

# Introducción a Python

Universidad Tecnológica de Panamá  
Facultad de Ingeniería Mecánica  
Departamento de Control y Automatización

Programación estructurada, funcional y orientada a objetos  
Febrero 6, 2024



# Contenido

Primeros pasos

Datos y tipos

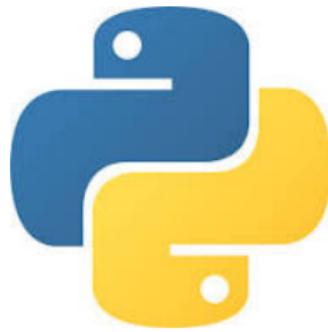
Funciones y Funcionales

Clases, objetos e interface gráfica

Aplicaciones avanzadas

# Empezando a aprender para poder desarrollar luego

```
>>>print("Hello World!")
```



-Al iniciar un entorno de línea de comandos, puede enviar comandos de texto al sistema operativo. Se trata principalmente de instrucciones para ejecutar otros programas.

# Empezando a aprender para poder desarrollar luego

-Sistemas operativos Windows, MacOs y Linux soportan python.

▶ Link

▶ Link

CMD

C:\Users\isaach>python3 #también puede escribirse python



# Empezando a aprender para poder desarrollar luego

-Asistentes Cibernéticos: AI y sitios de discusión

▶ Link

▶ Link

simple python code escribe en el buzón de petición

-ADVERTENCIA: Python 3 es la opción más nueva, fácil de aprender, segura y potente. Desde que Python dejó de dar soporte a Python 2, Python 3 es la opción más obvia para los nuevos desarrolladores. Comenzar con el lenguaje más actualizado es la elección más segura.

# Mis herramientas

-IDEs: múltiples habilidades en una sola *aplicación*. **All in one: coding, automation and debugging.**

▶ Link

▶ Link

▶ Link

opciones con el shell ó con ANACONDA

CMD

```
C:\Users\isaach>code #es casi igual que pycharm  
anaconda prompt  
(base) C:\Users\isaach>jupyter notebook
```



# Competencias necesarias

**"La actividad esencial de un programador es resolver problemas."**

-El único *lenguaje* que el ordenador puede "entender" realmente es muy distinto del que utilizamos nosotros mismos. El lenguaje que aprenderás en el curso es Python. Se trata de un lenguaje de alto nivel cuya traducción a código binario es compleja y por lo tanto siempre lleva algún tiempo dominarlo.

-El texto producido es lo que a partir de ahora llamaremos código fuente.

-A los errores de programación se les llama "bugs"; y el conjunto de técnicas utilizadas para detectarlos y corregirlos se denomina *debug*.

# Competencias necesarias

**"La actividad esencial de un programador es resolver problemas."**

-Una de las competencias más importantes que hay que adquirir durante el aprendizaje es hacer eficazmente **debug** de un programa. Esta actividad intelectual puede resultar un tanto irritante, pero siempre es muy enriquecedora y requiere mucha perspicacia.

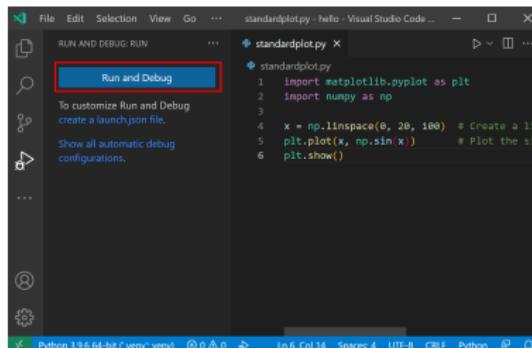


Figura 1.1. Testing y debug de código en la IDE de visual studio.

# Escritura

**"Un buen programador debe asegurarse de que sus líneas de instrucciones sean fáciles de leer."**

-Los operadores aritméticos para la suma, la resta, la multiplicación y la división son +, -, \* y / respectivamente. Los paréntesis incorporan operaciones y se usan en funciones.

-En Python, los nombres de las variables también deben obedecer unas sencillas reglas:

- Un nombre de variable es una secuencia de letras (a → z , A → Z) y números (0 → 9), que debe comenzar siempre por una letra.
- Sólo se admiten letras normales. Se prohíben las letras acentuadas, las cedillas, los espacios, los especiales como \$, #, @, etc. están prohibidos, a excepción del carácter \_ (subrayado)

# Escritura

**"Un buen programador debe asegurarse de que sus líneas de instrucciones sean fáciles de leer."**

-Hay dos formas de mostrar su valor en la pantalla. La primera es utilizar el teclado para introducir el nombre de la variable y luego <ENTER>. Python responde mostrando el valor correspondiente:

```
>>> n = 10 # variable n con valor de 10
>>> msg = "que onda wey!" #asignar la frase
>>> cr_7 = 0.72 #asignar el valor decimal
>>> n
10
>>> msg
'que onda wey!'
>>> cr_7
0.72
```

# Secuencia de Instrucciones

Que pasa con lo siguiente:

```
>>> x, y = 3, 7  
>>> x = y  
>>> y = x  
>>> print(x, y)
```

# Condicionamiento

El comando de instrucción por defecto es `if`

-Necesitamos instrucciones que puedan comprobar una determinada condición y modificar el comportamiento del programa en consecuencia.

```
>>> a = 50
>>> if (a > 75):
...     print("sobrepasa el valor superior")
... else:
...     print("menor al valor superior")
```

# Comparación

Los operadores son `==`, `!=`, `<=`, `>=`, `<`, `>`

-Observe que el operador de comparación para la igualdad de dos valores consta de dos signos y no un único.

```
>>> x = 7
>>> if (x % 2 == 0):
...     print("es par")
... else:
...     print("es impar")
```

# Ejercicios

Empiece a trabajar : ya sea en IDE o shell. Ayúdese con los asistentes AI en caso de tener muchas dificultades en la escritura y la sintaxis.

-Pruebe y comente que pasa con las siguientes líneas, en especial `type`:

```
>>> x , pi = 5 , 3.14159  
>>> y = pi * x**2  
>>> print(y)  
>>> print(type(x) , type(pi) , type(y))
```

# Secuencias y bucles para manipular datos

Nota: Siempre verifique la identación de las líneas de comandos.

## Secuencia de Fibonacci

```
a, b, c = 1, 1, 1
while c < 11 :
    print(b, end = " ")
    a, b, c = b, a+b, c+1
```

# Datos de números enteros-integer

Supongamos que queremos modificar ligeramente nuestro ejercicio anterior sobre la secuencia de Fibonacci, para mostrar un mayor número de términos.

`type()`:

```
>>> a, b, c = 1, 1, 1
>>> while c <50:
        print(c, ":", b, type(b))
        a, b, c = b, a+b, c+1
```

# Datos decimales-float

Este tipo permite realizar cálculos sobre números muy grandes o muy pequeños (datos científicos, por ejemplo), con un grado de precisión constante.

ejemplo:

```
>>> a, b, c = 1., 2., 1 # => a y b tipo 'float'  
>>> while c <11:  
...     a, b, c = b, b*a, c+1  
...     print(b)
```

# Datos string

En una primera aproximación, los datos de string pueden definirse como cualquier secuencia de caracteres..

ejemplo:

```
>>> frase1 = ' es la cerveza artesanal '
>>> frase2 = '"Si", responde ,'
>>> frase3 = "su favorita"
>>> print(frase2 , frase3 , frase1)
```

# Casos de conversión de datos

Convierte una cadena de caracteres que representa un número en un número real.

```
ch = '5476'  
print(ch + 86)  
n = int(ch)  
print(n + 65)
```

En este ejemplo, la función `int()` convierte la cadena en un número entero. También es posible convertir una cadena de caracteres en un número real, utilizando la función incorporada `float()`.

