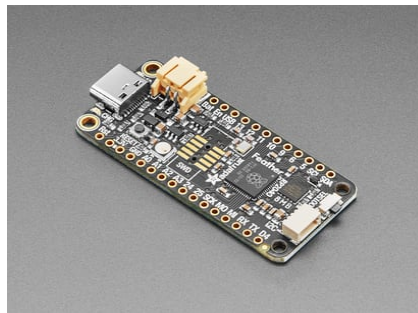


# Del 1 – Intro till CircuitPython

Grundläggande elektronik och programmering



1

Maskiner och datorer finns nästan överallt idag – allt från kylskåp till rymdsonder klassas som datorstyrda maskiner. I den här laborationen får du lära dig att kommunicera med en dator och få den att utföra uppgifter. Till det ska vi använda ett mikrokontrollerkort, i vårt fall en av typen *Adafruit Feather*, d.v.s. ett kretskort med en inbyggd mikrokontroller (processor), som är lättåtkomligt både för programmering och för att bygga elektronikprojekt. Feather kommunicerar med en dator via USB och kan både ta emot och läsa både digitala och analoga signaler via ett antal kopplingspunkter, så kallade *pins*.

## Material

- Adafruit Feather RP2040
- USB-sladd för att koppla Feather till datorn
- Dator

---

<sup>1</sup> SMART-1, By ESA, CC BY-SA 3.0-igo, <https://commons.wikimedia.org/w/index.php?curid=59274312>  
Kylskåp, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=539461>  
Feather RP2040, By Adafruit Industries, CC BY-SA 3.0,  
<https://cdn-shop.adafruit.com/970x728/5724-00.jpg>

**E-post:** cansat@au.se

**Telefon:** 070-000 90 56

**Senast uppdaterad:** 2024-11-19



**Astronomisk  
Ungdom**

# CircuitPython

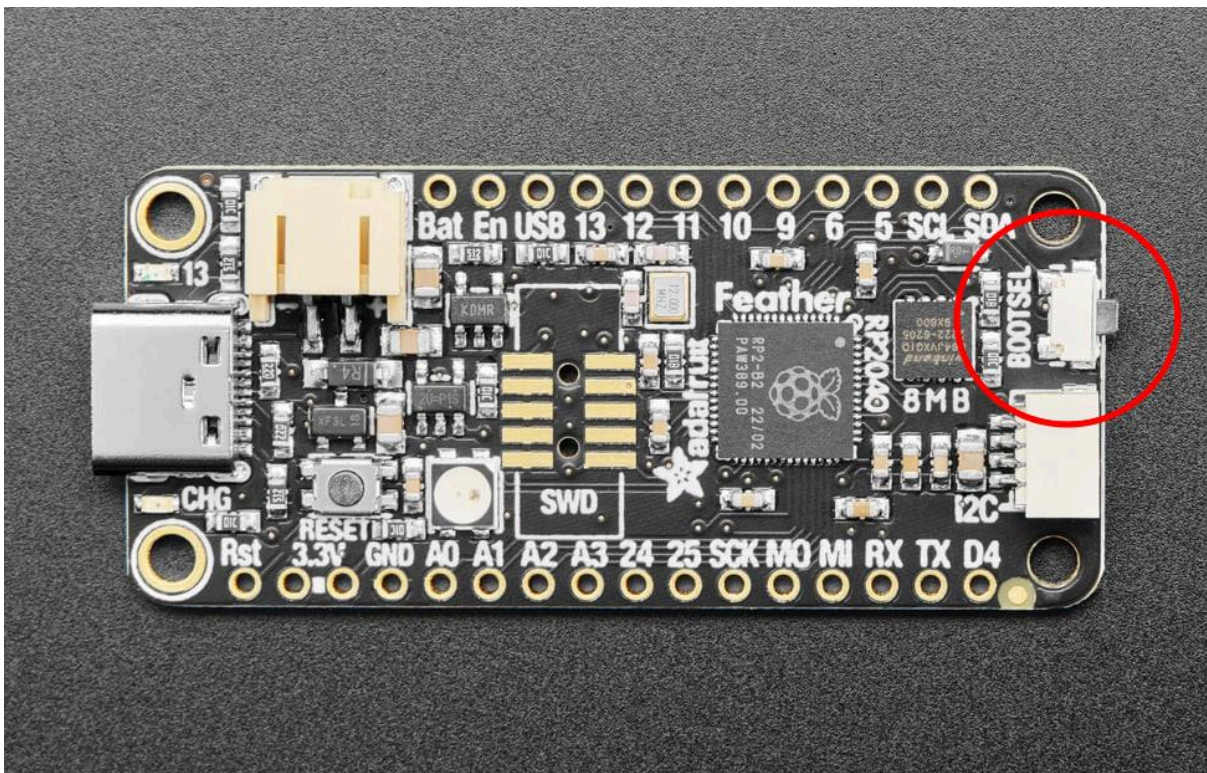
Det finns många olika programmeringsspråk som lämpar sig olika bra till olika ändamål. CircuitPython är vad som kommer att användas i dessa laborationer. Det är en variant av programmeringsspråket Python som är känt för sin användarvänlighet och smidighet. CircuitPython låter oss skriva Python-kod som körs på en mikrokontroller såsom en Feather.

