

# CS Bridge

Funciones



# Agenda

- Funciones
  - Sin parámetro, sin retorno
  - Con parámetros
  - Con parámetros y retorno
- Errores comunes

# Ya has usado funciones

```
turn_right()
```

```
move()
```

```
float("0.42")
```

```
print("hola mundo")
```

```
input("digite un número")
```

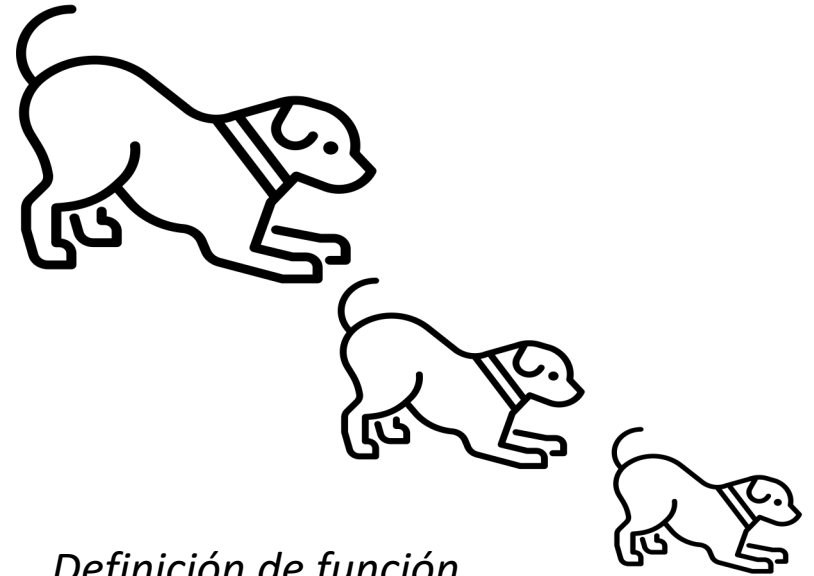
# Ya has definido funciones en Karel

```
def turn_right():  
    turn_left()  
    turn_left()  
    turn_left()
```



La principal diferencia con las funciones Python es:  
Las funciones de Python pueden tomar y devolver datos

# Analogía “buscar juguete”



*Definición de función*  
def buscar\_juguete (juguete) :

1

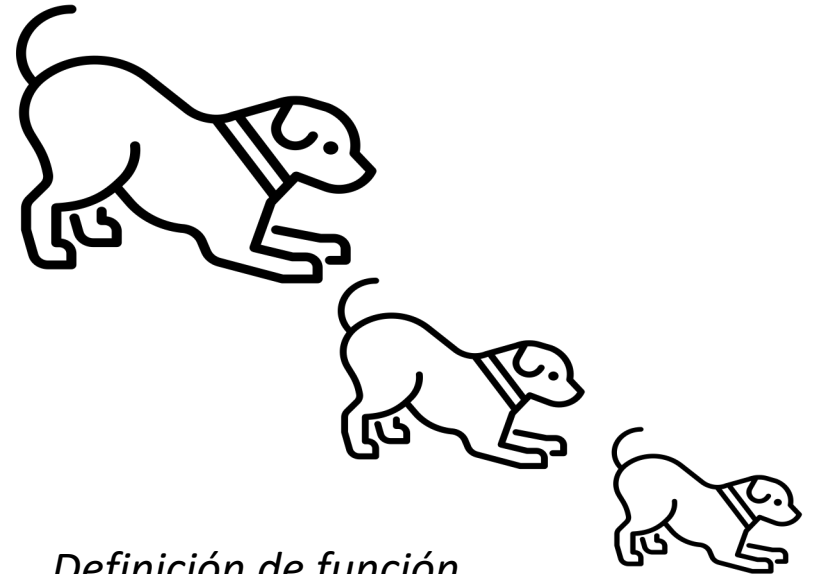
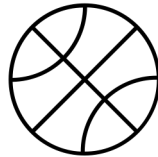
# Analogía “buscar juguete”

2

Programa principal  
U otra función



*Llamado*  
buscar\_juguete(“pelota”)  
Parámetro → “pelota”



*Definición de función*  
def buscar\_juguete (juguete) :

...