# Listas

En Python, una lista puede definirse como una colección de elementos separados por comas, con el conjunto entre corchetes. Ejemplo:

ejemplo:

```
>>> day = ['monday', 'tuesday',
'miercoles', 1500, 40.123, 'Samstag',
'Sonntag']
>>> print(day)
```

# Funciones predefinidas

Nota: Siempre verifique la identación de las líneas de comandos.

## Función `input()`

el método más sencillo es utilizar la función función `input()`. Esta función provoca una interrupción en el programa actual.

```
>>> x = input("poner un numero : ")
poner un numero : 3.1416
>>> type(x)
<class 'str'>
>>> y = float(x)
>>> type(y)
<class 'float'>
```

# Otros datos

## Datos complejos

```
>>> x=1+2j
>>> type(x)
<type 'complex'>
>>> x.real
1.0
>>> x.imag
2.0
>>> x.conjugate()
(1-2j)
```

# Otros datos

## Datos booleanos

```
>>> x=0  
>>> y=1  
>>> bool(x)  
False  
>>> bool(y)  
True  
>>> bool(1+1j)  
True  
>>> bool(0+0j)  
False
```

# Definir datos secuenciales y peculiaridades

- **x in s** comprueba si x está en s (resultado booleano)
- **x not in s** comprueba si x no está en s (resultado booleano)
- **s+t** genera la concatenación de s y t como una nueva secuencia
- **s+=t** añade el elemento t a la secuencia s
- **s\*n** devuelve una nueva secuencia en la que la secuencia s se copia n veces
- **s\*n** devuelve una nueva secuencia en la que la secuencia s se copia n veces
- **s[i]** devuelve el elemento i de la secuencia s

# Definir datos secuenciales y peculiaridades

- **s[i:j]** devuelve la sección desde el índice i hasta el índice j de s
- **s[i:j:k]** devuelve la sección desde el índice i hasta el índice j de s, por lo que sólo se tiene en cuenta cada k-ésimo elemento dentro de esta sección
- **len(s)** devuelve el número de elementos de s
- **min(s), max(s)** devuelve el elemento más pequeño de s o más grande si se define una relación de orden para los elementos de s
- **append(s)** agrega una secuencia o lista
- **del(s)** elimina una secuencia o lista

# Definir datos secuenciales y peculiaridades

- **extend(x)** añade los elementos de x al final de la lista
- **count(x)** cuenta la frecuencia con la que el elemento x aparece en s
- **index(x)** devuelve el índice en el que el valor x aparece por primera vez en s
- **insert(i,x)** inserta x en el índice i de s
- **sort()** ordena la lista según valores ascendentes. Con *s.sort(reverse=True)* la lista se ordena en orden descendente
- **range(a,b,c)** Los tres argumentos deben ser números enteros. Los argumentos inicio y step son opcionales.

# Trabajar con un string

Los siguientes métodos, entre otros, pueden utilizarse para dividir cadenas en varios componentes, así como buscar elementos:

*s.split(sep)*: separa la cadena s en los puntos donde aparece el separador *sep* en s. Si no se pasa ningún separador al método como argumento, s se separa en todos los espacios en blanco.

*s.splitlines()*: Divide el string en saltos de línea y devuelve una lista con las líneas individuales.

*s.find(x, start, end)*: devuelve el índice de la primera aparición de x en la cadena s

# Importar funciones

## Comando import

Importar se utiliza para llamar módulos o partes de módulos a otro módulo o a la consola interactiva. La instrucción import puede utilizarse en distintas variantes.

```
>>> import math  
>>> math.sqrt(25)  
5.0  
>>> math.cos(math.pi)  
-1.0  
>>> math.pow(4,4)  
256.0
```

# Importar funciones

## Comando `import`

Con esta variante, también se importa el módulo completo, pero ahora los nombres de las funciones se pueden llamar directamente, es decir, sin el prefijo con el nombre del módulo.

```
>>> from math import *
>>> sqrt(81)
9.0
```

# Importar funciones

## Comando import

Con esta variante, sólo se importan del módulo las funciones dedicadas. Se puede acceder directamente a las funciones importadas especificando el nombre de la función.

```
>>> from math import sqrt, sin  
>>> sin(0.5)  
0.4794255386  
>>> sqrt(49)  
7.0
```

# Importar funciones

## Comando `import`

También está la opción de especificar las funciones con su afiliación al módulo sin tener que introducir un nombre de módulo posiblemente largo antes de cada función.

```
>>> import math as m  
>>> m.sin(m.pi/2)  
1.0
```

# Verdadero o Falso

## Comando `bool`

La función `bool` asume exactamente dos estados: `True` (verdadero) o `False` (falso, no verdadero).

```
>>> for i in [],(),{},False,0,'',None,22:  
...     print (bool(i))  
  
...  
False  
False  
False  
False  
False  
False  
False  
True
```

# Ejercicios

Empiece a trabajar : ya sea en IDE o shell. Ayúdese con los asistentes AI en caso de tener muchas dificultades en la escritura y la sintaxis.

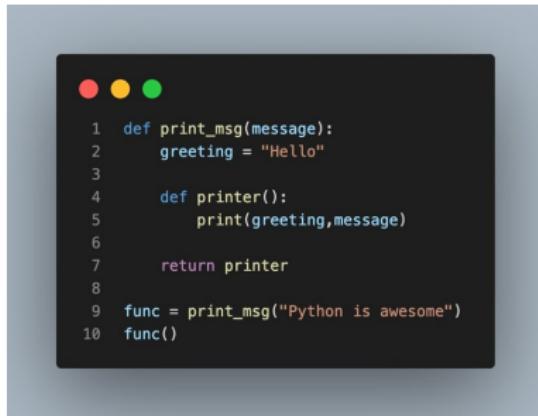
-Pruebe y comente que pasa con las siguientes líneas, en especial [con el resultado de set](#):

```
lst_numbers = [1, 2, 1, 3, 1, 4]
set_numbers = set(lst_numbers)
newList = list(set_numbers)
print(set_numbers)
print(newList)
```

# Funciones Originales

**"El arte de la simplicidad es un rompecabezas complejo."**

-Las funciones son "áreas de código separadas" que se ejecutan independientemente del resto del código.



A screenshot of a dark-themed code editor window. At the top, there are three colored circular icons: red, yellow, and green. Below them, the Python code is displayed:

```
1 def print_msg(message):
2     greeting = "Hello"
3
4     def printer():
5         print(greeting,message)
6
7     return printer
8
9 func = print_msg("Python is awesome")
10 func()
```

Figura 3.1. Ejemplo de una función.

# Funciones Originales

## Sintaxis de funciones

```
def NombreFuncion(Listaparametros) :  
    codigo normal  
    ...  
    return Resultados
```

La lista de parámetros es una lista de nombres de variables separados por comas. Estos nombres de variables se utilizan para "canalizar información hacia la función desde el exterior"(transferencia de valores). Entiéndase la lista de parámetros como un punto en el que se crean variables.

