

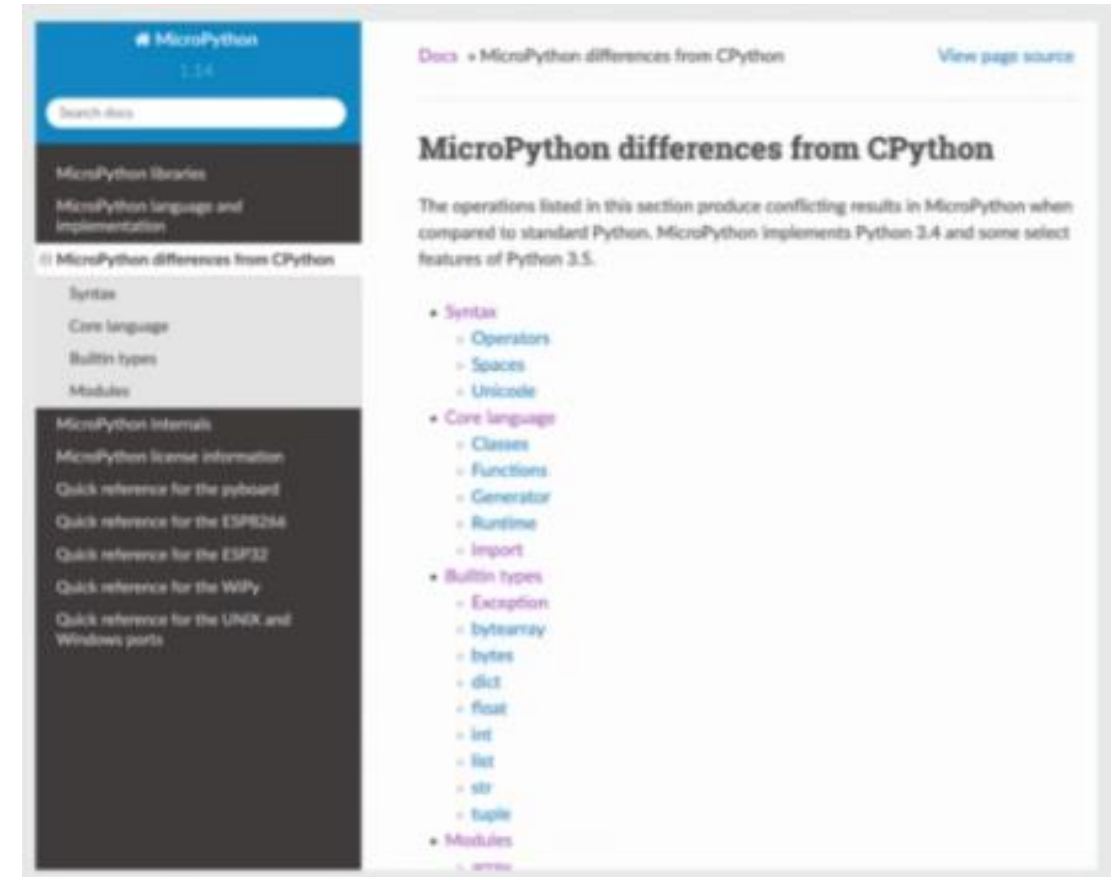
Micropython

Sergio Fco. Hernández Machuca

Micropython contra CPython

- Sintaxis
- Clase
- Funciones
- Tipos
- Módulos
- Detalles en:

<https://docs.micropython.org/en/latest/genrst/index.html>



Sintaxis: Espacios

- uPython requiere espacios entre literales, números y palabras reservada; CPython no.
- CPython:
 - `>>> 1and 0` # Sí funciona
 - `>>> 0`
- uPython:
 - `>>> 1and 0` # NO funciona, debe usarse '1 and 0'
 - Traceback (most recent call last):
 - File "<stdin>", line 1
 - SyntaxError: invalid syntax for integer with base 10
 - `>>>`

Parte central del lenguaje: Funciones

- Mensajes de errores para métodos pueden mostrar un número erróneo en el conteo de argumentos.
- CPython cuenta a 'self' como parte de los argumentos.
- uPython no toma en cuenta a 'self' como argumento.