# Analogía “buscar juguete”

Programa  
principal  
U otra función

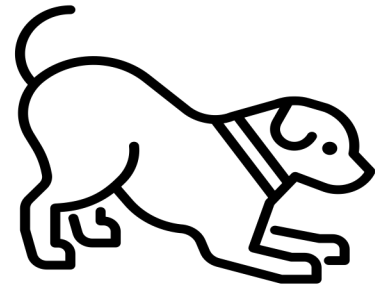


*Retorno*

True



3



*Definición de función*  
def buscar\_juguete (juguete) :

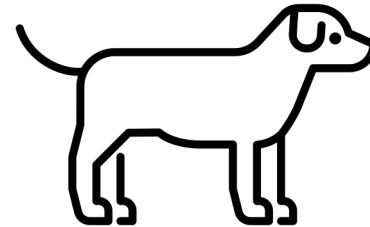
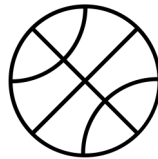
...

# Analogía “buscar juguete”

Programa  
principal  
U otra función



*Llamado*  
buscar\_juguete()  
Parámetro → ninguno



*Definición de función*  
def buscar\_juguete (juguete) :

...





# Formalmente

```
def nombre_de_función (parámetros) :  
    sentencias  
    # opcionalmente  
    return valor
```

- **Nombre:** nombre de la función
- **Parámetros:** información que se pasa a la función
- **Retorno:** información que devuelve la función

# Aprendamos a través de ejemplos



# Funciones sin parámetros, sin retorno

```
def imprimir_saludo():  
    print("Bienvenido a clase")  
    print("Es la mejor parte del día.")
```

```
def programa_principal():  
    imprimir_saludo()
```



# Función con parámetros

```
def imprimir_opinion(num):  
    if(num == 5):  
        print("Me gusta el 5")  
    else :  
        print("Cualquier cosa")
```

```
def programa_principal():  
    imprimir_opinion(5)
```

# Función con parámetros

Memoria principal

*No hay variables*

```
def imprimir_opinion(num) :  
    if (num == 5) :  
        print("Me gusta el 5")  
    else :  
        print("Cualquier cosa")
```

```
def programa_principal() :  
    imprimir_opinion(5)
```

terminal

```
> python3 opinion.py
```

# Función con parámetros

Memoria principal

*No hay variables*

```
def imprimir_opinion(num) :  
    if (num == 5) :  
        print("Me gusta el 5")  
    else :  
        print("Cualquier cosa")  
  
def programa_principal() :  
    imprimir_opinion(5)
```

terminal

```
> python3 opinion.py
```

# Función con parámetros

Memoria principal

*No hay variables*

Memoria imprimir\_opinion



terminal

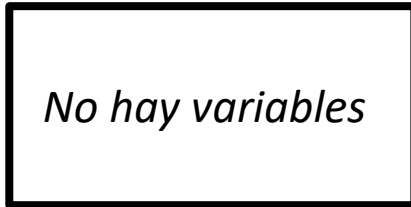
```
> python3 opinion.py
```

```
def imprimir_opinion(num):  
    if(num == 5):  
        print("Me gusta el 5")  
    else :  
        print("Cualquier cosa")
```

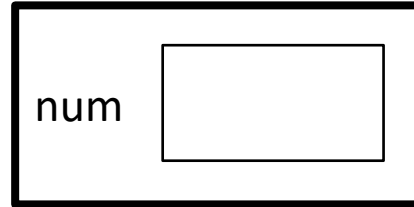
```
def programa_principal():  
    imprimir_opinion(5)
```

# Función con parámetros

Memoria principal



Memoria imprimir\_opinion



terminal

```
> python3 opinion.py
```

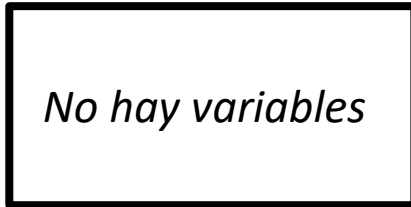
```
def imprimir_opinion(num):  
    if (num == 5):  
        print("Me gusta el 5")  
    else :  
        print("Cualquier cosa")
```

```
def programa_principal():  
    imprimir_opinion(5)
```

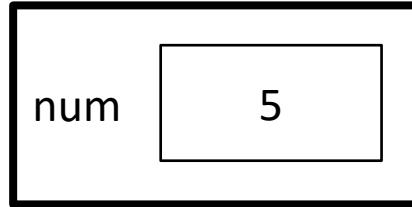


# Función con parámetros

Memoria principal



Memoria imprimir\_opinion



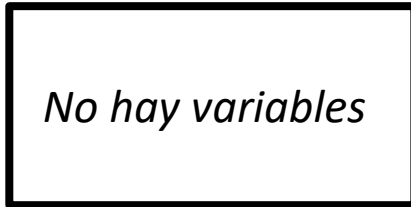
terminal

```
> python3 opinion.py
```

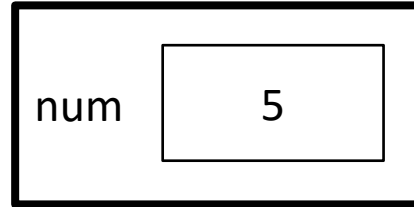
```
def imprimir_opinion(num):  
    if(num == 5):  
        print("Me gusta el 5")  
    else :  
        print("Cualquier cosa")  
  
def programa_principal():  
    imprimir_opinion(5)
```