# Ejemplo

## Función exponencial

```
def Funcionexp(y0) :  
    y = 1.0  
    denominador = 1.0  
    resultado = 1.0  
    iteraciones = 10  
    for k in range(1, iteraciones) :  
        y *= y0  
        denominador *= k  
        resultado += y / denominador  
    return resultado  
  
NroEuler = Funcionexp(1.0)  
print(f"el numero de euler es {NroEuler} ")  
print(f"e3 = {Funcionexp(3.0)}")
```

# Asignar variables por valor o por referencia

Nos hemos dado cuenta de que las variables que utilizamos en una función están separadas del resto del código. ¿Pero qué pasa con los parámetros?

ejemplo:

```
def tempovar(nd) :  
    nd.append(1)  
    nd = nd + [2]  
    print("in tempovar: nd =", nd)  
nd = []  
tempovar(nd)  
print("a nivel de modulo: nd =", nd)
```

# Variables Locales

Nota: Siempre verifique la identación de las líneas de comandos.

## Locales vs Globales

Digamos que definimos un string `¡Me encanta Panama en verano!`, antes de llamar a `f()`. El cuerpo de `f()` consiste únicamente en la línea `print(s)`. Como no hay variable local `s`, es decir, no hay asignación a `s`, se utilizará el valor de la variable global `s`. La salida será el string: `"¡Me encanta Panama en verano!"`. Preguntas, ¿qué ocurrirá si cambia el valor de `s` dentro de `f()`? ¿Afectará también a la variable global? :

```
def f():
    s = "Me encanta Miami!"
    print(s)
s = "Me encanta Panama!"
f()
print(s)
```

# Argumentosopcionales: Parámetros Keywords

El scope permite utilizar parámetros posicionales, parámetros de palabra clave, parámetros variádicos y parámetros de **palabra clave** variados en cualquier combinación.

ejemplo:

```
def introperso(**kwargs) :  
    print("algunos detalles acerca de mí:")  
    for (key, value) in kwargs.items():  
        print(key, ": ", value)  
  
introperso(  
    apodo="El bicho",  
    talentos="tomar cerveza",  
    Nrodecel="pregunta amablemente"  
)
```

# Argumentosopcionales: Parámetros por defecto

Los parámetros que sólo pueden asignarse a través de su posición en la lista se denominan parámetros posicionales.

ejemplo:

```
def defaults(m, n, p=0, q=0) :  
    print(m, n, p, q)  
  
defaults(3, 5, q = 7)  
defaults(m=10, n=13)  
defaults(5, p=20, n=12)
```

# Argumentosopcionales: Parámetros por defecto

Los parámetros que sólo pueden asignarse a través de su posición en la lista se denominan parámetros posicionales.

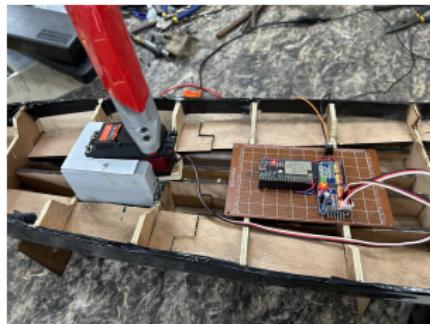
ejemplo:

```
def defaults(m, n, p=0, q=0) :  
    print(m, n, p, q)  
  
defaults(3, 5, q = 7)  
defaults(m=10, n=13)  
defaults(5, p=20, n=12)
```

# First steps to make a functional USV



```
Sketch: servoControl.ino
2
3   Servo Motor Control - 898: Pollar Triple Servo
4   by Dejan, https://hostwithcode.com
5
6 #define servopin 9
7
8 void setup() {
9   // initialize serial port:
10  //Serial.begin(9600);
11
12  // Set the servo to 180 degrees
13  //digitalWrite(servopin, HIGH);
14  //delayMicroseconds(4000); // Duration of the pulse in microseconds
15  //digitalWrite(servopin, LOW);
16  //delayMicroseconds(4000); // Duration of the pulse
17  //digitalWrite(servopin, HIGH);
18  //delayMicroseconds(4000);
19}
20
21 void loop() {
22   // A pulse with 2ms
23   digitalWrite(servopin, HIGH);
24   delayMicroseconds(4000); // Duration of the pulse in microseconds
25   digitalWrite(servopin, LOW);
26   delayMicroseconds(4000); // Duration of the pulse
27   // Pulse duration = 0.5ms * 5000 = 2.5ms
28 }
```



# Argumentosopcionales: Parámetros por defecto

Los parámetros que sólo pueden asignarse a través de su posición en la lista se denominan parámetros posicionales.

ejemplo:

```
def defaults(m, n, p=0, q=0) :  
    print(m, n, p, q)  
  
defaults(3, 5, q = 7)  
defaults(m=10, n=13)  
defaults(5, p=20, n=12)
```

# Argumentosopcionales: Parámetros por defecto

Los parámetros que sólo pueden asignarse a través de su posición en la lista se denominan parámetros posicionales.

ejemplo:

```
def defaults(m, n, p=0, q=0) :  
    print(m, n, p, q)  
  
defaults(3, 5, q = 7)  
defaults(m=10, n=13)  
defaults(5, p=20, n=12)
```

# Argumentosopcionales: Parámetros por defecto

Los parámetros que sólo pueden asignarse a través de su posición en la lista se denominan parámetros posicionales.

ejemplo:

```
def defaults(m, n, p=0, q=0) :  
    print(m, n, p, q)  
  
defaults(3, 5, q = 7)  
defaults(m=10, n=13)  
defaults(5, p=20, n=12)
```

# Argumentosopcionales: Parámetros por defecto

Los parámetros que sólo pueden asignarse a través de su posición en la lista se denominan parámetros posicionales.

ejemplo:

```
def defaults(m, n, p=0, q=0) :  
    print(m, n, p, q)  
  
defaults(3, 5, q = 7)  
defaults(m=10, n=13)  
defaults(5, p=20, n=12)
```

# Argumentosopcionales: Parámetros por defecto

Los parámetros que sólo pueden asignarse a través de su posición en la lista se denominan parámetros posicionales.

ejemplo:

```
def defaults(m, n, p=0, q=0) :  
    print(m, n, p, q)  
  
defaults(3, 5, q = 7)  
defaults(m=10, n=13)  
defaults(5, p=20, n=12)
```

# Argumentosopcionales: Parámetros por defecto

Los parámetros que sólo pueden asignarse a través de su posición en la lista se denominan parámetros posicionales.

ejemplo:

```
def defaults(m, n, p=0, q=0) :  
    print(m, n, p, q)  
  
defaults(3, 5, q = 7)  
defaults(m=10, n=13)  
defaults(5, p=20, n=12)
```

# Argumentosopcionales: Parámetros por defecto

Los parámetros que sólo pueden asignarse a través de su posición en la lista se denominan parámetros posicionales.

ejemplo:

```
def defaults(m, n, p=0, q=0) :  
    print(m, n, p, q)  
  
defaults(3, 5, q = 7)  
defaults(m=10, n=13)  
defaults(5, p=20, n=12)
```

