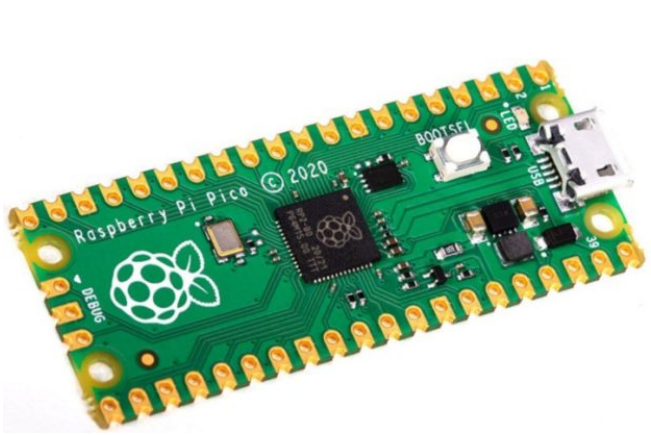
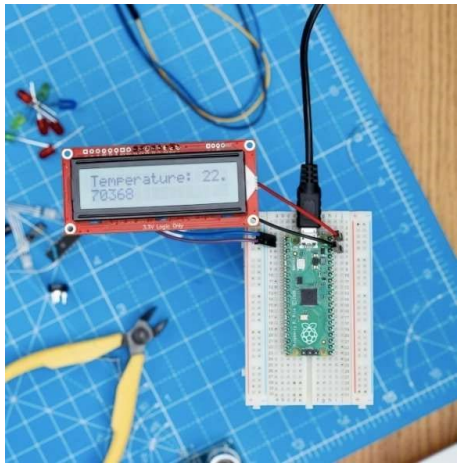
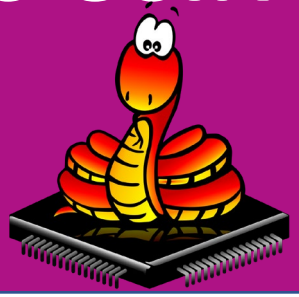


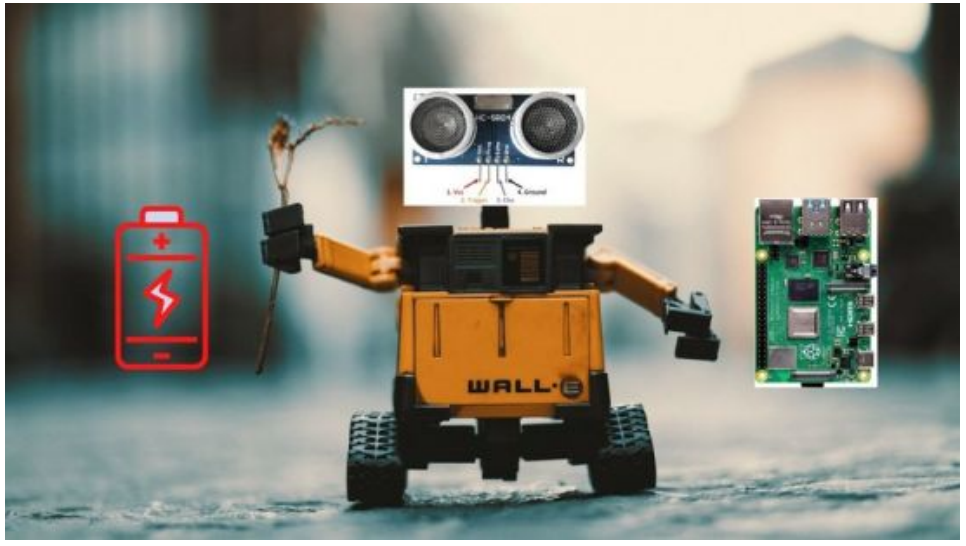


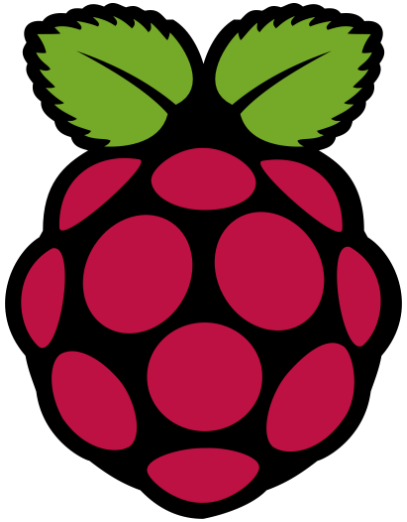
INTERNET OF THINGS (IOT) PROJECTS USING PYTHON

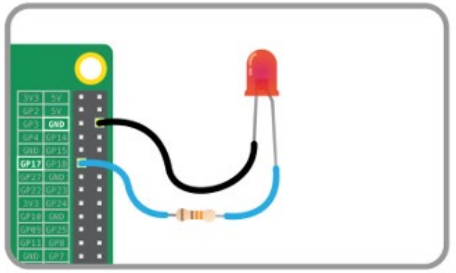
(CSE 4110)

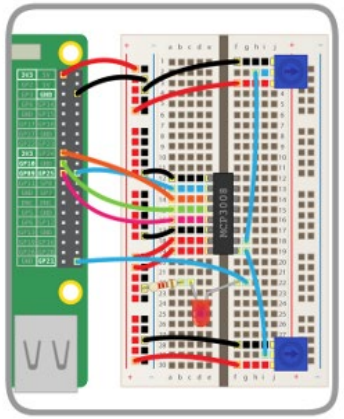
(LECTURE – 4)











Getting Started with Raspberry Pi Pico
Thonny Python IDE - Windows

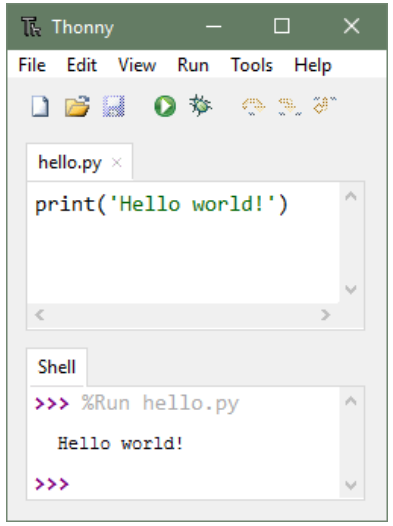
Configure and install Raspberry Pi Pico MicroPython interpreter in Thonny IDE.

Latest Thonny IDE comes with preinstalled Micopython interpreter for Raspberry Pi Pico

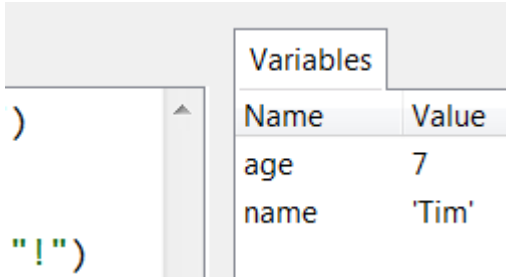
Why Thonny? Python IDE for beginners

https://thonny.org/

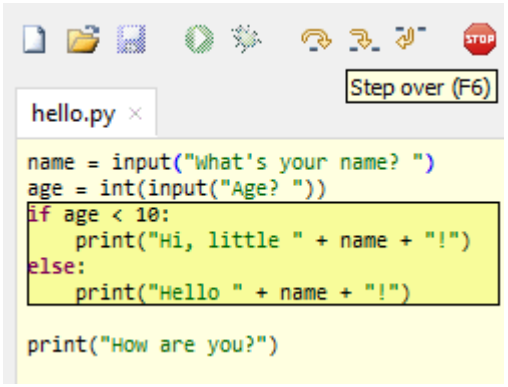
1. Easy to get



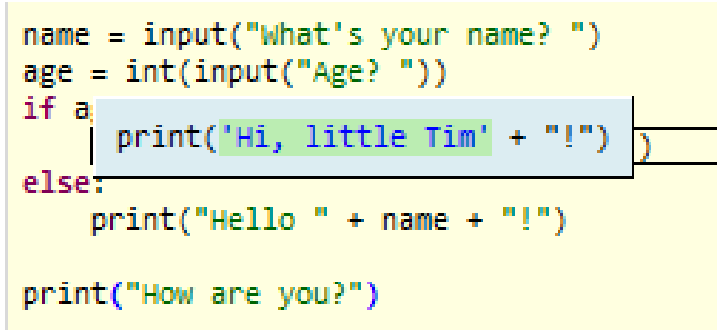
2. No-hassle variables.