# Función con parámetros

Memoria principal



Memoria imprimir\_opinion



terminal

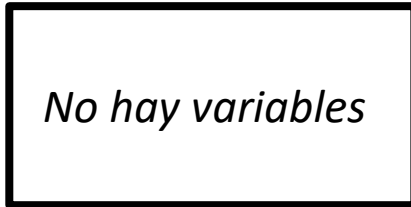
```
> python3 opinion.py
```

```
def imprimir_opinion(num) :  
    if (num == 5) :  
        print("Me gusta el 5")  
    else :  
        print("Cualquier cosa")
```

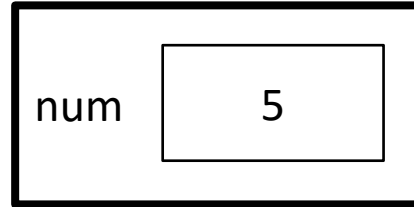
```
def programa_principal() :  
    imprimir_opinion(5)
```

# Función con parámetros

Memoria principal



Memoria imprimir\_opinion



terminal

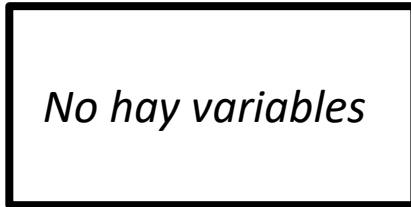
```
> python3 opinion.py  
Me gusta el 5
```

```
def imprimir_opinion(num):  
    if (num == 5):  
        print("Me gusta el 5")  
    else :  
        print("Cualquier cosa")
```

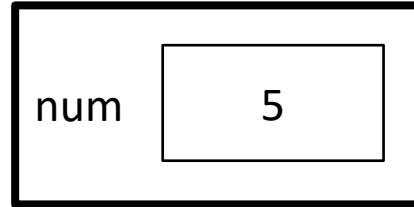
```
def programa_principal():  
    imprimir_opinion(5)
```

# Función con parámetros

Memoria principal



Memoria imprimir\_opinion



terminal

```
> python3 opinion.py  
Me gusta el 5
```

```
def imprimir_opinion(num):  
    if (num == 5):  
        print("Me gusta el 5")  
    else :  
        print("Cualquier cosa")
```



```
def programa_principal():  
    imprimir_opinion(5)
```

# Función con parámetros

Memoria principal

*No hay variables*

```
def imprimir_opinion(num):  
    if(num == 5):  
        print("Me gusta el 5")  
    else :  
        print("Cualquier cosa")
```



```
def programa_principal():  
    imprimir_opinion(5)
```

terminal

```
> python3 opinion.py  
Me gusta el 5
```

# Función con parámetro y retorno

```
def metros_a_cm(metros) :  
    return 100 * metros
```

```
def programa_principal():  
    resultado = metros_a_cm(5.2)  
    print(resultado)
```

terminal

```
> python3 m2cm.py
```

# Función con parámetro y retorno

Memoria principal

No hay variables

```
def metros_a_cm(metros):  
    return 100 * metros
```

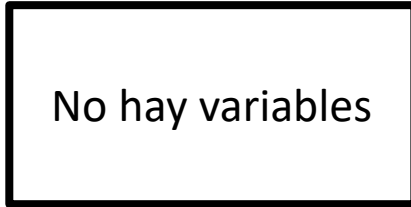
```
def programa_principal():  
    resultado = metros_a_cm(5.2)  
    print(resultado)
```

terminal

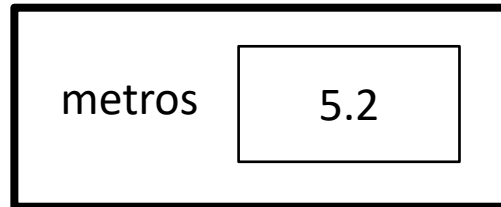
```
> python3 m2cm.py
```

# Función con parámetro y retorno

Memoria principal



Memoria metros\_a\_cm



terminal

```
> python3 m2cm.py
```

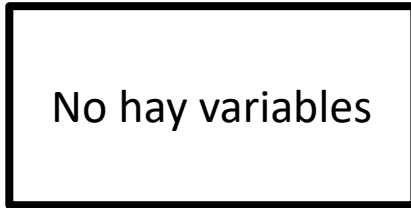
```
def metros_a_cm(metros):  
    return 100 * metros
```

```
def programa_principal():  
    resultado = metros_a_cm(5.2)  
    print(resultado)
```