CPython

```
>>> a = Calculator(1) # Requires 2 arguments
Traceback (most recent call last):
File "<pyshell>", line 1, in <module>
TypeError: __init__() missing 1 required
positional argument: 'num2'
```

uPython

```
>>> a = Calculator(1) # Requires 2 arguments
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
TypeError: function takes 3 positional
arguments but 2 were given
```

Tipos: Formato de 'float'

- Cuando se imprimen números con puntos flotantes, el resultado es diferente entre CPython y uPython.

CPython

```
>>> print("%.1g" % -9.9)
-1e+01
>>> print("%.2g" % -9.9)
-9.9
```

uPython

```
>>> print("%.1g" % -9.9)
-10
>>> print("%.2g" % -9.9)
-9.9
```

Tipos: 'str'

- Los índices de inicio / fin, tales como, `str.endswith(s, start)` no está implementado en uPython.

CPython

```
>>> "testing 123".endswith("23")
True
>>> "testing 123".endswith("23",3,5)
False
```

uPython

```
>>> "testing 123".endswith("23")
True
>>> "testing 123".endswith("23",3,5)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: function expected at most 3 arguments, got 4
```

Módulos: 'json'

- El módulo 'json' no indica excepciones cuando un objeto se serializable.

Sample code:

```
import json

a = bytes(x for x in range(256))
try:
    z = json.dumps(a)
    x = json.loads(z)
    print("Should not get here")
except TypeError:
    print("TypeError")
```

CPy output:

TypeError

uPy output:

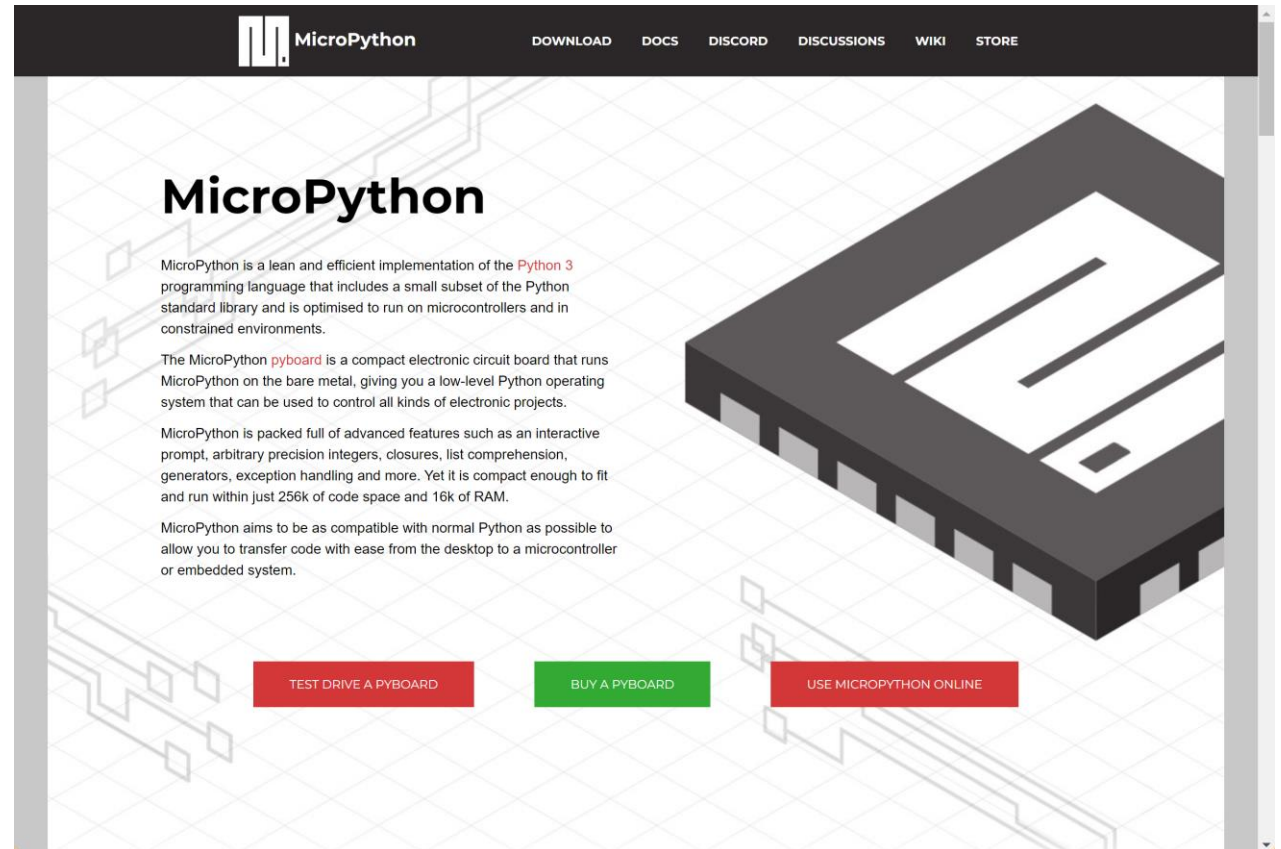
Should **not** get here

Recursos de uPython

- Permite trasladar a dispositivos (uControladores, en particular) las facilidades de programarles con Python, en una versión adecuada.
- El usar Python en el diseño de sistemas electrónicos permite un ciclo de trabajo más eficiente: edita – envía – enmienda – envía a diferencia de: edita – compila (zzz) – envía – enmienda – compila (zzz)
- Existe una comunidad de CPython que le ha posicionado como uno de los lenguajes de programación más usados.