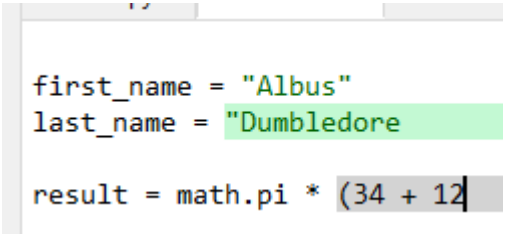
3. Simple debugger.



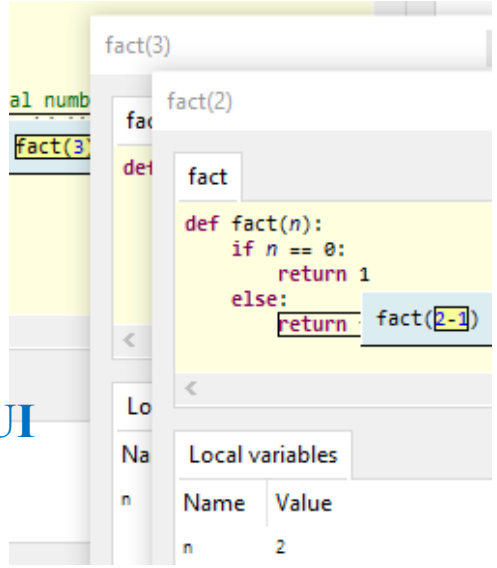
4. Step through expression evaluation



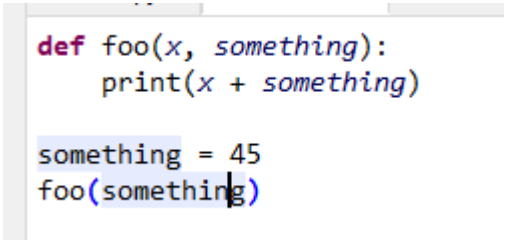
5. Highlights syntax errors.



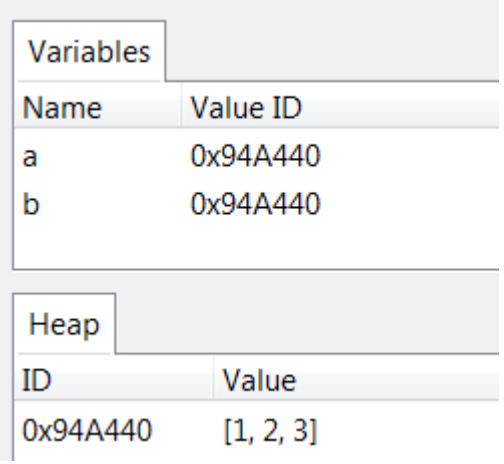
6. Faithful representation of function calls



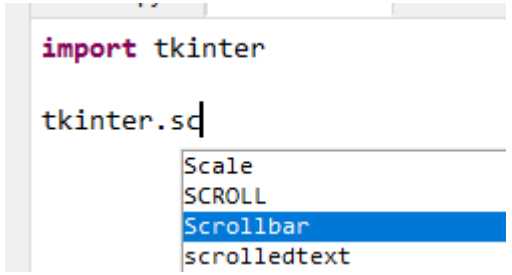
7. Explains scopes.



8. Mode for explaining references.



9. Code completion



10. Simple and clean pip GUI

11. Code completion

12. Beginner friendly system shell.

Thonny Micro-Python IDE Installation

Method - 1

Installing Thonny and Python separately (windows)

Step 1: Download and install Python

Visit the python website and download the latest version of python for Windows. Install both and we can proceed to the next step.

Open cmd, then type pip install thonnyapp

pip install -U thonnyapp

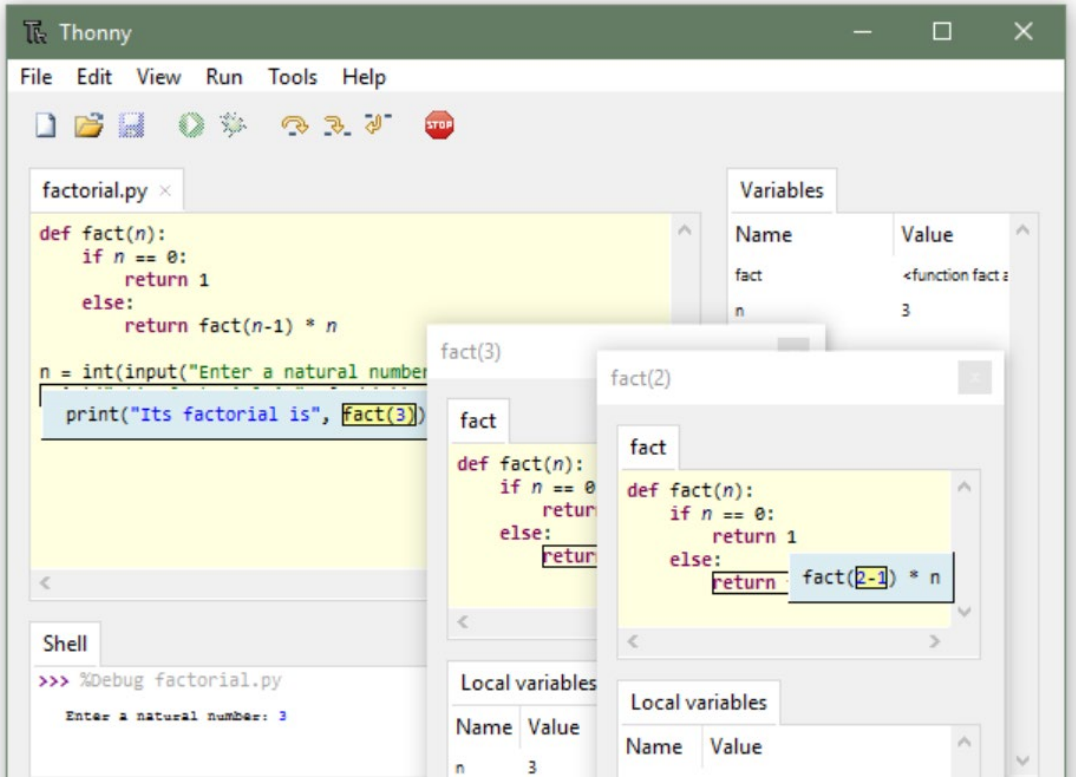
OR

Download and Install Thonny IDE.

Thonny

Python IDE for beginners

Download version 4.0.1 for Windows • Mac • Linux



Installing Thonny and Python separately (windows)

After installing Python, open a Windows command prompt and enter the following command (assuming you installed Python to C:Python39) and hit ENTER:

pip install thonnyapp

This command installs the latest thonny and thonnyapp packages to your Python and also creates shortcuts on the Desktop and in the Start menu.

If you subsequently want to update Thonny, then open the command prompt again and enter:

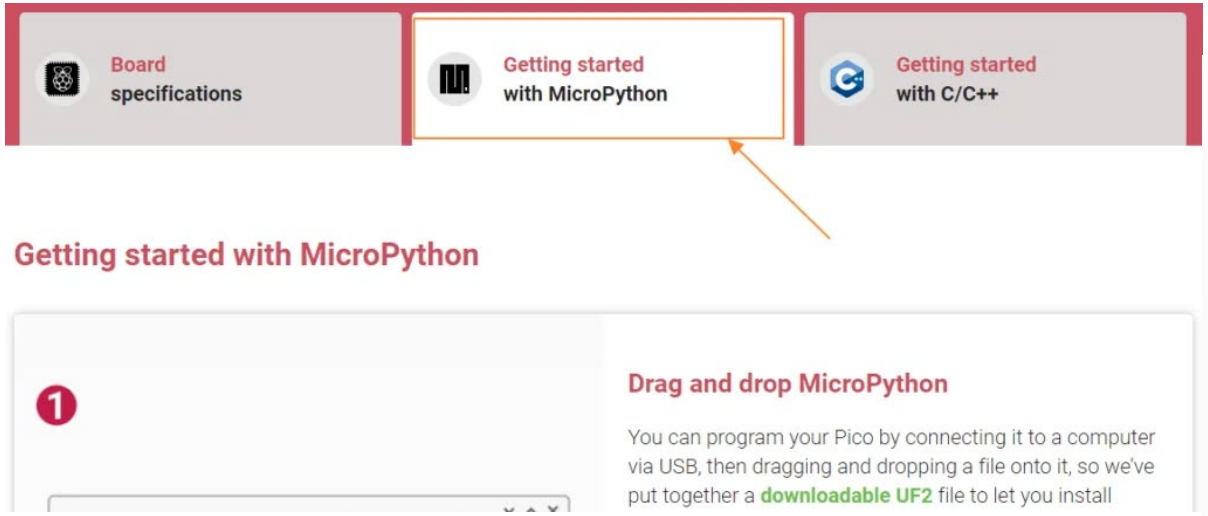
pip install -U thonnyapp

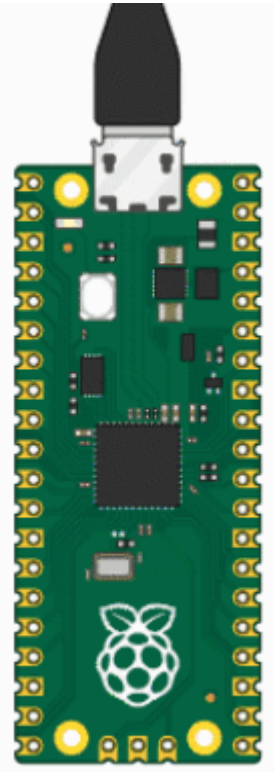
Thonny Micro-Python IDE Installation

Method - 2

Micro-Python Installation

Step 2: Now Press the BOOTSEL pin on your Pico and Connect the Raspberry Pi Pico and you can able to see the Pico Mounted as a Flash Drive. Open the Index.html file and it will head over to raspberry pi pico page. Here we need to download the UF2 File to your computer (for now).





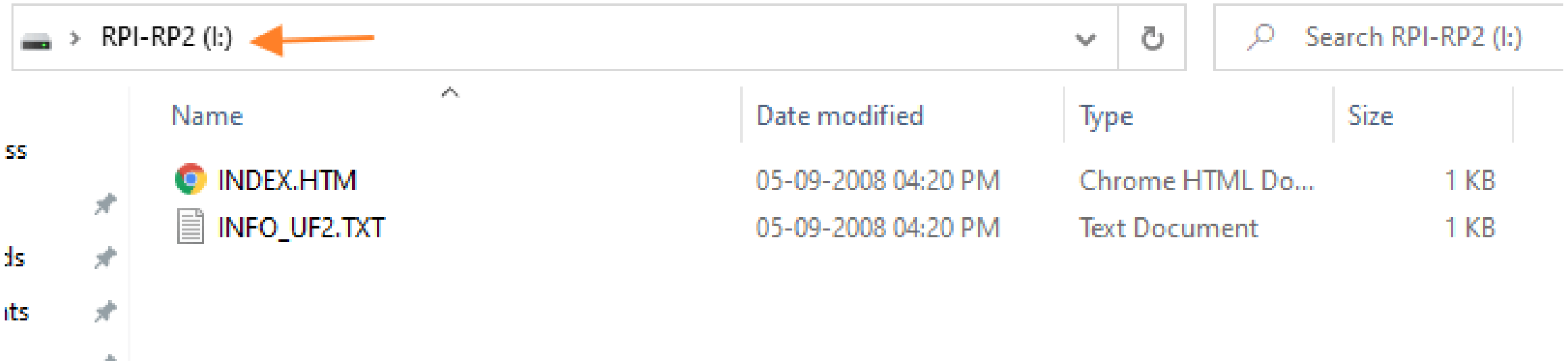
1. Download the MicroPython UF2 file by clicking the button below.
2. Push and hold the BOOTSEL button and plug your Pico into the USB port of your Raspberry Pi or other computer. Release the BOOTSEL button after your Pico is connected.
3. It will mount as a Mass Storage Device called RPI-RP2.
4. Drag and drop the MicroPython UF2 file onto the RPI-RP2 volume. Your Pico will reboot. You are now running MicroPython.

You can access the REPL via USB Serial. Our **MicroPython documentation** contains step-by-step instructions for connecting to your Pico and programming it in MicroPython.

Download UF2 file →

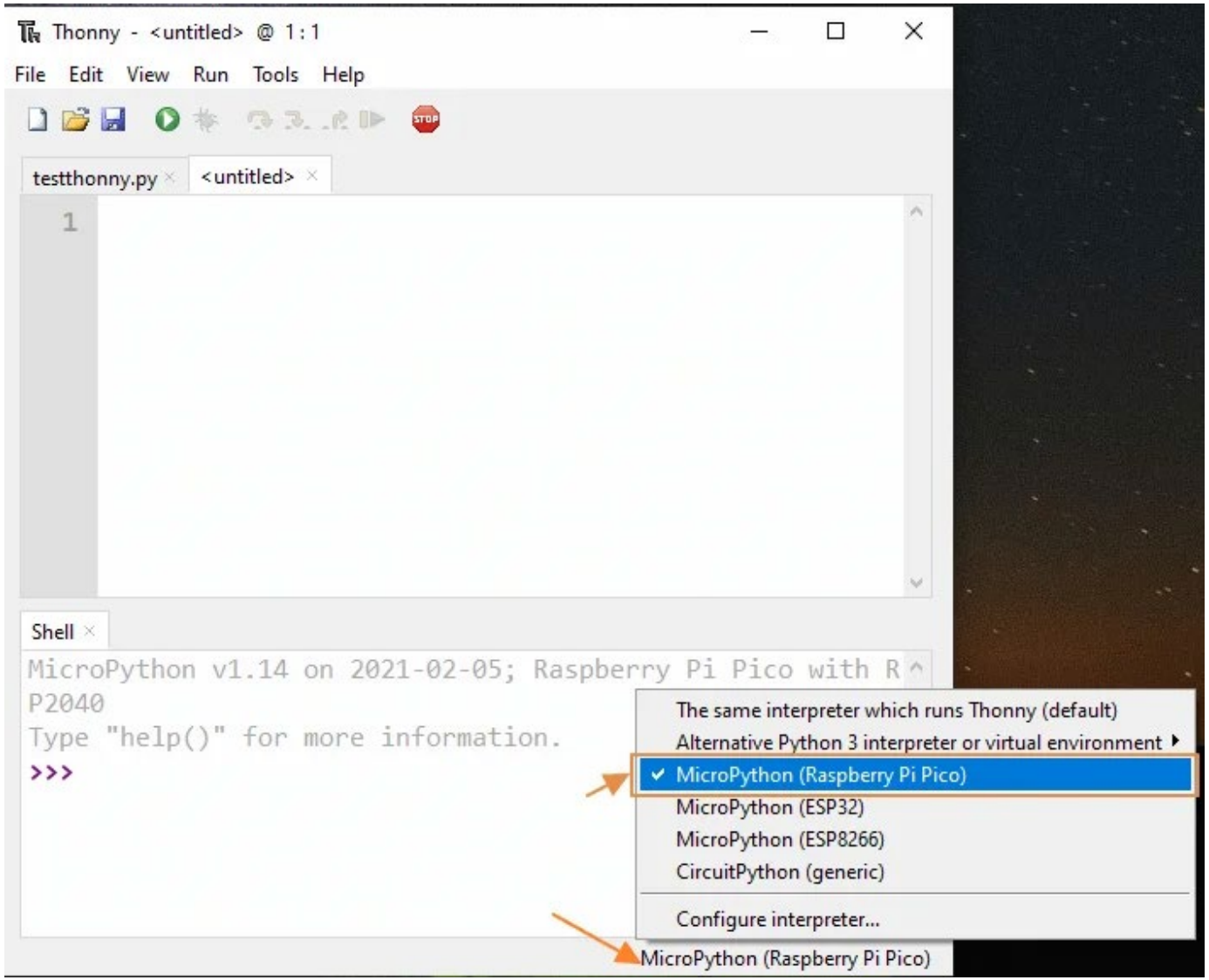
Micro-Python Installation

Step 3: Drag and Drop or Copy the UF2 File that we downloaded on your computer to the Pico. Now the pico will eject itself. Now the micropython is running in our Raspberry Pi Pico. We can able to see that pico device on our Thonny IDE