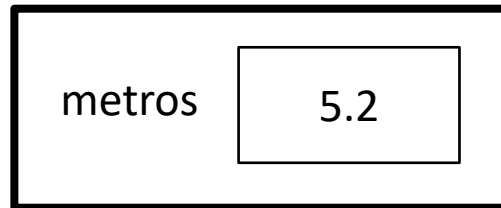


# Función con parámetro y retorno

Memoria principal



Memoria metros\_a\_cm



terminal

```
> python3 m2cm.py
```

```
def metros_a_cm(metros):  
    return 100 * metros
```

520.0

```
def programa_principal():  
    resultado = metros_a_cm(5.2)  
    print(resultado)
```

# Función con parámetro y retorno

Memoria principal

No hay variables

terminal

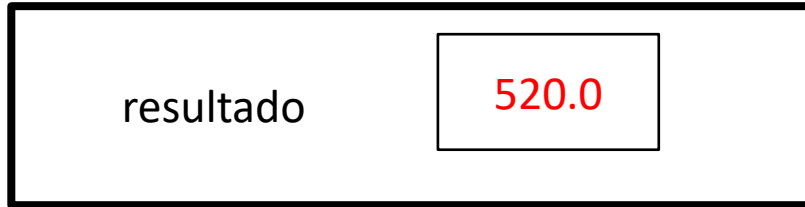
```
> python3 m2cm.py
```

```
def metros_a_cm(metros):  
    return 100 * metros
```

```
def programa_principal(): 520.0  
    resultado = metros_a_cm(5.2)  
    print(resultado)
```

# Función con parámetro y retorno

Memoria principal



terminal

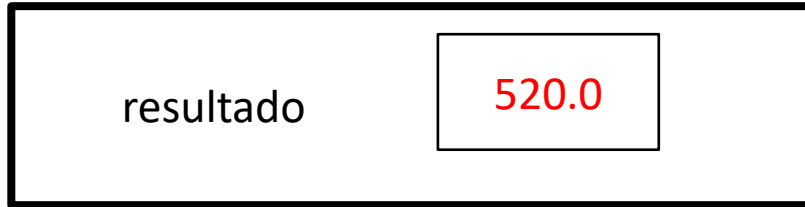
```
> python3 m2cm.py
```

```
def metros_a_cm(metros):  
    return 100 * metros
```

```
def programa_principal(): 520.0  
    resultado = metros_a_cm(5.2)  
    print(resultado)
```

# Función con parámetro y retorno

Memoria principal



terminal

```
> python3 m2cm.py  
520.0
```

```
def metros_a_cm(metros):  
    return 100 * metros
```

```
def programa_principal(): 520.0  
    resultado = metros_a_cm(5.2)  
    print(resultado)
```

# Función con parámetro y retorno

```
def metros_a_cm(metros) :  
    return 100 * metros
```

```
def programa_principal():  
    print(metros_a_cm(5.2))  
    print(metros_a_cm(9.1))
```

terminal

```
> python3 m2cm.py
```

# Función con parámetro y retorno

terminal

```
> python3 m2cm.py
```

```
def metros_a_cm(metros) :  
    return 100 * metros
```

```
def programa_principal() :  
    print(metros_a_cm(5.2))  
    print(metros_a_cm(9.1))
```

Si un método retorna algo, se puede  
usar directo en una expresión

# ¿Qué tienen de diferente las funciones?

```
def metros_a_cm_caso1(metros):  
    return 100 * metros
```

```
def metros_a_cm_caso2(metros):  
    print(100 * metros)
```



¿Es lo mismo retornar que  
imprimir?



¿Es lo mismo retornar que  
imprimir?

