RECURSIVIDAD
CHEATSHEET

Pila

La llamada recursiva es parte del resultado de la función

```python
def funcion(x):
    if isinstance(x, int):
        return funcion_aux(x)
    else:
        return "Error"


def funcion_aux(x):
    if x == 0:
        return 0
    elif x % 2 == 0:
        return x + funcion_aux(x - 1)
```

Cola

El resultado se va arrastrando como parte de un argumento de la funcion

```python
def funcion(x):
    if isinstance(x, int):
        return funcion_aux(x, 0)
    else:
        return "Error"

def funcion_aux(x, resultado):
    if x == 0:
        return resultado
    elif x % 2 == 0:
        return funcion_aux(x - 1, resultado + x)
    else:
        return funcion_aux(x - 1, resultado)
```

Trabajando con números

n % 10 = Toma el residuo 10, osea, el último dígito
(n // 10) % 10 = Toma el penúltimo dígito

n // 10 = Quita el último dígito
(n // 10) // 10 = Quita los dos últimos dígitos

```python
def suma_digitos(num):
    if isinstance(num, int):
        return suma_digitos_aux(num)
    else:
        return "Error"

def suma_digitos_aux(num):
    if num == 0:
        return 0
    else:
        return num % 10 + suma_digitos_aux(num // 10)
```

Trabajando con matrices

i = Fila
j = Columna

¡Cada vez que recorres todas las columnas de una fila, tienes que saltar la fila!

Casi siempre tienen 2 casos base.

```python
def matriz_lista(matriz):
    if isinstance(matriz, list):
        return matriz_lista_aux(matriz, 0, 0)
    else:
        return "Error"

def matriz_lista_aux(matriz, i, j):
    if i == len(matriz): # Primer caso base: Ya no hay mas filas
        return []
    elif j == len(matriz[i]): # Segundo caso base: Se llegó hasta el final
                              # De la fila. Hay que saltar a la siguiente
        return matriz_lista_aux(matriz, i + 1, 0)
    else:
        return [matriz[i][j]] + matriz_lista_aux(matriz, i, j + 1)
```

Trabajando con listas

lista[indice] = *Toma el elemento de la lista en ese indice.*
[2 3 4 5 8 2] = *Elementos*
0 1 2 3 4 5 = *Indices, empiezan en cero*
lista[n:m] = *Toma los elementos de la lista desde una posición n hasta m - 1*
lista[1:] = *Quita el primer elemento de la lista*
lista[0] = *Primer elemento*
¡Cuidado con los casos base, no combinar slicing e indices!

Indices

```python
def suma_digitos(lista):
    if isinstance(lista, list):
        return suma_digitos_aux(lista, 0)
    else:
        return "Error"

def suma_digitos_aux(lista, i):
    if i == len(lista):
        return 0
    else:
        return lista[i] + suma_digitos_aux(lista, i + 1)
```

Slicing

```python
def suma_digitos(lista):
    if isinstance(lista, list):
        return suma_digitos_aux(lista)
    else:
        return "Error"

def suma_digitos_aux(lista):
    if lista == []:
        return 0
    else:
        return lista[0] + suma_digitos_aux(lista[1:])
```