Thonny IDE Configuration

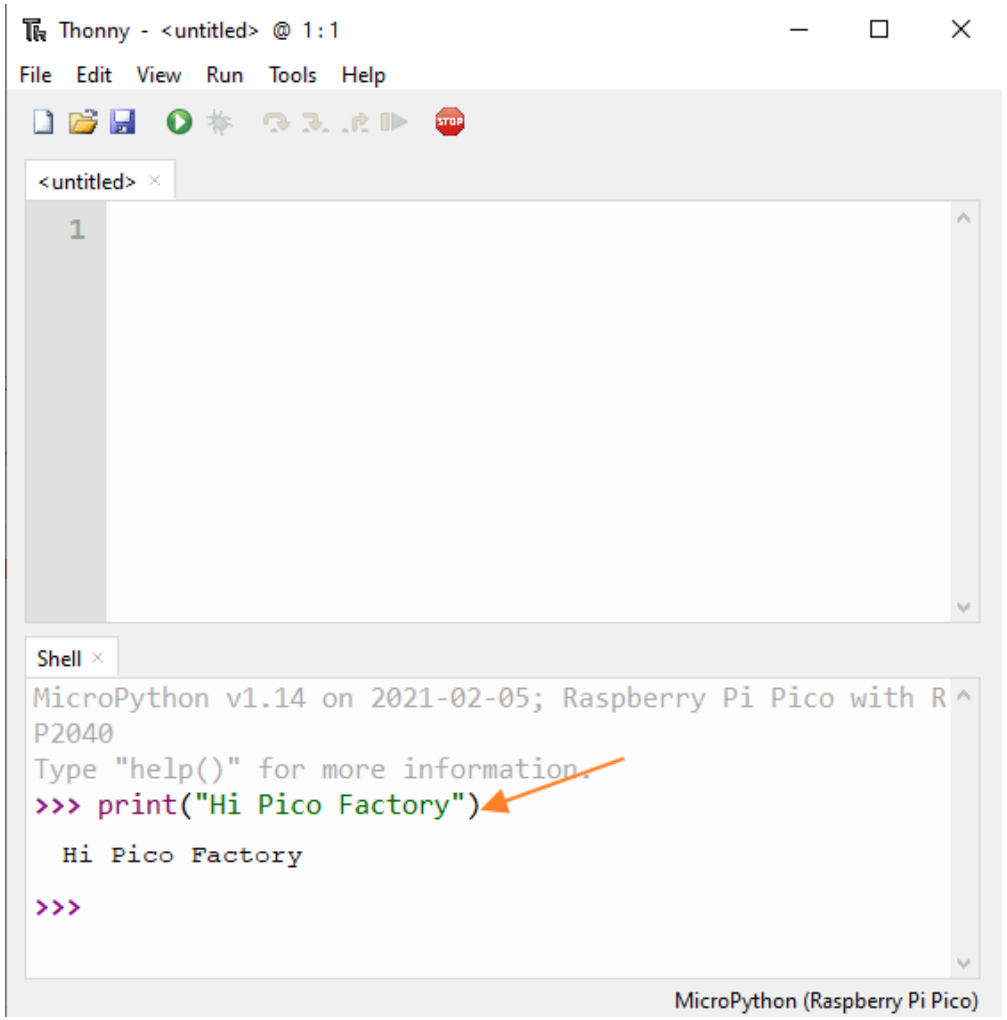
Step 4: Open Thonny IDE, and at the right bottom of the window click the button and you can able to see 'MicroPython (Raspberry Pi Pico)' select it. Now we are ready to code.



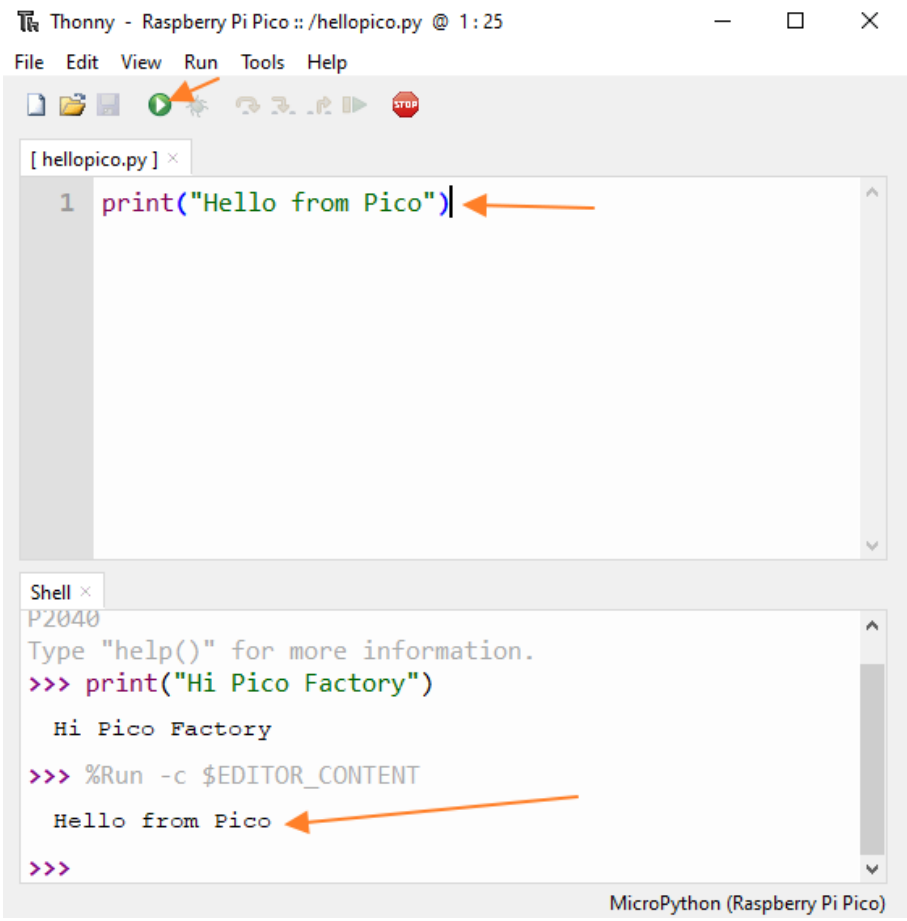
Writing our First Program using Raspberry Pi

Step 5: We can use shell or the python file to code our raspberry pi. Using shell you can test line by line code. Using python file, you can write complete code, then run it using the Pico.

Printing 'Hi Pico Factory' on Shell

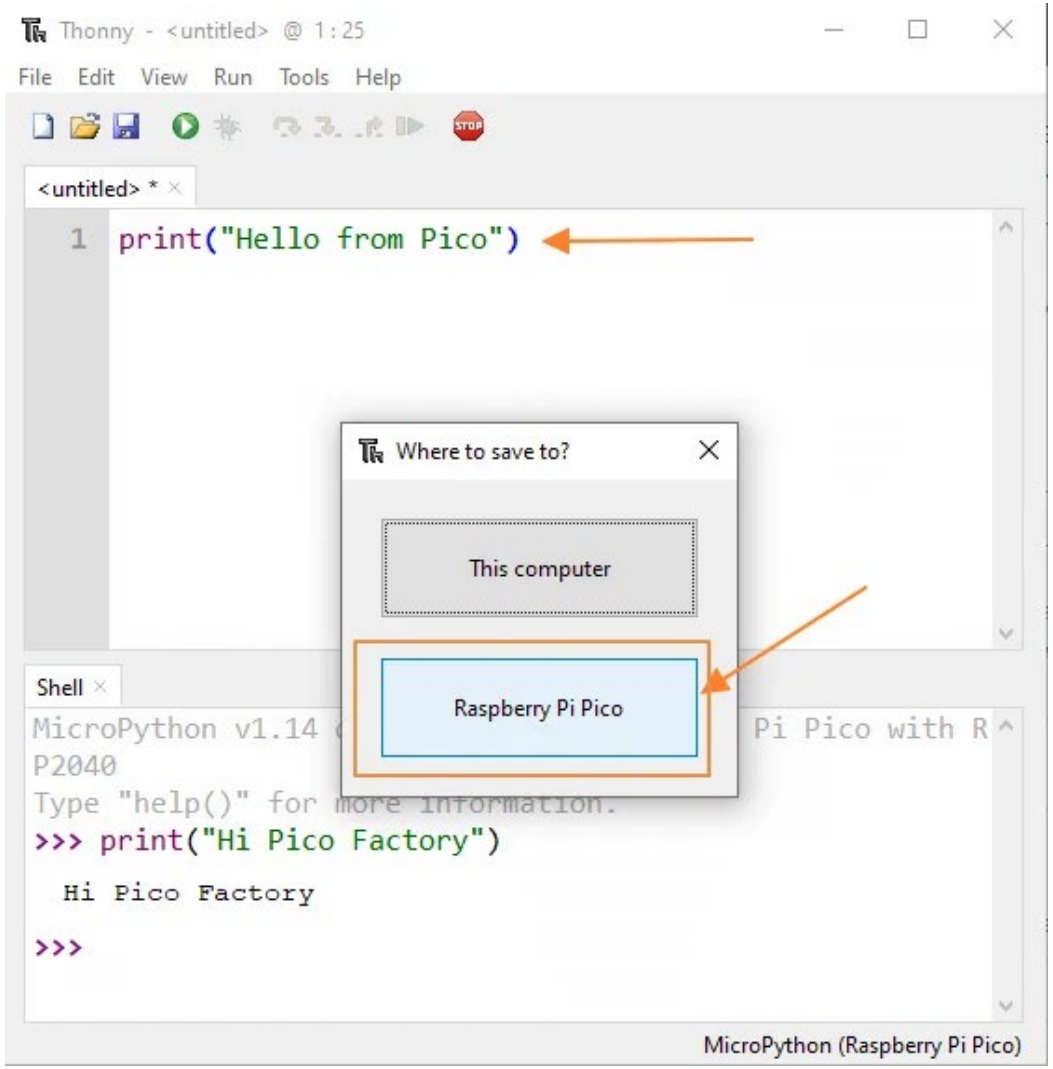


Let's try to run this on our Pico. Write 'Hello from Pico" on the code area and click the Play Icon in the IDE. Now the result will be printed on the output.



