))
```

### Pràctica:

Dissenya un programa diferent, enunciat inclòs, que combini els exemples anteriors, prova'l, mostra'l i enganxa'l aquí.