NO

# Múltiples instrucciones de retorno

```
def max(num1, num2):  
    if num1 >= num2:  
        return num1  
  
    return num2
```

```
def programa_principal():  
    mas_grande = max(5, 1)
```

terminal

```
> python3 maxmax.py
```

# Múltiples instrucciones de retorno

Memoria principal

*No variables*

Memoria max

num1	5	num2	1
------	---	------	---

```
def max(num1, num2):  
    if num1 >= num2:  
        return num1  
  
    return num2
```

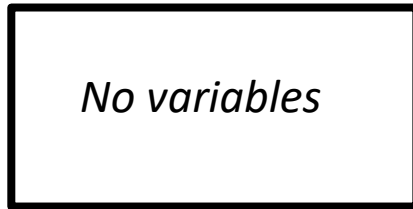
```
def programa_principal():  
    mas_grande = max(5, 1)
```

terminal

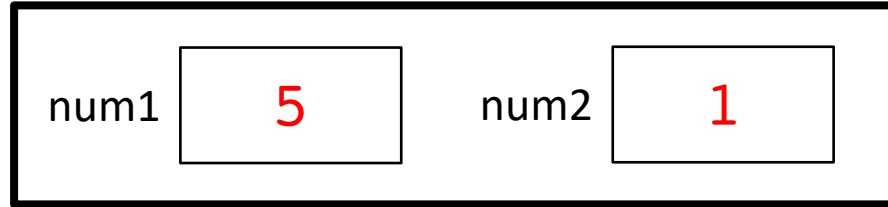
```
> python3 maxmax.py
```

# Múltiples instrucciones de retorno

Memoria principal



Memoria max



```
def max(num1, num2):  
    if num1 >= num2:  
        return num1  
  
    return num2
```

```
def programa_principal():  
    mas_grande = max(5, 1)
```

terminal

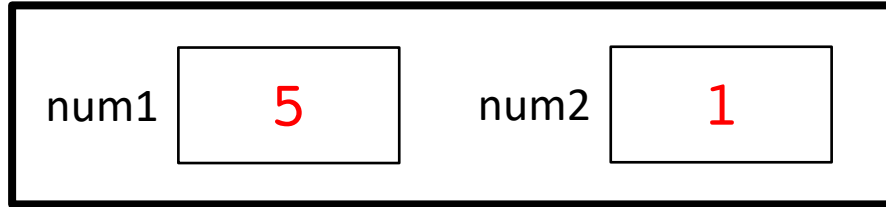
```
> python3 maxmax.py
```

# Múltiples instrucciones de retorno

Memoria principal

*No variables*

Memoria max



```
def max(num1, num2):
```

```
    if num1 >= num2:
```

```
        return num1
```

5

```
    return num2
```

```
def programa_principal():
```

```
    mas_grande = max(5, 1)
```

terminal

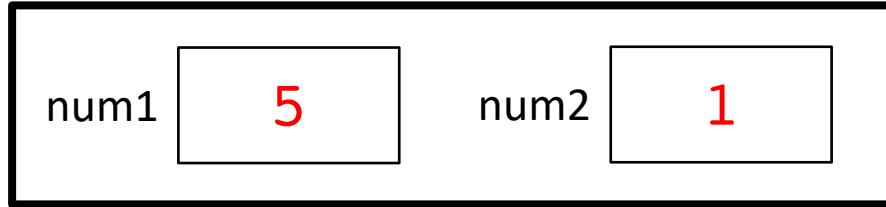
```
> python3 maxmax.py
```

# Múltiples instrucciones de retorno

Memoria principal

*No variables*

Memoria max



```
def max(num1, num2):
```

```
    if num1 >= num2:
```

```
        return num1
```

5

```
    return num2
```

```
def programa_principal():
```

```
    mas_grande = max(5, 1)
```

terminal

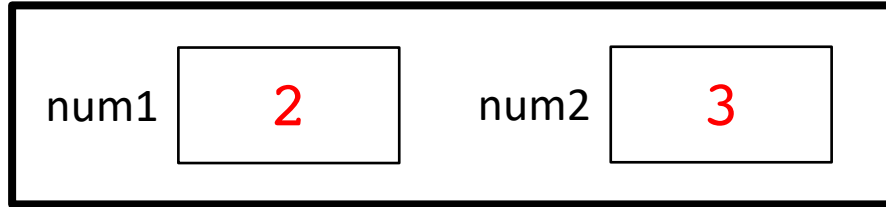
```
> python3 maxmax.py
```

# Múltiples instrucciones de retorno

Memoria principal

*No variables*

Memoria max



```
def max(num1, num2):  
    if num1 >= num2:  
        return num1
```

```
    return num2
```

```
def programa_principal():  
    mas_grande = max(2, 3)
```

terminal

```
> python3 maxmax.py
```

# Problemas con funciones





# ¿Qué revisamos en este rato?

- Funciones
  - Sin parámetro, sin retorno
  - Con parámetros
  - Con parámetros y retorno
- Errores comunes