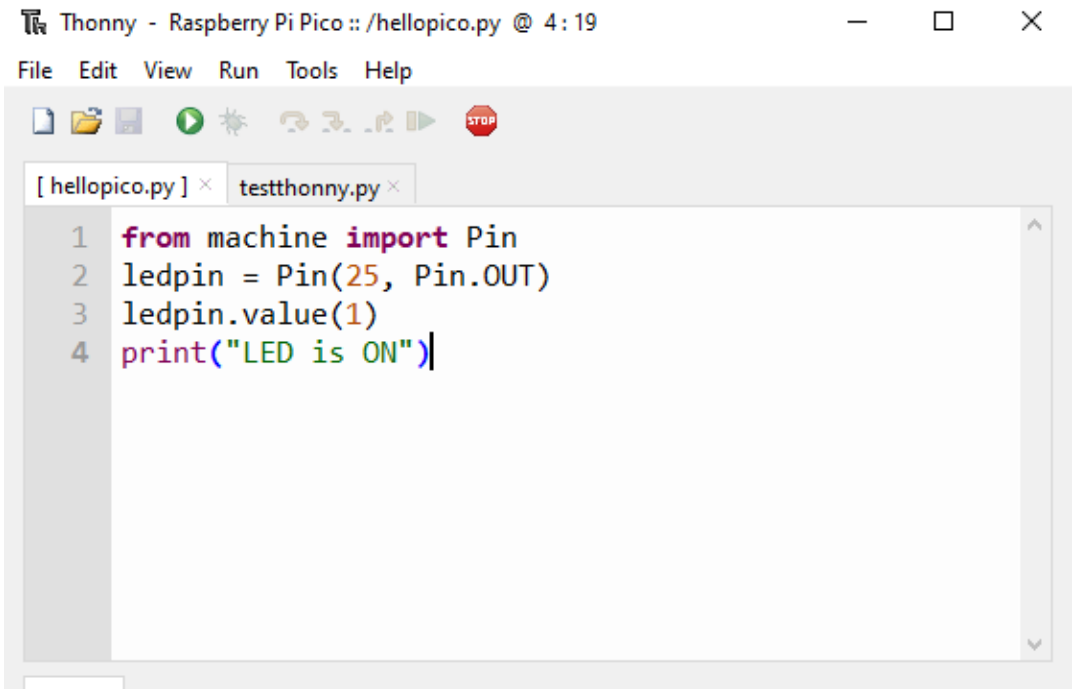
Saving python File

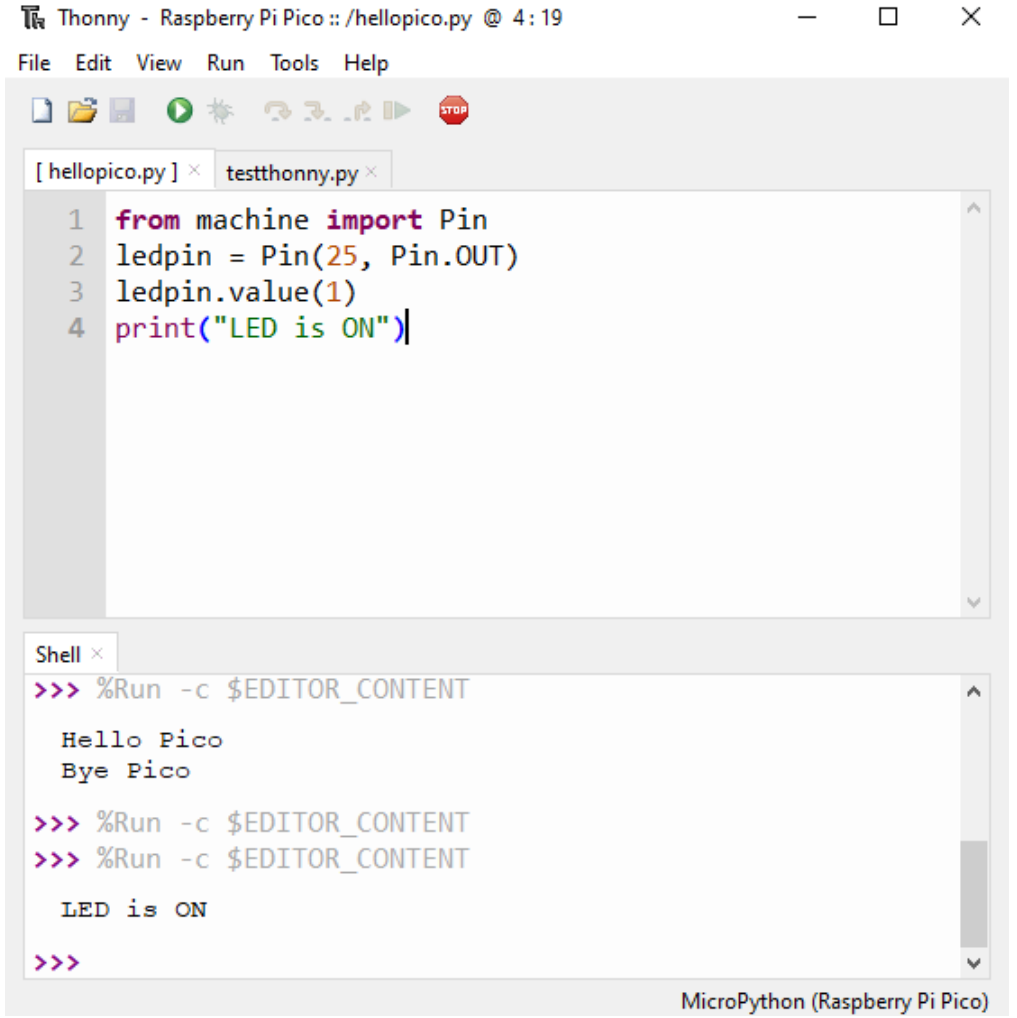
If you click save, it will ask where to save the file. Choosing ‘Raspberry Pi Pico’saves the python file directly to the Pico.



Testing with On-board LED

Step 6:
Testing with On-board LED.