## Installation

### Förberedelse:

Första gången man ska använda en mikrokontroller kan man behöva installera en "grundmjukvara" på dem som tar hand om att köra Python-koden som vi senare kommer att skriva. Ofta finns denna från början, men det är en bra vana att inför ett projekt installera om den så att man säkert har senaste versionen redo. För att ladda upp grundmjukvaran till en Feather RP2040 börjar man med att ladda ned mjukvaran från [https://circuitpython.org/board/adafruit\\_feather\\_rp2040/](https://circuitpython.org/board/adafruit_feather_rp2040/) i form av en .uf2-fil.

Håll sedan in BOOT SELECT-knappen medan man kopplar in Feathern i datorn, alternativt trycker på reset-knappen om den redan är inkopplad. Se bild nedan.



2

<sup>2</sup>Feather RP2040, By Adafruit Industries, CC BY-SA 3.0, <https://learn.adafruit.com/assets/116895>

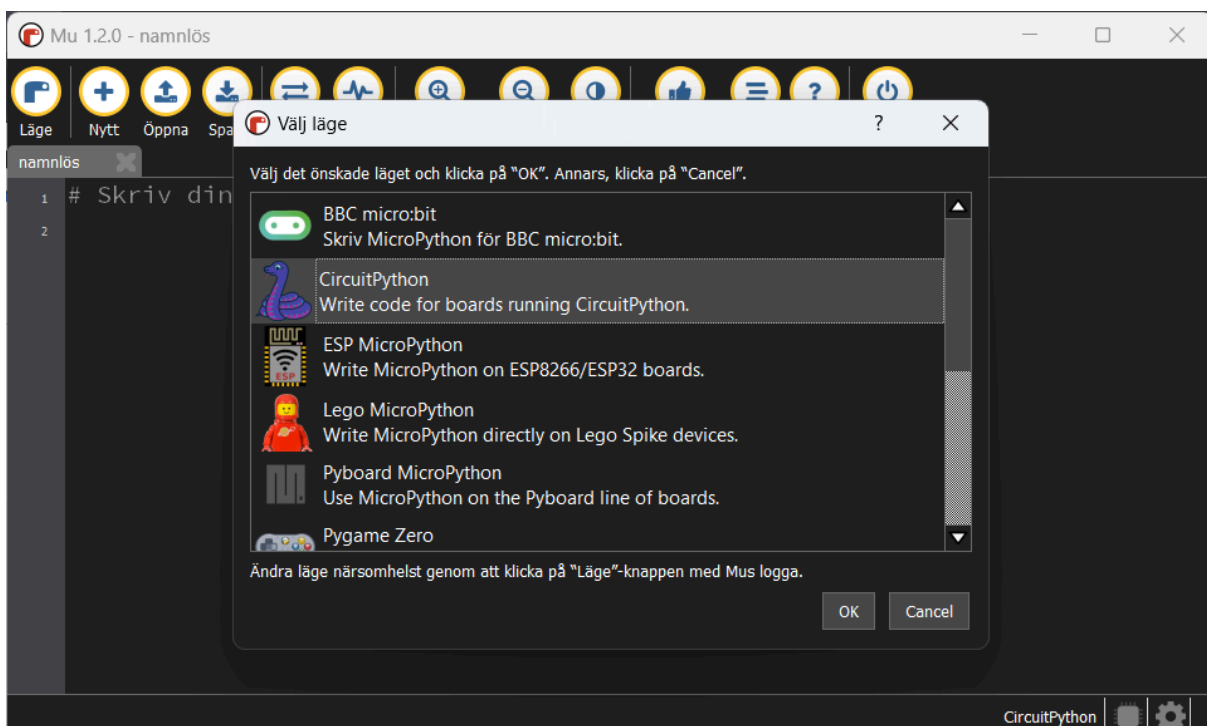
Feathern kommer att dyka upp i datorn som om det var ett USB-minne. Kopiera .uf2-filen hit genom att exempelvis dra och släppa.