# Argumentosopcionales: Parámetros por defecto

Los parámetros que sólo pueden asignarse a través de su posición en la lista se denominan parámetros posicionales.

ejemplo:

```
def defaults(m, n, p=0, q=0) :  
    print(m, n, p, q)  
  
defaults(3, 5, q = 7)  
defaults(m=10, n=13)  
defaults(5, p=20, n=12)
```

# Argumentosopcionales: Parámetros por defecto

Los parámetros que sólo pueden asignarse a través de su posición en la lista se denominan parámetros posicionales.

ejemplo:

```
def defaults(m, n, p=0, q=0) :  
    print(m, n, p, q)  
  
defaults(3, 5, q = 7)  
defaults(m=10, n=13)  
defaults(5, p=20, n=12)
```

# Argumentosopcionales: Parámetros por defecto

Los parámetros que sólo pueden asignarse a través de su posición en la lista se denominan parámetros posicionales.

ejemplo:

```
def defaults(m, n, p=0, q=0) :  
    print(m, n, p, q)  
  
defaults(3, 5, q = 7)  
defaults(m=10, n=13)  
defaults(5, p=20, n=12)
```

# Argumentosopcionales: Parámetros por defecto

Los parámetros que sólo pueden asignarse a través de su posición en la lista se denominan parámetros posicionales.

ejemplo:

```
def defaults(m, n, p=0, q=0) :  
    print(m, n, p, q)  
  
defaults(3, 5, q = 7)  
defaults(m=10, n=13)  
defaults(5, p=20, n=12)
```



# References

- Alves, C., Aguiar, P., Sebatiao, L. (2010). Sailboat: Autonomous Marine Robot of Eolic Propulsion . *Instituto Superior Técnico – Institute for Systems and Robotics (IST-ISR) Lisbon 1049-001, Portugal.*
- Giraldo, Nelson(2023). Sailboat navigation control system based on spiking neural networks. Master thesis, Universidad de Antioquia.
- Chacon, Esteban (2021). Autonomous Sailboat Prototype Sensors and Electronics Implementation with Machine Learning for Navigation, Masther's thesis, Universidad Politécnica de Catalunya.
- Dhomé, Ulyssen (2021). Further development and performance evaluation of the autonomous sailing boat Maribot Vane, Degre Project in Naval Architecture, KTH Royal Institute of Technology Sweden.
- Fossen, T. I., Pettersen, K. Y., & Galeazzi, R. (2015). Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827. <https://doi.org/10.1109/TCST.2014.2338354>
- Plummet, F., Petres, C., Rómero-Ramírez, M. A., Gas, B., Sio-Hoi, I. (2015). Towards an Autonomous Sailing Boat. *IEEE Journal of Oceanic Engineering*, pp. 397–407. DOI: 10.1109/JOE.2014.2321714.
- Tranzatto, Marco (2015a) Navigation and Control for an Autonomous Sailing Model Boat. Chapter 5: Tracking a Constant Heading pp. 25-29. Master Thesis, University of Pisa.
- Utkin, V. (2002) Sliding mode Control, p. 23. EOSS: UNESCO Encyclopedia of Life Support Systems.
- Velueta Guzmán, M. J., Rullán Lara, J. L., Ruiz Hernández, J. A., & Alazki, H. (2018).Robust control for the dynamics of an unmanned surface vehicle under the perturbations (marine waves and currents). *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6. <https://doi.org/10.1109/ICEEE.2018.8533948>
- von Ellenrieder, K. D. (2021b). Control of Marine Vehicles. *Sliding Mode Control*, pp. 489–526. Cham: Springer International Publishing.



# References

- Alves, C., Aguiar, P., Sebatiao, L. (2010). Sailboat: Autonomous Marine Robot of Eolic Propulsion . *Instituto Superior Técnico – Institute for Systems and Robotics (IST-ISR) Lisbon 1049-001, Portugal.*
- Giraldo, Nelson(2023). Sailboat navigation control system based on spiking neural networks. Master thesis, Universidad de Antioquia.
- Chacon, Esteban (2021). Autonomous Sailboat Prototype Sensors and Electronics Implementation with Machine Learning for Navigation, Masther's thesis, Universidad Politécnica de Catalunya.
- Dhomé, Ulyssen (2021). Further development and performance evaluation of the autonomous sailing boat Maribot Vane, Degre Project in Naval Architecture, KTH Royal Institute of Technology Sweden.
- Fossen, T. I., Pettersen, K. Y., & Galeazzi, R. (2015). Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827. <https://doi.org/10.1109/TCST.2014.2338354>
- Plummet, F., Petres, C., Rómero-Ramírez, M. A., Gas, B., Sio-Hoi, I. (2015). Towards an Autonomous Sailing Boat. *IEEE Journal of Oceanic Engineering*, pp. 397–407. DOI: 10.1109/JOE.2014.2321714.
- Tranzatto, Marco (2015a) Navigation and Control for an Autonomous Sailing Model Boat. Chapter 5: Tracking a Constant Heading pp. 25-29. Master Thesis, University of Pisa.
- Utkin, V. (2002) Sliding mode Control, p. 23. EOSS: UNESCO Encyclopedia of Life Support Systems.
- Velueta Guzmán, M. J., Rullán Lara, J. L., Ruiz Hernández, J. A., & Alazki, H. (2018).Robust control for the dynamics of an unmanned surface vehicle under the perturbations (marine waves and currents). *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6. <https://doi.org/10.1109/ICEEE.2018.8533948>
- von Ellenrieder, K. D. (2021b). Control of Marine Vehicles. *Sliding Mode Control*, pp. 489–526. Cham: Springer International Publishing.



# References

- Alves, C., Aguiar, P., Sebatiao, L. (2010). Sailboat: Autonomous Marine Robot of Eolic Propulsion . *Instituto Superior Técnico – Institute for Systems and Robotics (IST-ISR) Lisbon 1049-001, Portugal.*
- Giraldo, Nelson(2023). Sailboat navigation control system based on spiking neural networks. Master thesis, Universidad de Antioquia.
- Chacon, Esteban (2021). Autonomous Sailboat Prototype Sensors and Electronics Implementation with Machine Learning for Navigation, Masther's thesis, Universidad Politécnica de Catalunya.
- Dhomé, Ulyssen (2021). Further development and performance evaluation of the autonomous sailing boat Maribot Vane, Degre Project in Naval Architecture, KTH Royal Institute of Technology Sweden.
- Fossen, T. I., Pettersen, K. Y., & Galeazzi, R. (2015). Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827. <https://doi.org/10.1109/TCST.2014.2338354>
- Plummet, F., Petres, C., Rómero-Ramírez, M. A., Gas, B., Sio-Hoi, I. (2015). Towards an Autonomous Sailing Boat. *IEEE Journal of Oceanic Engineering*, pp. 397–407. DOI: 10.1109/JOE.2014.2321714.
- Tranzatto, Marco (2015a) Navigation and Control for an Autonomous Sailing Model Boat. Chapter 5: Tracking a Constant Heading pp. 25-29. Master Thesis, University of Pisa.
- Utkin, V. (2002) Sliding mode Control, p. 23. EOSS: UNESCO Encyclopedia of Life Support Systems.
- Velueta Guzmán, M. J., Rullán Lara, J. L., Ruiz Hernández, J. A., & Alazki, H. (2018).Robust control for the dynamics of an unmanned surface vehicle under the perturbations (marine waves and currents). *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6. <https://doi.org/10.1109/ICEEE.2018.8533948>
- von Ellenrieder, K. D. (2021b). Control of Marine Vehicles. *Sliding Mode Control*, pp. 489–526. Cham: Springer International Publishing.