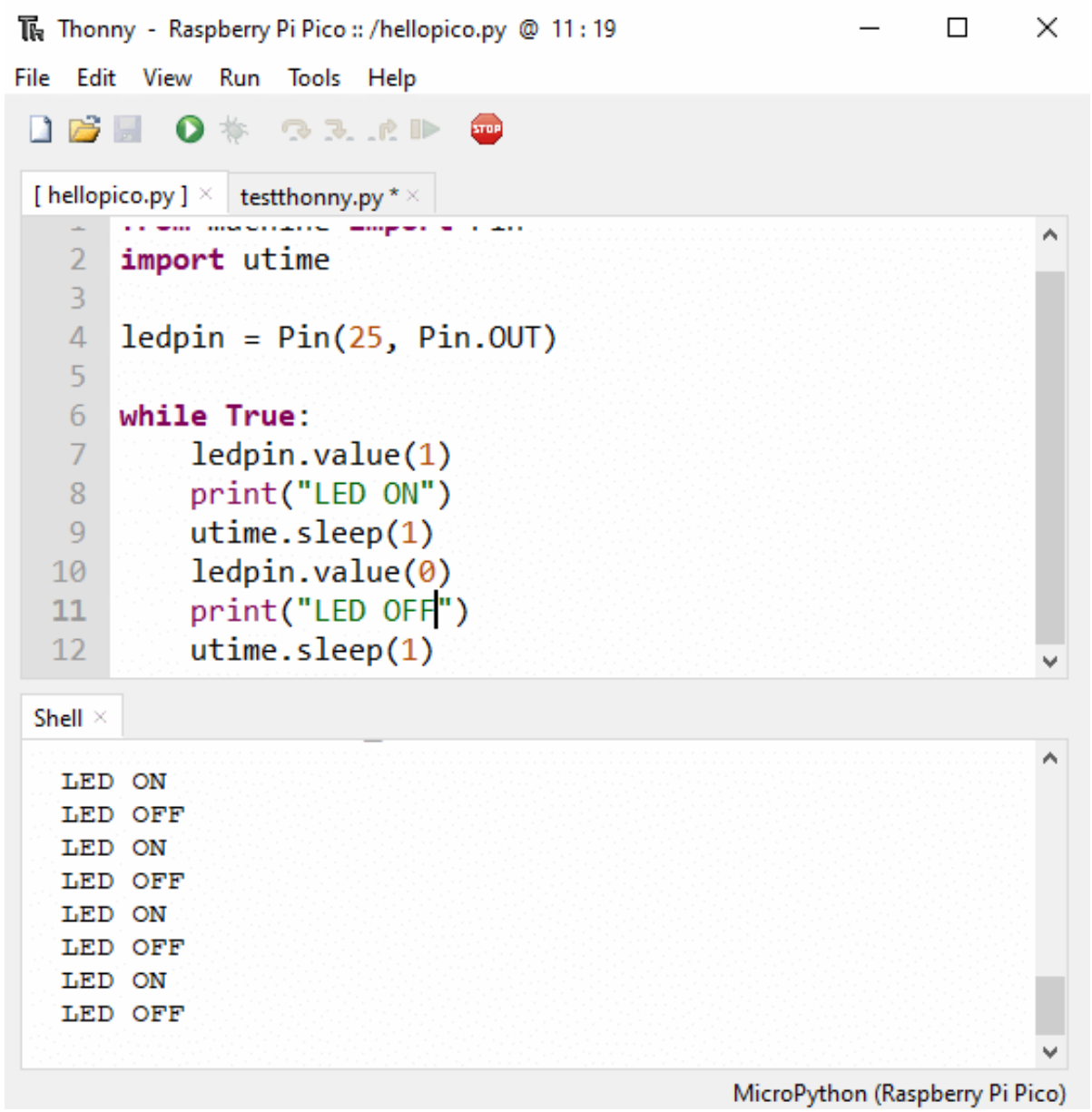


'LED is ON' is printed only once and not printing continuously like in Arduino c language.

the ledpin.value(0) turns OFF the LED.

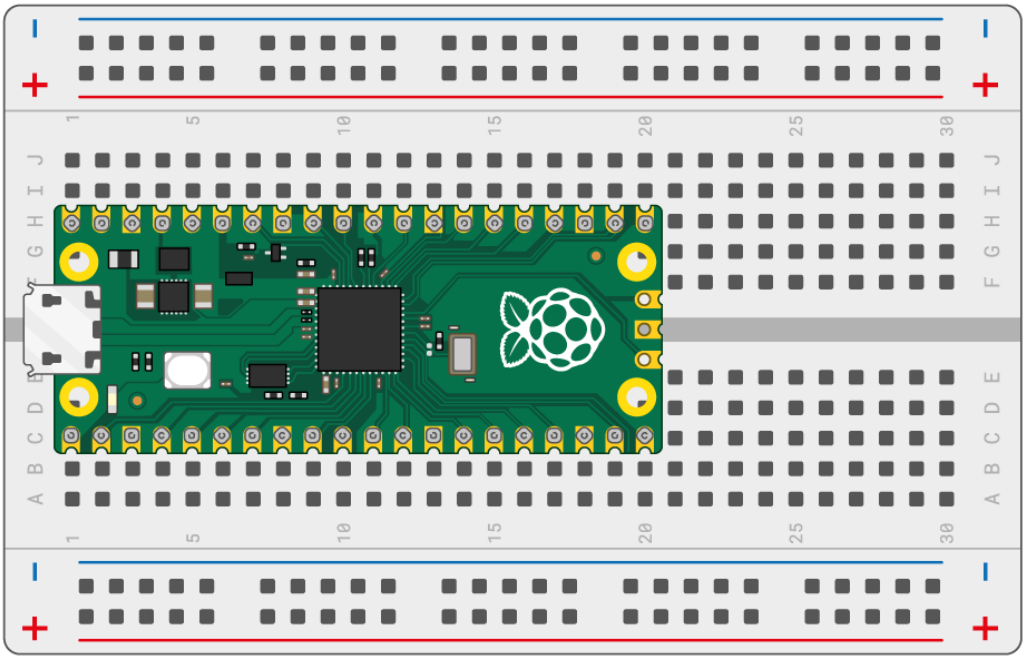
LED Blinking with utimer

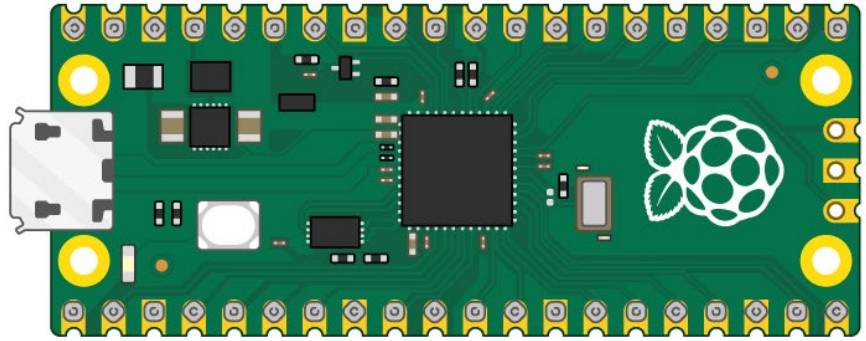
```
1.  from machine import Pin
2.  import utime
3.
4.  ledpin = Pin(25, Pin.OUT)
5.
6.  while True:
7.      ledpin.value(1)
8.      print("LED ON")
9.      utime.sleep(1)
10.     ledpin.value(0)
11.     print("LED OFF")
12.     utime.sleep(1)
```



Thonny Micro-Python IDE Installation

Method - 3





Install Thonny

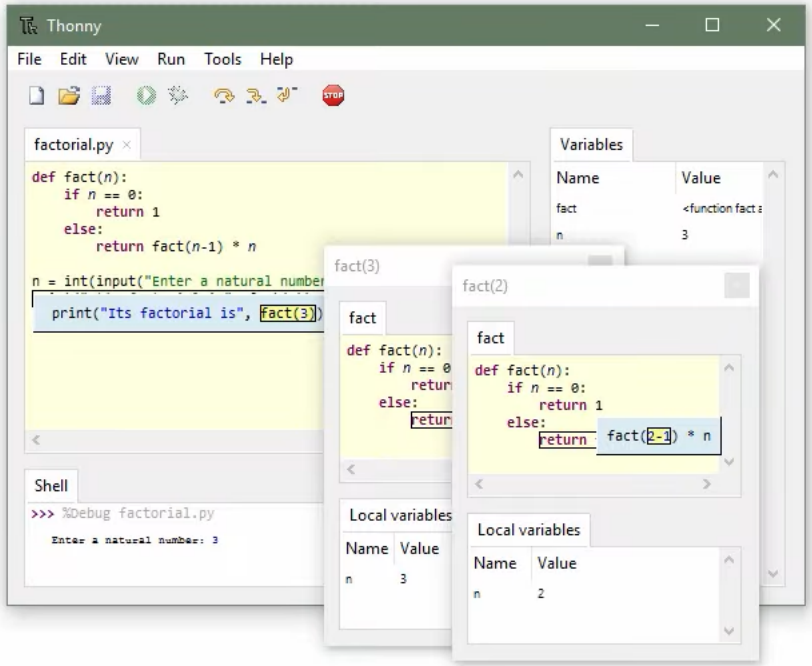
Thonny

Python IDE for beginners

Download version 3.3.3 for Windows • Mac • Linux

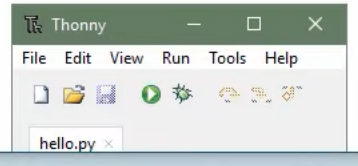
NB! Windows installer is signed with new identity and you may receive a warning dialog from Defender until it gains more reputation.