Grundmjukvaran kommer nu att installeras på Feathern. När det är klart dyker Feathern upp som om den vore ett usb minne igen, fast nu med namnet CIRCUITPYTHON. Filerna som ligger på är dem som Feathern kommer köra. "code.py" är filen med den Python-koden du vill köra. Mappen med namn libs är där du senare kommer att installera kod för olika sensorer och liknande som andra har skapat.

### Programvara:

För att skriva kod och interagera med mikroprocessorn används en kodredigerare. Förslag på sådana är Mu-editor eller Thonny. Denna guide kommer att utgå ifrån Mu-editor men förfarandet är snarlikt i Thonny. Mu-editorn installeras lättast från <https://codewith.mu/>.

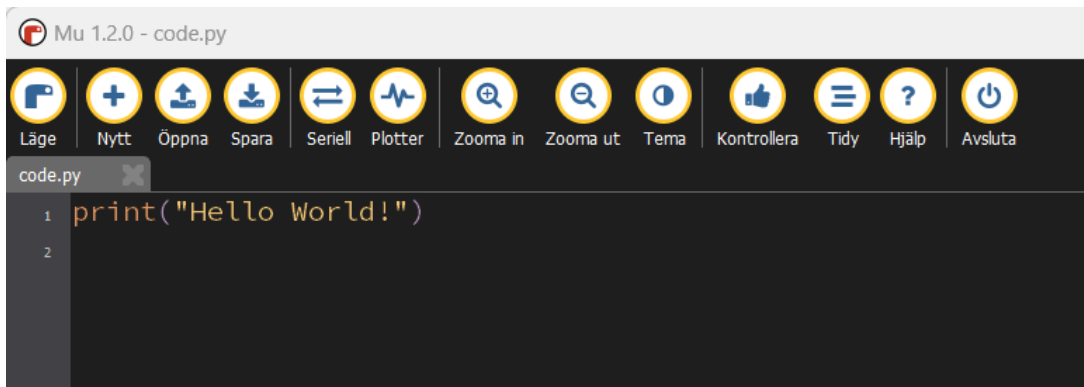
Första gången du startar Mu kommer du bli tillfrågad hur du vill använda programmet, välj då alternativet som heter CircuitPython.



Om Feathern är inkopplad i datorn ska den automatisk hittas och du får upp det Python-program som finns på den. Dyker det inte upp så dubbelkolla att Feathern är ansluten ordentligt, tryck sedan på Öppna (längst upp till vänster) för att sedan välja code.py från CIRCUITPYTHON enheten.