# References

- Alves, C., Aguiar, P., Sebatiao, L. (2010). Sailboat: Autonomous Marine Robot of Eolic Propulsion . *Instituto Superior Técnico – Institute for Systems and Robotics (IST-ISR) Lisbon 1049-001, Portugal.*
- Giraldo, Nelson(2023). Sailboat navigation control system based on spiking neural networks. Master thesis, Universidad de Antioquia.
- Chacon, Esteban (2021). Autonomous Sailboat Prototype Sensors and Electronics Implementation with Machine Learning for Navigation, Masther's thesis, Universidad Politécnica de Catalunya.
- Dhomé, Ulyssen (2021). Further development and performance evaluation of the autonomous sailing boat Maribot Vane, Degre Project in Naval Architecture, KTH Royal Institute of Technology Sweden.
- Fossen, T. I., Pettersen, K. Y., & Galeazzi, R. (2015). Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827. <https://doi.org/10.1109/TCST.2014.2338354>
- Plummet, F., Petres, C., Rómero-Ramírez, M. A., Gas, B., Sio-Hoi, I. (2015). Towards an Autonomous Sailing Boat. *IEEE Journal of Oceanic Engineering*, pp. 397–407. DOI: 10.1109/JOE.2014.2321714.
- Tranzatto, Marco (2015a) Navigation and Control for an Autonomous Sailing Model Boat. Chapter 5: Tracking a Constant Heading pp. 25-29. Master Thesis, University of Pisa.
- Utkin, V. (2002) Sliding mode Control, p. 23. EOSS: UNESCO Encyclopedia of Life Support Systems.
- Velueta Guzmán, M. J., Rullán Lara, J. L., Ruiz Hernández, J. A., & Alazki, H. (2018).Robust control for the dynamics of an unmanned surface vehicle under the perturbations (marine waves and currents). *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6. <https://doi.org/10.1109/ICEEE.2018.8533948>
- von Ellenrieder, K. D. (2021b). Control of Marine Vehicles. *Sliding Mode Control*, pp. 489–526. Cham: Springer International Publishing.



# References

- Alves, C., Aguiar, P., Sebatiao, L. (2010). Sailboat: Autonomous Marine Robot of Eolic Propulsion . *Instituto Superior Técnico – Institute for Systems and Robotics (IST-ISR) Lisbon 1049-001, Portugal.*
- Giraldo, Nelson(2023). Sailboat navigation control system based on spiking neural networks. Master thesis, Universidad de Antioquia.
- Chacon, Esteban (2021). Autonomous Sailboat Prototype Sensors and Electronics Implementation with Machine Learning for Navigation, Masther's thesis, Universidad Politécnica de Catalunya.
- Dhomé, Ulyssen (2021). Further development and performance evaluation of the autonomous sailing boat Maribot Vane, Degre Project in Naval Architecture, KTH Royal Institute of Technology Sweden.
- Fossen, T. I., Pettersen, K. Y., & Galeazzi, R. (2015). Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827. <https://doi.org/10.1109/TCST.2014.2338354>
- Plummet, F., Petres, C., Rómero-Ramírez, M. A., Gas, B., Sio-Hoi, I. (2015). Towards an Autonomous Sailing Boat. *IEEE Journal of Oceanic Engineering*, pp. 397–407. DOI: 10.1109/JOE.2014.2321714.
- Tranzatto, Marco (2015a) Navigation and Control for an Autonomous Sailing Model Boat. Chapter 5: Tracking a Constant Heading pp. 25-29. Master Thesis, University of Pisa.
- Utkin, V. (2002) Sliding mode Control, p. 23. EOSS: UNESCO Encyclopedia of Life Support Systems.
- Velueta Guzmán, M. J., Rullán Lara, J. L., Ruiz Hernández, J. A., & Alazki, H. (2018).Robust control for the dynamics of an unmanned surface vehicle under the perturbations (marine waves and currents). *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6. <https://doi.org/10.1109/ICEEE.2018.8533948>
- von Ellenrieder, K. D. (2021b). Control of Marine Vehicles. *Sliding Mode Control*, pp. 489–526. Cham: Springer International Publishing.



# References

- Alves, C., Aguiar, P., Sebatiao, L. (2010). Sailboat: Autonomous Marine Robot of Eolic Propulsion . *Instituto Superior Técnico – Institute for Systems and Robotics (IST-ISR) Lisbon 1049-001, Portugal.*
- Giraldo, Nelson(2023). Sailboat navigation control system based on spiking neural networks. Master thesis, Universidad de Antioquia.
- Chacon, Esteban (2021). Autonomous Sailboat Prototype Sensors and Electronics Implementation with Machine Learning for Navigation, Masther's thesis, Universidad Politécnica de Catalunya.
- Dhomé, Ulyssen (2021). Further development and performance evaluation of the autonomous sailing boat Maribot Vane, Degre Project in Naval Architecture, KTH Royal Institute of Technology Sweden.
- Fossen, T. I., Pettersen, K. Y., & Galeazzi, R. (2015). Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827. <https://doi.org/10.1109/TCST.2014.2338354>
- Plummet, F., Petres, C., Rómero-Ramírez, M. A., Gas, B., Sio-Hoi, I. (2015). Towards an Autonomous Sailing Boat. *IEEE Journal of Oceanic Engineering*, pp. 397–407. DOI: 10.1109/JOE.2014.2321714.
- Tranzatto, Marco (2015a) Navigation and Control for an Autonomous Sailing Model Boat. Chapter 5: Tracking a Constant Heading pp. 25-29. Master Thesis, University of Pisa.
- Utkin, V. (2002) Sliding mode Control, p. 23. EOSS: UNESCO Encyclopedia of Life Support Systems.
- Velueta Guzmán, M. J., Rullán Lara, J. L., Ruiz Hernández, J. A., & Alazki, H. (2018).Robust control for the dynamics of an unmanned surface vehicle under the perturbations (marine waves and currents). *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6. <https://doi.org/10.1109/ICEEE.2018.8533948>
- von Ellenrieder, K. D. (2021b). Control of Marine Vehicles. *Sliding Mode Control*, pp. 489–526. Cham: Springer International Publishing.