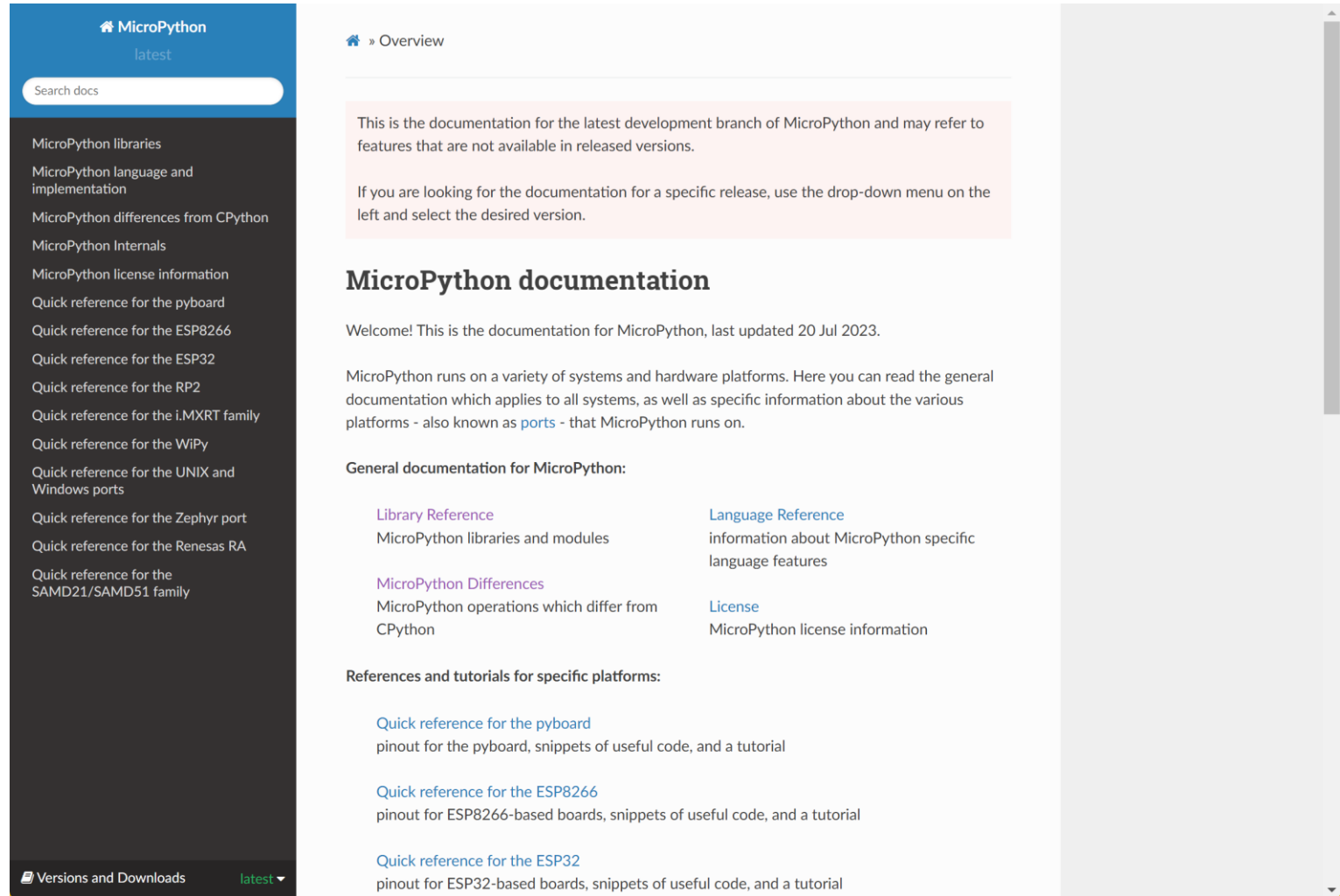
Micropython

- Sitio principal:
<https://micropython.org/>
- Es importante revisar los elementos del lenguaje, su fundación, enfoque de desarrollo, soporte a muchos elementos (uCs).



Micropython: Documentación

- Se encuentra en:
<https://docs.micropython.org/en/latest/>
- Contiene detalles acerca del lenguaje, implementación y librerías de manera general.



The screenshot displays the MicroPython documentation website. The left sidebar features a search bar and a list of navigation links including 'MicroPython libraries', 'MicroPython language and implementation', 'MicroPython differences from CPython', 'MicroPython Internals', 'MicroPython license information', and various 'Quick reference' links for different hardware platforms like pyboard, ESP8266, ESP32, RP2, i.MXRT, WiPy, and various ports. The main content area shows the 'Overview' page, which includes a disclaimer about the latest development branch, a welcome message dated 20 Jul 2023, and a table of general documentation links such as 'Library Reference', 'Language Reference', 'MicroPython Differences', and 'License'. At the bottom, there are links for 'References and tutorials for specific platforms' for the pyboard, ESP8266, and ESP32.

MicroPython
latest

Search docs

MicroPython libraries
MicroPython language and implementation
MicroPython differences from CPython
MicroPython Internals
MicroPython license information
Quick reference for the pyboard
Quick reference for the ESP8266
Quick reference for the ESP32
Quick reference for the RP2
Quick reference for the i.MXRT family
Quick reference for the WiPy
Quick reference for the UNIX and Windows ports
Quick reference for the Zephyr port
Quick reference for the Renesas RA
Quick reference for the SAMD21/SAMD51 family

Versions and Downloads latest ▼

» Overview

This is the documentation for the latest development branch of MicroPython and may refer to features that are not available in released versions.

If you are looking for the documentation for a specific release, use the drop-down menu on the left and select the desired version.

MicroPython documentation

Welcome! This is the documentation for MicroPython, last updated 20 Jul 2023.

MicroPython runs on a variety of systems and hardware platforms. Here you can read the general documentation which applies to all systems, as well as specific information about the various platforms - also known as **ports** - that MicroPython runs on.