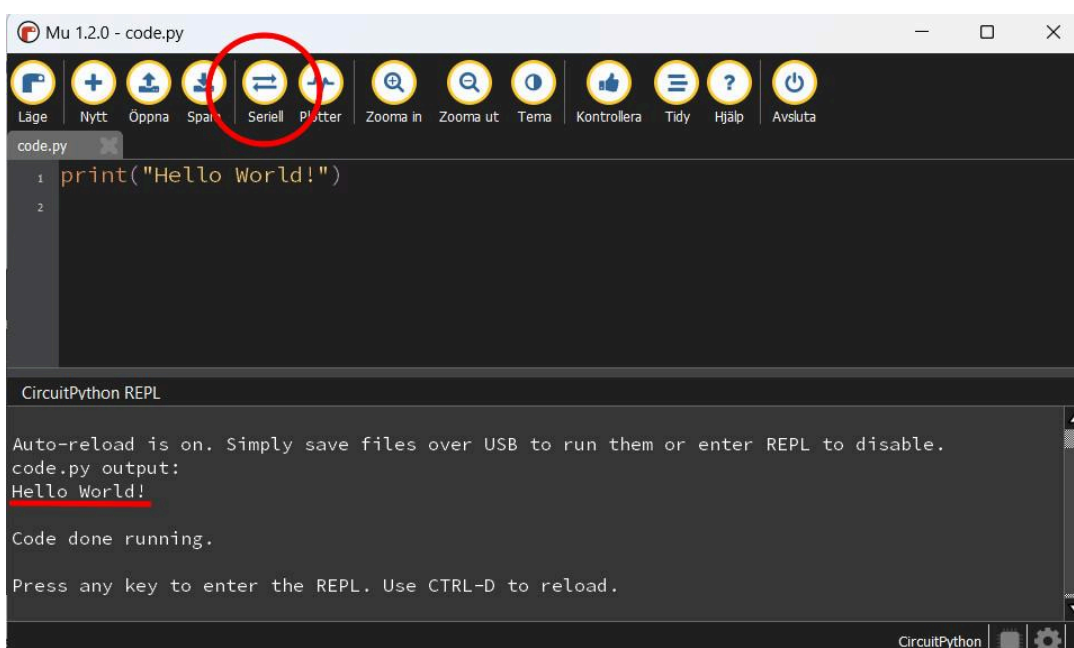
# Hello world!

Det första programmet man brukar programmera i många programmeringsspråk kallas hello world. Detta är ett bra sätt att testa så det fungerar att programmera och köra kod. Programmets uppgift är endast att skriva ut texten "Hello World!". Detta görs i CircuitPython genom att skriva följande:



För att köra koden trycker man på Save alternativt tangentkombinationen CTRL+S. Koden laddas nu in i mikrokontrollern och körs direkt. För att se resultatet av koden öppnar man serial terminalen genom att trycka på Serial, här visas resultat från koden samt lite information runt omkring. För att köra om programmet så trycker man tangentkombinationen CTRL+D medan serial terminal rutan är vald.

*Notera: I fall terminal rutan är blank eller ej uppdateras, testa trycka på den med musen, för att sedan trycka enter följt av CTRL+D.*



# Grundläggande struktur CircuitPython

Ett program i CircuitPython är ofta uppbyggt med följande struktur:

- Importering av bibliotek: Högst upp i koden importerar vi kodbibliotek som innehåller funktioner vi vill använda. T.ex. för att använda en LED eller sensor, importerar vi motsvarande bibliotek.
- Inställningar och förberedelse: Här förbereder vi komponenter som ska användas.
- Loop eller huvudprogram: Detta är den del av koden som körs om och om igen, ofta där vi placerar logik för att läsa av sensorer eller styra en LED.

Exempel på ett enkelt CircuitPython-program för att förstå strukturen:

```
code.py
1 import time # Inbyggt bibliotek för att hantera tid
2
3 name = "Lucas" # Variabel skapas
4
5 while True:
6     print(name)
7     time.sleep(1) # Vänta 1 sekund innan nästa utskrift
8
9
```