# References

- Alves, C., Aguiar, P., Sebatiao, L. (2010). Sailboat: Autonomous Marine Robot of Eolic Propulsion . *Instituto Superior Técnico – Institute for Systems and Robotics (IST-ISR) Lisbon 1049-001, Portugal.*
- Giraldo, Nelson(2023). Sailboat navigation control system based on spiking neural networks. Master thesis, Universidad de Antioquia.
- Chacon, Esteban (2021). Autonomous Sailboat Prototype Sensors and Electronics Implementation with Machine Learning for Navigation, Masther's thesis, Universidad Politécnica de Catalunya.
- Dhomé, Ulyssen (2021). Further development and performance evaluation of the autonomous sailing boat Maribot Vane, Degre Project in Naval Architecture, KTH Royal Institute of Technology Sweden.
- Fossen, T. I., Pettersen, K. Y., & Galeazzi, R. (2015). Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827. <https://doi.org/10.1109/TCST.2014.2338354>
- Plummet, F., Petres, C., Rómero-Ramírez, M. A., Gas, B., Sio-Hoi, I. (2015). Towards an Autonomous Sailing Boat. *IEEE Journal of Oceanic Engineering*, pp. 397–407. DOI: 10.1109/JOE.2014.2321714.
- Tranzatto, Marco (2015a) Navigation and Control for an Autonomous Sailing Model Boat. Chapter 5: Tracking a Constant Heading pp. 25-29. Master Thesis, University of Pisa.
- Utkin, V. (2002) Sliding mode Control, p. 23. EOSS: UNESCO Encyclopedia of Life Support Systems.
- Velueta Guzmán, M. J., Rullán Lara, J. L., Ruiz Hernández, J. A., & Alazki, H. (2018).Robust control for the dynamics of an unmanned surface vehicle under the perturbations (marine waves and currents). *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6. <https://doi.org/10.1109/ICEEE.2018.8533948>
- von Ellenrieder, K. D. (2021b). Control of Marine Vehicles. *Sliding Mode Control*, pp. 489–526. Cham: Springer International Publishing.



# References

- Alves, C., Aguiar, P., Sebatiao, L. (2010). Sailboat: Autonomous Marine Robot of Eolic Propulsion . *Instituto Superior Técnico – Institute for Systems and Robotics (IST-ISR) Lisbon 1049-001, Portugal.*
- Giraldo, Nelson(2023). Sailboat navigation control system based on spiking neural networks. Master thesis, Universidad de Antioquia.
- Chacon, Esteban (2021). Autonomous Sailboat Prototype Sensors and Electronics Implementation with Machine Learning for Navigation, Masther's thesis, Universidad Politécnica de Catalunya.
- Dhomé, Ulyssen (2021). Further development and performance evaluation of the autonomous sailing boat Maribot Vane, Degre Project in Naval Architecture, KTH Royal Institute of Technology Sweden.
- Fossen, T. I., Pettersen, K. Y., & Galeazzi, R. (2015). Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827. <https://doi.org/10.1109/TCST.2014.2338354>
- Plummet, F., Petres, C., Rómero-Ramírez, M. A., Gas, B., Sio-Hoi, I. (2015). Towards an Autonomous Sailing Boat. *IEEE Journal of Oceanic Engineering*, pp. 397–407. DOI: 10.1109/JOE.2014.2321714.
- Tranzatto, Marco (2015a) Navigation and Control for an Autonomous Sailing Model Boat. Chapter 5: Tracking a Constant Heading pp. 25-29. Master Thesis, University of Pisa.
- Utkin, V. (2002) Sliding mode Control, p. 23. EOSS: UNESCO Encyclopedia of Life Support Systems.
- Velueta Guzmán, M. J., Rullán Lara, J. L., Ruiz Hernández, J. A., & Alazki, H. (2018).Robust control for the dynamics of an unmanned surface vehicle under the perturbations (marine waves and currents). *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6. <https://doi.org/10.1109/ICEEE.2018.8533948>
- von Ellenrieder, K. D. (2021b). Control of Marine Vehicles. *Sliding Mode Control*, pp. 489–526. Cham: Springer International Publishing.



# References

- Alves, C., Aguiar, P., Sebatiao, L. (2010). Sailboat: Autonomous Marine Robot of Eolic Propulsion . *Instituto Superior Técnico – Institute for Systems and Robotics (IST-ISR) Lisbon 1049-001, Portugal.*
- Giraldo, Nelson(2023). Sailboat navigation control system based on spiking neural networks. Master thesis, Universidad de Antioquia.
- Chacon, Esteban (2021). Autonomous Sailboat Prototype Sensors and Electronics Implementation with Machine Learning for Navigation, Masther's thesis, Universidad Politécnica de Catalunya.
- Dhomé, Ulyssen (2021). Further development and performance evaluation of the autonomous sailing boat Maribot Vane, Degre Project in Naval Architecture, KTH Royal Institute of Technology Sweden.
- Fossen, T. I., Pettersen, K. Y., & Galeazzi, R. (2015). Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827. <https://doi.org/10.1109/TCST.2014.2338354>
- Plummet, F., Petres, C., Rómero-Ramírez, M. A., Gas, B., Sio-Hoi, I. (2015). Towards an Autonomous Sailing Boat. *IEEE Journal of Oceanic Engineering*, pp. 397–407. DOI: 10.1109/JOE.2014.2321714.
- Tranzatto, Marco (2015a) Navigation and Control for an Autonomous Sailing Model Boat. Chapter 5: Tracking a Constant Heading pp. 25-29. Master Thesis, University of Pisa.
- Utkin, V. (2002) Sliding mode Control, p. 23. EOSS: UNESCO Encyclopedia of Life Support Systems.
- Velueta Guzmán, M. J., Rullán Lara, J. L., Ruiz Hernández, J. A., & Alazki, H. (2018).Robust control for the dynamics of an unmanned surface vehicle under the perturbations (marine waves and currents). *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6. <https://doi.org/10.1109/ICEEE.2018.8533948>
- von Ellenrieder, K. D. (2021b). Control of Marine Vehicles. *Sliding Mode Control*, pp. 489–526. Cham: Springer International Publishing.



# References

- Alves, C., Aguiar, P., Sebatiao, L. (2010). Sailboat: Autonomous Marine Robot of Eolic Propulsion . *Instituto Superior Técnico – Institute for Systems and Robotics (IST-ISR) Lisbon 1049-001, Portugal.*
- Giraldo, Nelson(2023). Sailboat navigation control system based on spiking neural networks. Master thesis, Universidad de Antioquia.
- Chacon, Esteban (2021). Autonomous Sailboat Prototype Sensors and Electronics Implementation with Machine Learning for Navigation, Masther's thesis, Universidad Politécnica de Catalunya.
- Dhomé, Ulyssen (2021). Further development and performance evaluation of the autonomous sailing boat Maribot Vane, Degre Project in Naval Architecture, KTH Royal Institute of Technology Sweden.
- Fossen, T. I., Pettersen, K. Y., & Galeazzi, R. (2015). Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827. <https://doi.org/10.1109/TCST.2014.2338354>
- Plummet, F., Petres, C., Rómero-Ramírez, M. A., Gas, B., Sio-Hoi, I. (2015). Towards an Autonomous Sailing Boat. *IEEE Journal of Oceanic Engineering*, pp. 397–407. DOI: 10.1109/JOE.2014.2321714.
- Tranzatto, Marco (2015a) Navigation and Control for an Autonomous Sailing Model Boat. Chapter 5: Tracking a Constant Heading pp. 25-29. Master Thesis, University of Pisa.
- Utkin, V. (2002) Sliding mode Control, p. 23. EOSS: UNESCO Encyclopedia of Life Support Systems.
- Velueta Guzmán, M. J., Rullán Lara, J. L., Ruiz Hernández, J. A., & Alazki, H. (2018).Robust control for the dynamics of an unmanned surface vehicle under the perturbations (marine waves and currents). *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6. <https://doi.org/10.1109/ICEEE.2018.8533948>
- von Ellenrieder, K. D. (2021b). Control of Marine Vehicles. *Sliding Mode Control*, pp. 489–526. Cham: Springer International Publishing.