General documentation for MicroPython:

| | |
|--|---|
| Library Reference MicroPython libraries and modules | Language Reference information about MicroPython specific language features |
| MicroPython Differences MicroPython operations which differ from CPython | License MicroPython license information |

References and tutorials for specific platforms:

Quick reference for the pyboard
pinout for the pyboard, snippets of useful code, and a tutorial

Quick reference for the ESP8266
pinout for ESP8266-based boards, snippets of useful code, and a tutorial

Quick reference for the ESP32
pinout for ESP32-based boards, snippets of useful code, and a tutorial

Micropython: Referencias del Lenguaje

- Dado que uPython derive de CPython, se requerirá consultar la información de Python.
- Se encuentra en:
<https://docs.python.org/3/reference/index.html>

The screenshot shows the Python 3.11.4 Documentation page for 'The Python Language Reference'. The page has a header with 'Python » English » 3.11.4 » 3.11.4 Documentation » The Python Language Reference' and a search bar. On the left, there is a sidebar with 'Previous topic: 6. Editors and IDEs', 'Next topic: 1. Introduction', and 'This Page: Report a Bug, Show Source'. The main content area is titled 'The Python Language Reference' and contains a paragraph describing the manual's purpose. Below this is a table of contents with sections 1 through 5, each with sub-sections. The page is styled with a light gray sidebar and a white main content area.

Python » English » 3.11.4 » 3.11.4 Documentation » The Python Language Reference

Quick search Go | previous | next | modules | index

Previous topic
6. Editors and IDEs

Next topic
1. Introduction

This Page
Report a Bug
Show Source

The Python Language Reference

This reference manual describes the syntax and “core semantics” of the language. It is terse, but attempts to be exact and complete. The semantics of non-essential built-in object types and of the built-in functions and modules are described in [The Python Standard Library](#). For an informal introduction to the language, see [The Python Tutorial](#). For C or C++ programmers, two additional manuals exist: [Extending and Embedding the Python Interpreter](#) describes the high-level picture of how to write a Python extension module, and the [Python/C API Reference Manual](#) describes the interfaces available to C/C++ programmers in detail.

- 1. Introduction
 - 1.1. Alternate Implementations
 - 1.2. Notation
- 2. Lexical analysis
 - 2.1. Line structure
 - 2.2. Other tokens
 - 2.3. Identifiers and keywords
 - 2.4. Literals
 - 2.5. Operators
 - 2.6. Delimiters
- 3. Data model
 - 3.1. Objects, values and types
 - 3.2. The standard type hierarchy
 - 3.3. Special method names
 - 3.4. Coroutines
- 4. Execution model
 - 4.1. Structure of a program
 - 4.2. Naming and binding
 - 4.3. Exceptions
- 5. The import system
 - 5.1. `importlib`
 - 5.2. Packages
 - 5.3. Searching
 - 5.4. Loading
 - 5.5. The Path Based Finder

