



# Guía Rápida de *Diccionarios*

IWI 131: Programación

## Creación de diccionarios

<code>{}</code> o <code>dict()</code>	Diccionario vacío
<code>patas = {</code> <code>'humano': 2,</code> <code>'pulpo': 8,</code> <code>'perro': 5,</code> <code>'gato': 4</code> <code>}</code>	Diccionario con valores

## Nota

Las llaves de un diccionario puede ser de tipo int, float, o str. Los valores pueden ser de cualquier tipo de datos, incluso otra estructura como lista u otro diccionario.

## Nota

Los diccionarios no tienen orden. Lo único que se garantiza es que al recorrerlos con un for se recuperarán todas las llaves y, a partir de ahí podemos recuperar los valores asociados usando `[ ]`.

## Operadores sobre diccionarios

<code>[ ]</code>	Recupera el valor asociado a una llave; la llave debe estar en el diccionario, de lo contrario se produce un error Ej: <code>print(patas['gato'])</code> También puede usarse en una asignación, para agregar un par llave-valor o para cambiar el valor asociado a una llave Ej: <code>patas['hormiga'] = 6</code>
<code>k in d</code>	Retorna True si la llave <code>k</code> se encuentra entre las llaves del diccionario <code>d</code> ; de lo contrario retorna False Ej: <code>if 'mosca' in patas:</code> Puede utilizarse <code>not in</code> para preguntar en negativo
<code>del d[k]</code>	Elimina del diccionario la llave <code>k</code> y su valor asociado. Si no existe se produce error Ej: <code>del patas['mosca']</code>

## Recorrido de diccionarios con for

```
for llave in diccionario:  
    print(llave, diccionario[llave])
```

## Patrón de conteo

```
d = {}  
for valor in datos:  
    if valor not in d:  
        d[valor] = 0  
    d[valor] += 1
```

## Patrón de clasificación

```
d = {}  
for dato in datos:  
    categoria = dato[i]  
    elemento = dato[j]  
    if categoria not in d:  
        d[categoria] = [ ]  
    d[categoria].append(elemento)
```