# References

- Alves, C., Aguiar, P., Sebatiao, L. (2010). Sailboat: Autonomous Marine Robot of Eolic Propulsion . *Instituto Superior Técnico – Institute for Systems and Robotics (IST-ISR) Lisbon 1049-001, Portugal.*
- Giraldo, Nelson(2023). Sailboat navigation control system based on spiking neural networks. Master thesis, Universidad de Antioquia.
- Chacon, Esteban (2021). Autonomous Sailboat Prototype Sensors and Electronics Implementation with Machine Learning for Navigation, Masther's thesis, Universidad Politécnica de Catalunya.
- Dhomé, Ulyssen (2021). Further development and performance evaluation of the autonomous sailing boat Maribot Vane, Degre Project in Naval Architecture, KTH Royal Institute of Technology Sweden.
- Fossen, T. I., Pettersen, K. Y., & Galeazzi, R. (2015). Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827. <https://doi.org/10.1109/TCST.2014.2338354>
- Plummet, F., Petres, C., Rómero-Ramírez, M. A., Gas, B., Sio-Hoi, I. (2015). Towards an Autonomous Sailing Boat. *IEEE Journal of Oceanic Engineering*, pp. 397–407. DOI: 10.1109/JOE.2014.2321714.
- Tranzatto, Marco (2015a) Navigation and Control for an Autonomous Sailing Model Boat. Chapter 5: Tracking a Constant Heading pp. 25-29. Master Thesis, University of Pisa.
- Utkin, V. (2002) Sliding mode Control, p. 23. EOSS: UNESCO Encyclopedia of Life Support Systems.
- Velueta Guzmán, M. J., Rullán Lara, J. L., Ruiz Hernández, J. A., & Alazki, H. (2018).Robust control for the dynamics of an unmanned surface vehicle under the perturbations (marine waves and currents). *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6. <https://doi.org/10.1109/ICEEE.2018.8533948>
- von Ellenrieder, K. D. (2021b). Control of Marine Vehicles. *Sliding Mode Control*, pp. 489–526. Cham: Springer International Publishing.

# Argumentosopcionales: Parámetros por defecto

Los parámetros que sólo pueden asignarse a través de su posición en la lista se denominan parámetros posicionales.

ejemplo:

```
def defaults(m, n, p=0, q=0) :  
    print(m, n, p, q)  
  
defaults(3, 5, q = 7)  
defaults(m=10, n=13)  
defaults(5, p=20, n=12)
```

# References

- Alves, C., Aguiar, P., Sebatiao, L. (2010). Sailboat: Autonomous Marine Robot of Eolic Propulsion . *Instituto Superior Técnico – Institute for Systems and Robotics (IST-ISR) Lisbon 1049-001, Portugal.*
- Giraldo, Nelson(2023). Sailboat navigation control system based on spiking neural networks. Master thesis, Universidad de Antioquia.
- Chacon, Esteban (2021). Autonomous Sailboat Prototype Sensors and Electronics Implementation with Machine Learning for Navigation, Masther's thesis, Universidad Politécnica de Catalunya.
- Dhomé, Ulyssen (2021). Further development and performance evaluation of the autonomous sailing boat Maribot Vane, Degre Project in Naval Architecture, KTH Royal Institute of Technology Sweden.
- Fossen, T. I., Pettersen, K. Y., & Galeazzi, R. (2015). Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827. <https://doi.org/10.1109/TCST.2014.2338354>
- Plummet, F., Petres, C., Rómero-Ramírez, M. A., Gas, B., Sio-Hoi, I. (2015). Towards an Autonomous Sailing Boat. *IEEE Journal of Oceanic Engineering*, pp. 397–407. DOI: 10.1109/JOE.2014.2321714.
- Tranzatto, Marco (2015a) Navigation and Control for an Autonomous Sailing Model Boat. Chapter 5: Tracking a Constant Heading pp. 25-29. Master Thesis, University of Pisa.
- Utkin, V. (2002) Sliding mode Control, p. 23. EOSS: UNESCO Encyclopedia of Life Support Systems.
- Velueta Guzmán, M. J., Rullán Lara, J. L., Ruiz Hernández, J. A., & Alazki, H. (2018).Robust control for the dynamics of an unmanned surface vehicle under the perturbations (marine waves and currents). *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6. <https://doi.org/10.1109/ICEEE.2018.8533948>
- von Ellenrieder, K. D. (2021b). Control of Marine Vehicles. *Sliding Mode Control*, pp. 489–526. Cham: Springer International Publishing.

# Argumentosopcionales: Parámetros por defecto

Los parámetros que sólo pueden asignarse a través de su posición en la lista se denominan parámetros posicionales.

ejemplo:

```
def defaults(m, n, p=0, q=0) :  
    print(m, n, p, q)  
  
defaults(3, 5, q = 7)  
defaults(m=10, n=13)  
defaults(5, p=20, n=12)
```

# References

- Alves, C., Aguiar, P., Sebatiao, L. (2010). Sailboat: Autonomous Marine Robot of Eolic Propulsion . *Instituto Superior Técnico – Institute for Systems and Robotics (IST-ISR) Lisbon 1049-001, Portugal.*
- Giraldo, Nelson(2023). Sailboat navigation control system based on spiking neural networks. Master thesis, Universidad de Antioquia.
- Chacon, Esteban (2021). Autonomous Sailboat Prototype Sensors and Electronics Implementation with Machine Learning for Navigation, Masther's thesis, Universidad Politécnica de Catalunya.
- Dhomé, Ulyssen (2021). Further development and performance evaluation of the autonomous sailing boat Maribot Vane, Degre Project in Naval Architecture, KTH Royal Institute of Technology Sweden.
- Fossen, T. I., Pettersen, K. Y., & Galeazzi, R. (2015). Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827. <https://doi.org/10.1109/TCST.2014.2338354>
- Plummet, F., Petres, C., Rómero-Ramírez, M. A., Gas, B., Sio-Hoi, I. (2015). Towards an Autonomous Sailing Boat. *IEEE Journal of Oceanic Engineering*, pp. 397–407. DOI: 10.1109/JOE.2014.2321714.
- Tranzatto, Marco (2015a) Navigation and Control for an Autonomous Sailing Model Boat. Chapter 5: Tracking a Constant Heading pp. 25-29. Master Thesis, University of Pisa.
- Utkin, V. (2002) Sliding mode Control, p. 23. EOSS: UNESCO Encyclopedia of Life Support Systems.
- Velueta Guzmán, M. J., Rullán Lara, J. L., Ruiz Hernández, J. A., & Alazki, H. (2018).Robust control for the dynamics of an unmanned surface vehicle under the perturbations (marine waves and currents). *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6. <https://doi.org/10.1109/ICEEE.2018.8533948>
- von Ellenrieder, K. D. (2021b). Control of Marine Vehicles. *Sliding Mode Control*, pp. 489–526. Cham: Springer International Publishing.