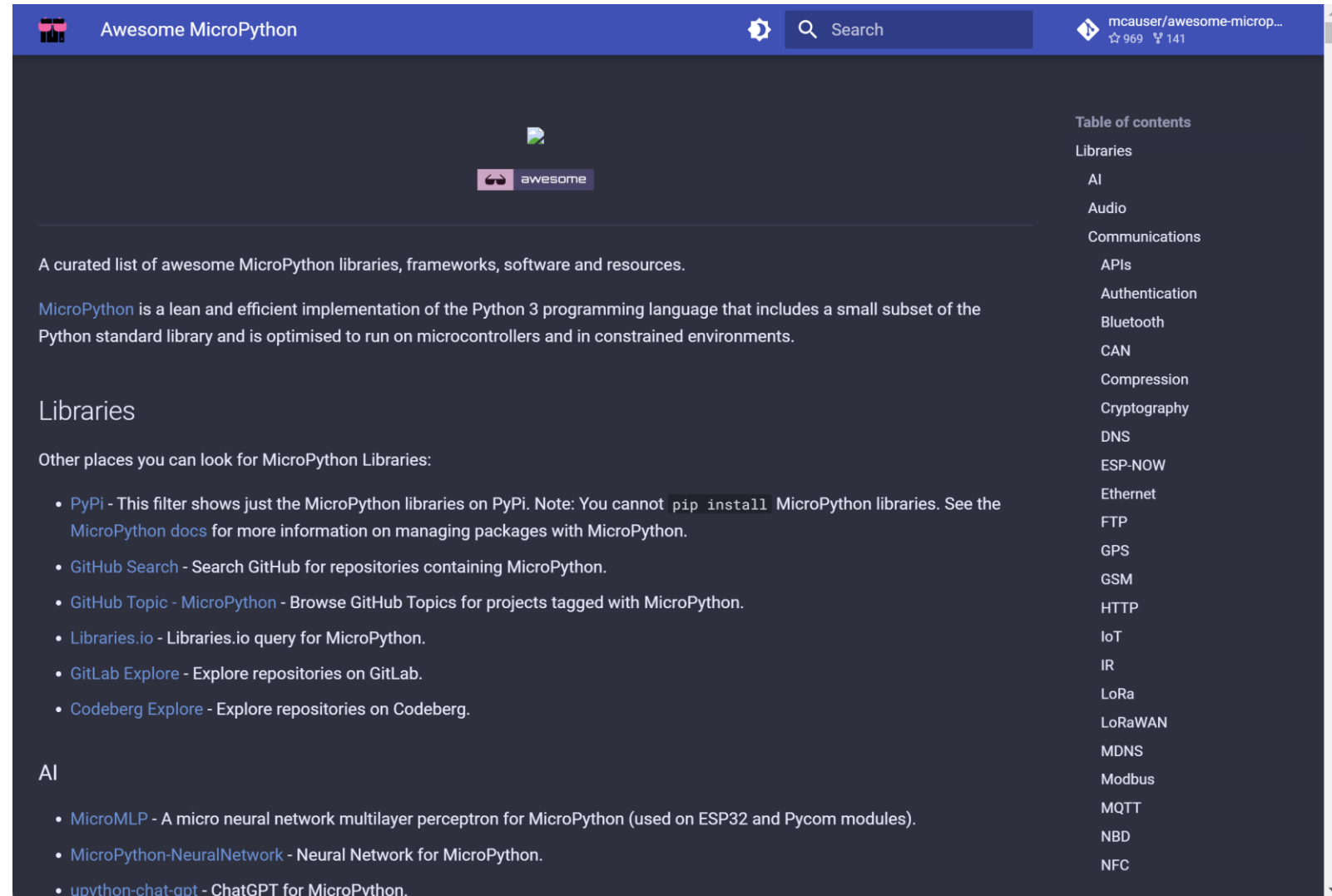
Micropython: Librerías

- Se pueden encontrar en Python Package Index (PyPi), es un repositorio donde se pueden encontrar paquetes (módulos) para uPython.
- Está en: <https://pypi.org/>

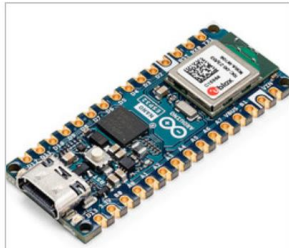


Micropython: Otras ayudas

- Un sitio interesante es: <https://awesome-micropython.com/>
- Contiene una lista organizada de recursos.
- Incluye muchas librerías.
- Organiza los recursos por áreas de aplicación.
- Es el principal sitio de consulta eficaz.



ESP32: Tarjetas principales



Arduino Nano ESP32
Arduino



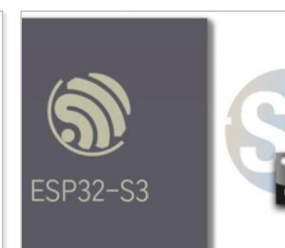
ESP32
Espressif



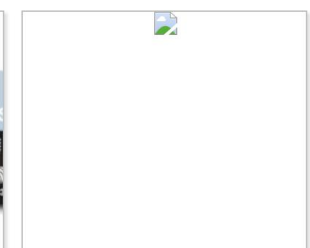
ESP32 D2WD
Espressif



ESP32-S2 WROVER
Espressif



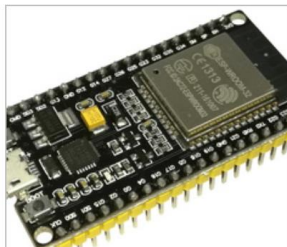
ESP32-S3
Espressif



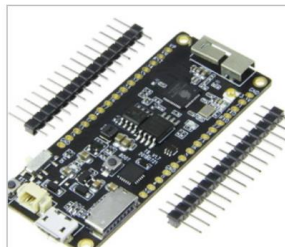
Generic ESP32-S3 (SPIRAM Octal)
Espressif