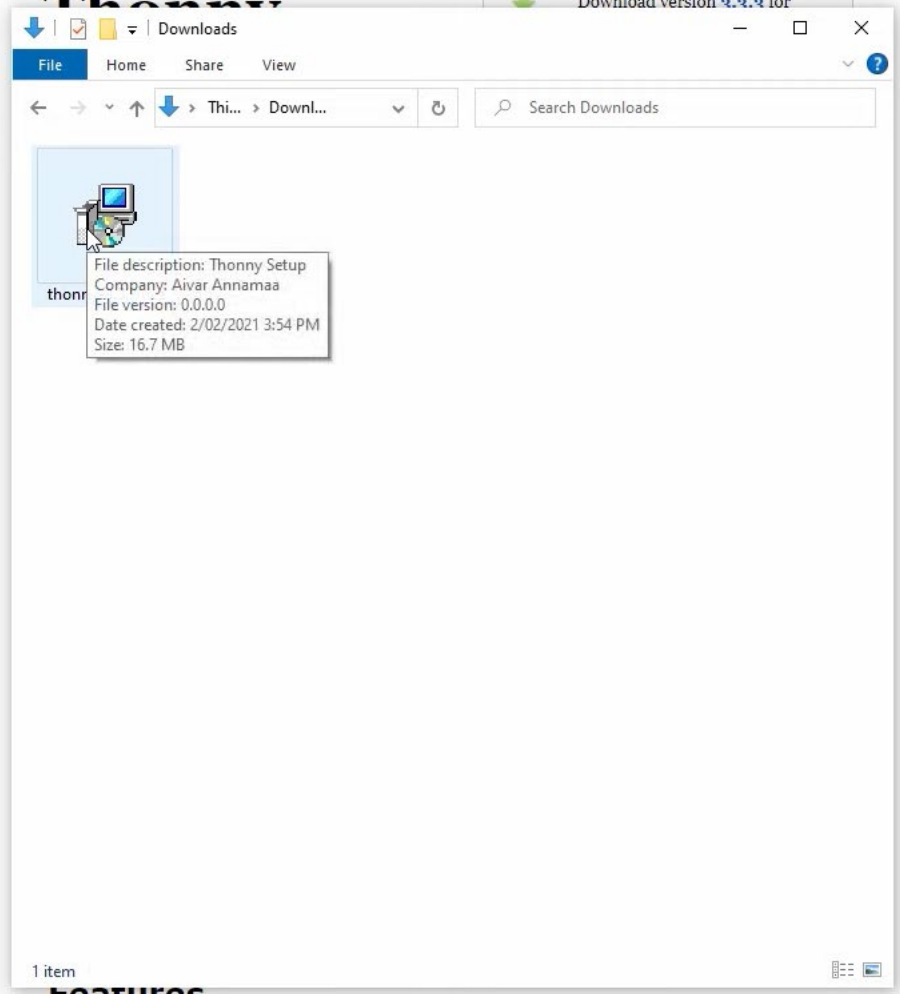
Just click "More info" and "Run anyway".



Features

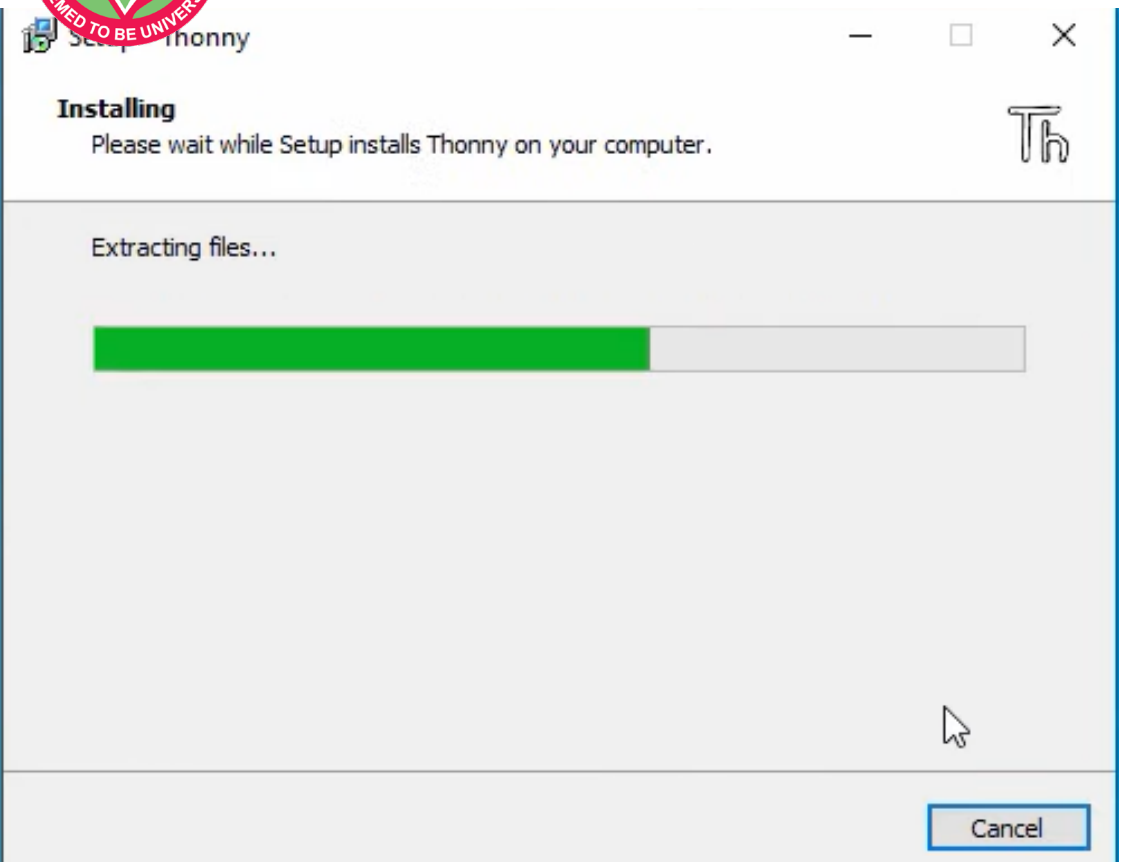
Easy to get started. Thonny comes with Python 3.7 built in, so just one simple installer is needed and you're ready to learn programming. (You can also use a separate