# Argumentosopcionales: Parámetros por defecto

Los parámetros que sólo pueden asignarse a través de su posición en la lista se denominan parámetros posicionales.

ejemplo:

```
def defaults(m, n, p=0, q=0) :  
    print(m, n, p, q)  
  
defaults(3, 5, q = 7)  
defaults(m=10, n=13)  
defaults(5, p=20, n=12)
```

# References

- Alves, C., Aguiar, P., Sebatiao, L. (2010). Sailboat: Autonomous Marine Robot of Eolic Propulsion . *Instituto Superior Técnico – Institute for Systems and Robotics (IST-ISR) Lisbon 1049-001, Portugal.*
- Giraldo, Nelson(2023). Sailboat navigation control system based on spiking neural networks. Master thesis, Universidad de Antioquia.
- Chacon, Esteban (2021). Autonomous Sailboat Prototype Sensors and Electronics Implementation with Machine Learning for Navigation, Masther's thesis, Universidad Politécnica de Catalunya.
- Dhomé, Ulyssen (2021). Further development and performance evaluation of the autonomous sailing boat Maribot Vane, Degre Project in Naval Architecture, KTH Royal Institute of Technology Sweden.
- Fossen, T. I., Pettersen, K. Y., & Galeazzi, R. (2015). Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827. <https://doi.org/10.1109/TCST.2014.2338354>
- Plummet, F., Petres, C., Rómero-Ramírez, M. A., Gas, B., Sio-Hoi, I. (2015). Towards an Autonomous Sailing Boat. *IEEE Journal of Oceanic Engineering*, pp. 397–407. DOI: 10.1109/JOE.2014.2321714.
- Tranzatto, Marco (2015a) Navigation and Control for an Autonomous Sailing Model Boat. Chapter 5: Tracking a Constant Heading pp. 25-29. Master Thesis, University of Pisa.
- Utkin, V. (2002) Sliding mode Control, p. 23. EOSS: UNESCO Encyclopedia of Life Support Systems.
- Velueta Guzmán, M. J., Rullán Lara, J. L., Ruiz Hernández, J. A., & Alazki, H. (2018).Robust control for the dynamics of an unmanned surface vehicle under the perturbations (marine waves and currents). *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6. <https://doi.org/10.1109/ICEEE.2018.8533948>
- von Ellenrieder, K. D. (2021b). Control of Marine Vehicles. *Sliding Mode Control*, pp. 489–526. Cham: Springer International Publishing.

# Argumentosopcionales: Parámetros por defecto

Los parámetros que sólo pueden asignarse a través de su posición en la lista se denominan parámetros posicionales.

ejemplo:

```
def defaults(m, n, p=0, q=0) :  
    print(m, n, p, q)  
  
defaults(3, 5, q = 7)  
defaults(m=10, n=13)  
defaults(5, p=20, n=12)
```

# References

- Alves, C., Aguiar, P., Sebatiao, L. (2010). Sailboat: Autonomous Marine Robot of Eolic Propulsion . *Instituto Superior Técnico – Institute for Systems and Robotics (IST-ISR) Lisbon 1049-001, Portugal.*
- Giraldo, Nelson(2023). Sailboat navigation control system based on spiking neural networks. Master thesis, Universidad de Antioquia.
- Chacon, Esteban (2021). Autonomous Sailboat Prototype Sensors and Electronics Implementation with Machine Learning for Navigation, Masther's thesis, Universidad Politécnica de Catalunya.
- Dhomé, Ulyssen (2021). Further development and performance evaluation of the autonomous sailing boat Maribot Vane, Degre Project in Naval Architecture, KTH Royal Institute of Technology Sweden.
- Fossen, T. I., Pettersen, K. Y., & Galeazzi, R. (2015). Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827. <https://doi.org/10.1109/TCST.2014.2338354>
- Plummet, F., Petres, C., Rómero-Ramírez, M. A., Gas, B., Sio-Hoi, I. (2015). Towards an Autonomous Sailing Boat. *IEEE Journal of Oceanic Engineering*, pp. 397–407. DOI: 10.1109/JOE.2014.2321714.
- Tranzatto, Marco (2015a) Navigation and Control for an Autonomous Sailing Model Boat. Chapter 5: Tracking a Constant Heading pp. 25-29. Master Thesis, University of Pisa.
- Utkin, V. (2002) Sliding mode Control, p. 23. EOSS: UNESCO Encyclopedia of Life Support Systems.
- Velueta Guzmán, M. J., Rullán Lara, J. L., Ruiz Hernández, J. A., & Alazki, H. (2018).Robust control for the dynamics of an unmanned surface vehicle under the perturbations (marine waves and currents). *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6. <https://doi.org/10.1109/ICEEE.2018.8533948>
- von Ellenrieder, K. D. (2021b). Control of Marine Vehicles. *Sliding Mode Control*, pp. 489–526. Cham: Springer International Publishing.

# Argumentosopcionales: Parámetros por defecto

Los parámetros que sólo pueden asignarse a través de su posición en la lista se denominan parámetros posicionales.

ejemplo:

```
def defaults(m, n, p=0, q=0) :  
    print(m, n, p, q)  
  
defaults(3, 5, q = 7)  
defaults(m=10, n=13)  
defaults(5, p=20, n=12)
```

# References

- Alves, C., Aguiar, P., Sebatiao, L. (2010). Sailboat: Autonomous Marine Robot of Eolic Propulsion . *Instituto Superior Técnico – Institute for Systems and Robotics (IST-ISR) Lisbon 1049-001, Portugal.*
- Giraldo, Nelson(2023). Sailboat navigation control system based on spiking neural networks. Master thesis, Universidad de Antioquia.
- Chacon, Esteban (2021). Autonomous Sailboat Prototype Sensors and Electronics Implementation with Machine Learning for Navigation, Masther's thesis, Universidad Politécnica de Catalunya.
- Dhomé, Ulyssen (2021). Further development and performance evaluation of the autonomous sailing boat Maribot Vane, Degre Project in Naval Architecture, KTH Royal Institute of Technology Sweden.
- Fossen, T. I., Pettersen, K. Y., & Galeazzi, R. (2015). Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827. <https://doi.org/10.1109/TCST.2014.2338354>
- Plummet, F., Petres, C., Rómero-Ramírez, M. A., Gas, B., Sio-Hoi, I. (2015). Towards an Autonomous Sailing Boat. *IEEE Journal of Oceanic Engineering*, pp. 397–407. DOI: 10.1109/JOE.2014.2321714.
- Tranzatto, Marco (2015a) Navigation and Control for an Autonomous Sailing Model Boat. Chapter 5: Tracking a Constant Heading pp. 25-29. Master Thesis, University of Pisa.
- Utkin, V. (2002) Sliding mode Control, p. 23. EOSS: UNESCO Encyclopedia of Life Support Systems.
- Velueta Guzmán, M. J., Rullán Lara, J. L., Ruiz Hernández, J. A., & Alazki, H. (2018).Robust control for the dynamics of an unmanned surface vehicle under the perturbations (marine waves and currents). *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6. <https://doi.org/10.1109/ICEEE.2018.8533948>
- von Ellenrieder, K. D. (2021b). Control of Marine Vehicles. *Sliding Mode Control*, pp. 489–526. Cham: Springer International Publishing.

# Argumentosopcionales: Parámetros por defecto

Los parámetros que sólo pueden asignarse a través de su posición en la lista se denominan parámetros posicionales.

ejemplo:

```
def defaults(m, n, p=0, q=0) :  
    print(m, n, p, q)  
  
defaults(3, 5, q = 7)  
defaults(m=10, n=13)  
defaults(5, p=20, n=12)
```