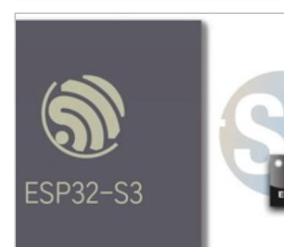
ESP32 Unicore
Espressif



ESP32 with OTA support
Espressif



ESP32 with SPIRAM
Espressif



Generic ESP32-S3 (SPIRAM)
Espressif



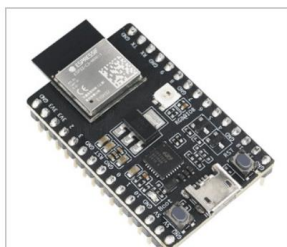
LILYGO TTGO LoRa32
LILYGO



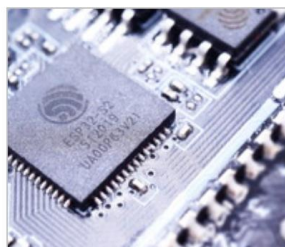
M5 Stack Atom
M5 Stack



ESP32-C3
Espressif



ESP32-C3 with USB
Espressif



ESP32-S2
Espressif



Olimex ESP32 POE
OLIMEX



SIL WESP32
Silicognition LLC



Feather S2
Unexpected Maker

ESP32: Tarjetas principales



Feather S2 Neo
Unexpected Maker



FeatherS3
Unexpected Maker



ProS3
Unexpected Maker



Tiny S2
Unexpected Maker



TinyPICO
Unexpected Maker



TinyS3
Unexpected Maker



C3 mini
Wemos



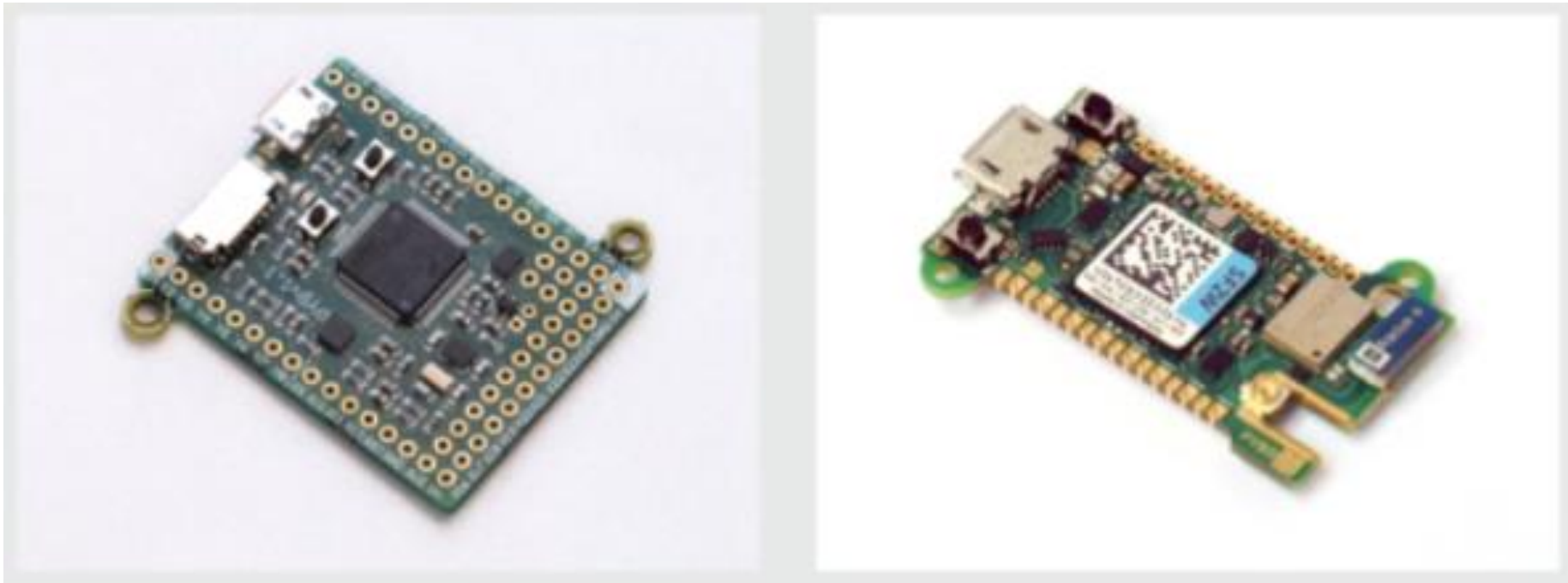
S2 mini
Wemos



S2 pico
Wemos

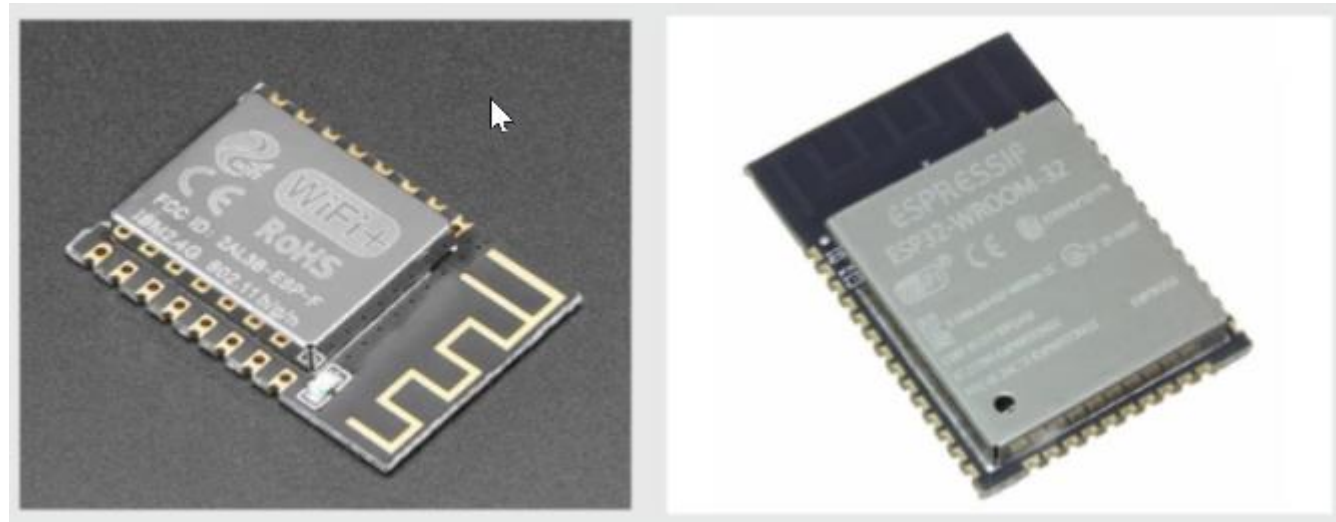
ESP32: Pyboard

- Tarjeta bandera, la que da origen al proyecto de uPython
- Serie D-series
- Los microcontroladores originales para uPython



ESP32: Basados en productos Espressif

- Crean el ESP8266
- Ahora ESP32
- Potente, bajo costo, buenas características



Otras tarjetas: RP2040

- Raspberry Pi PICO
- Feather 2040
- ItsBitsy 2040
- Tiny 2040
- Arduino 2040



Otras tarjetas: STM32

- STM32 Núcleo y Discovery
- Espruino Pico
- Muchas más

STM32

There is currently support for the following ST boards:

- B-L072Z-LRWAN1
- B-L475E-IOT01A
- NUCLEO-F091RC
- NUCLEO-F401RE

- STM32F4DISCOVERY (with STM32F407 MCU)
- STM32F769I-DISCO
- STM32F7DISCOVERY (with STM32F746 MCU)
- STM32L476G-DISCO
- STM32L496G-DISCO
- USB Dongle-WB55

The official reference hardware for MicroPython is the pyboard which contains an STM32F405 microcontroller.

<http://micropython.org/stm32/>