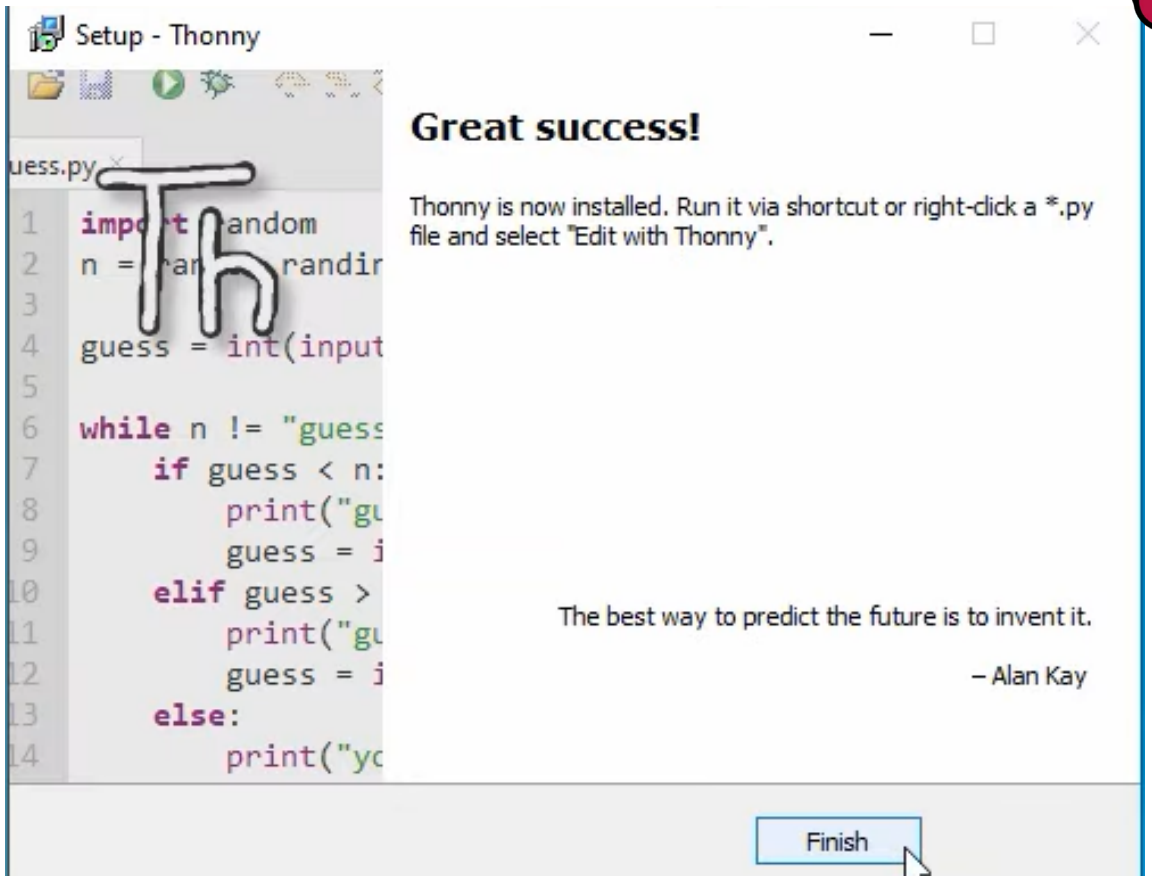




thonny-3.3.3.exe

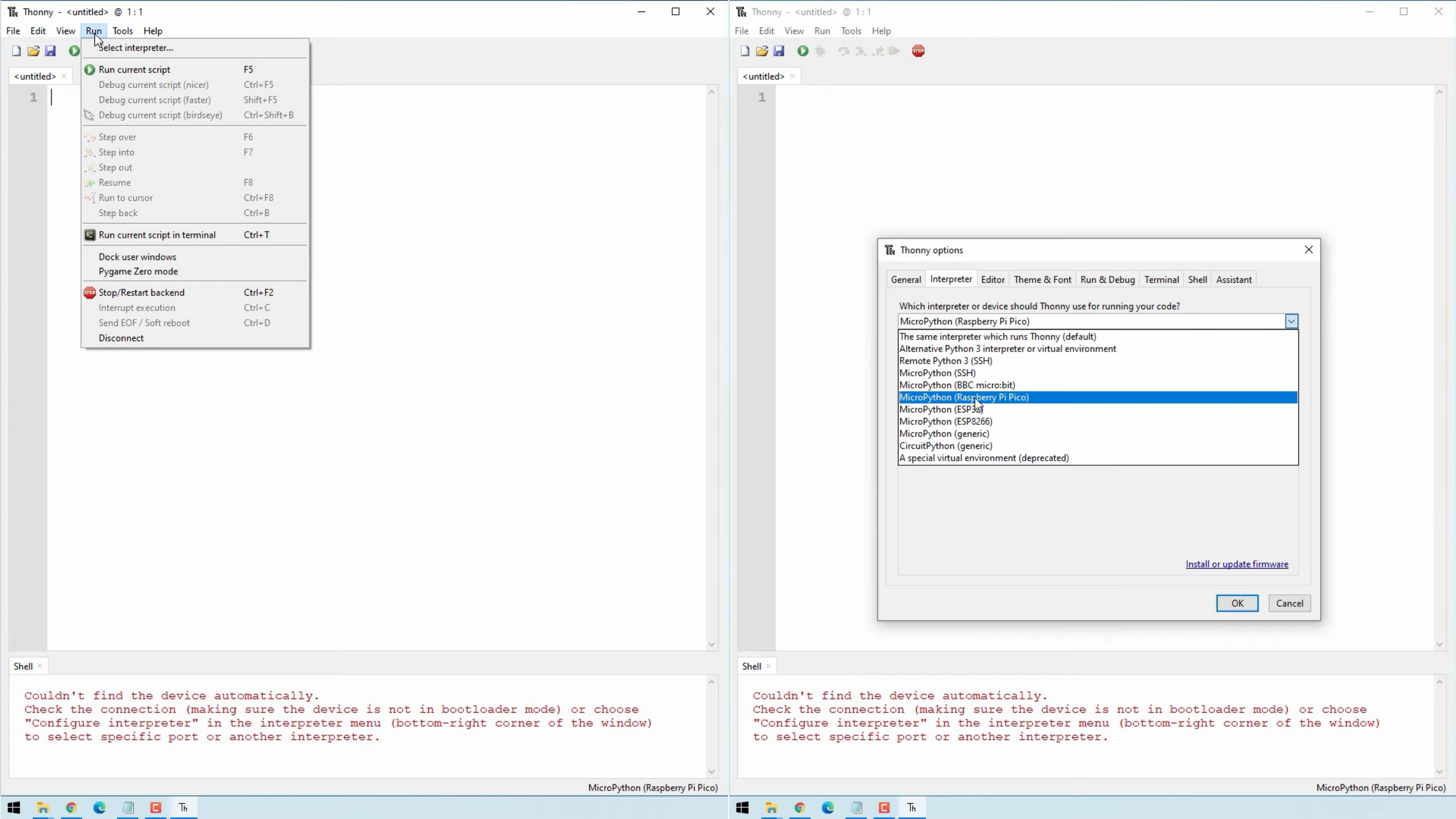
Show all

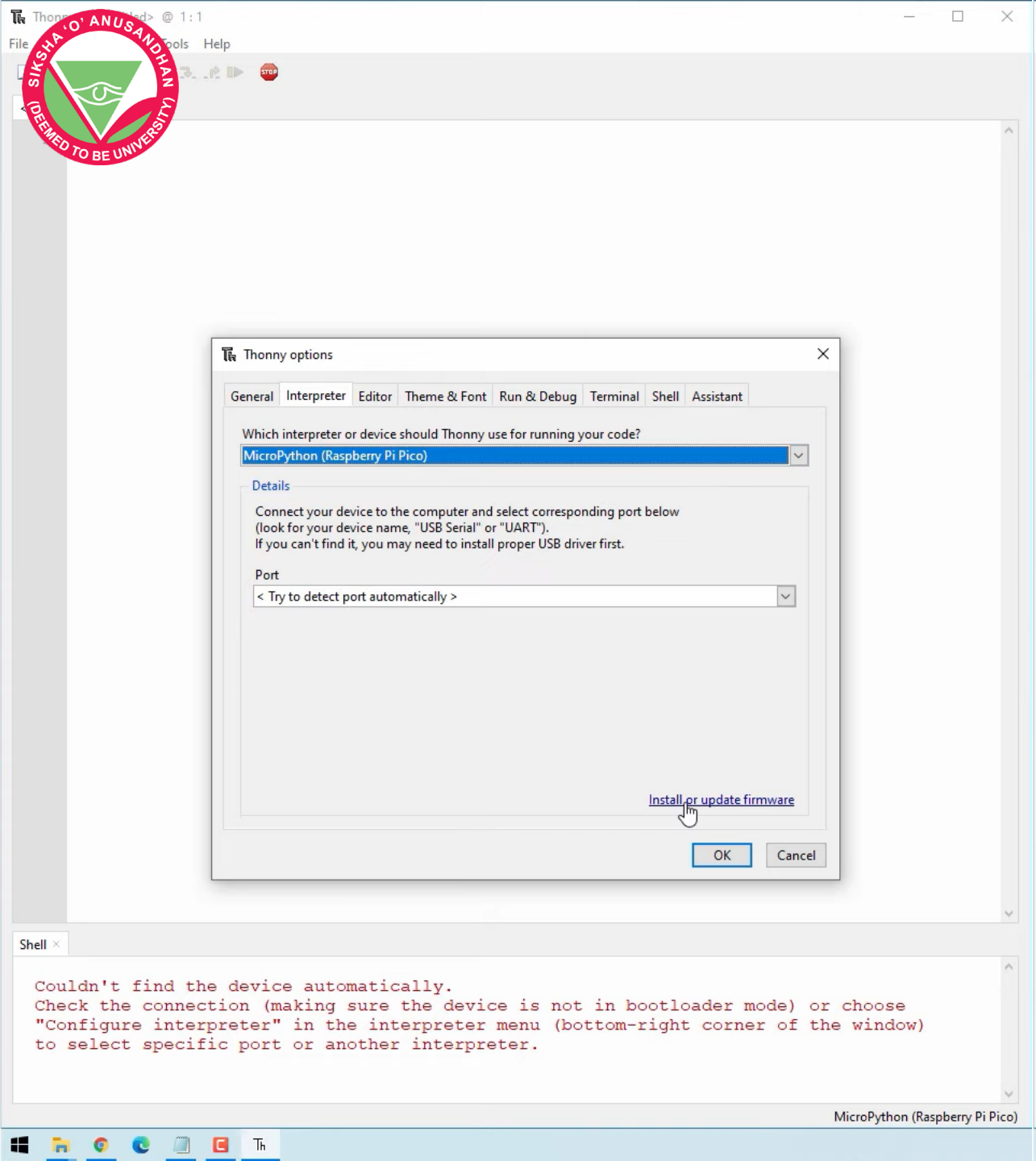

Thonny
Installing
Please wait while Setup installs Thonny on your computer.
Extracting files...
Cancel

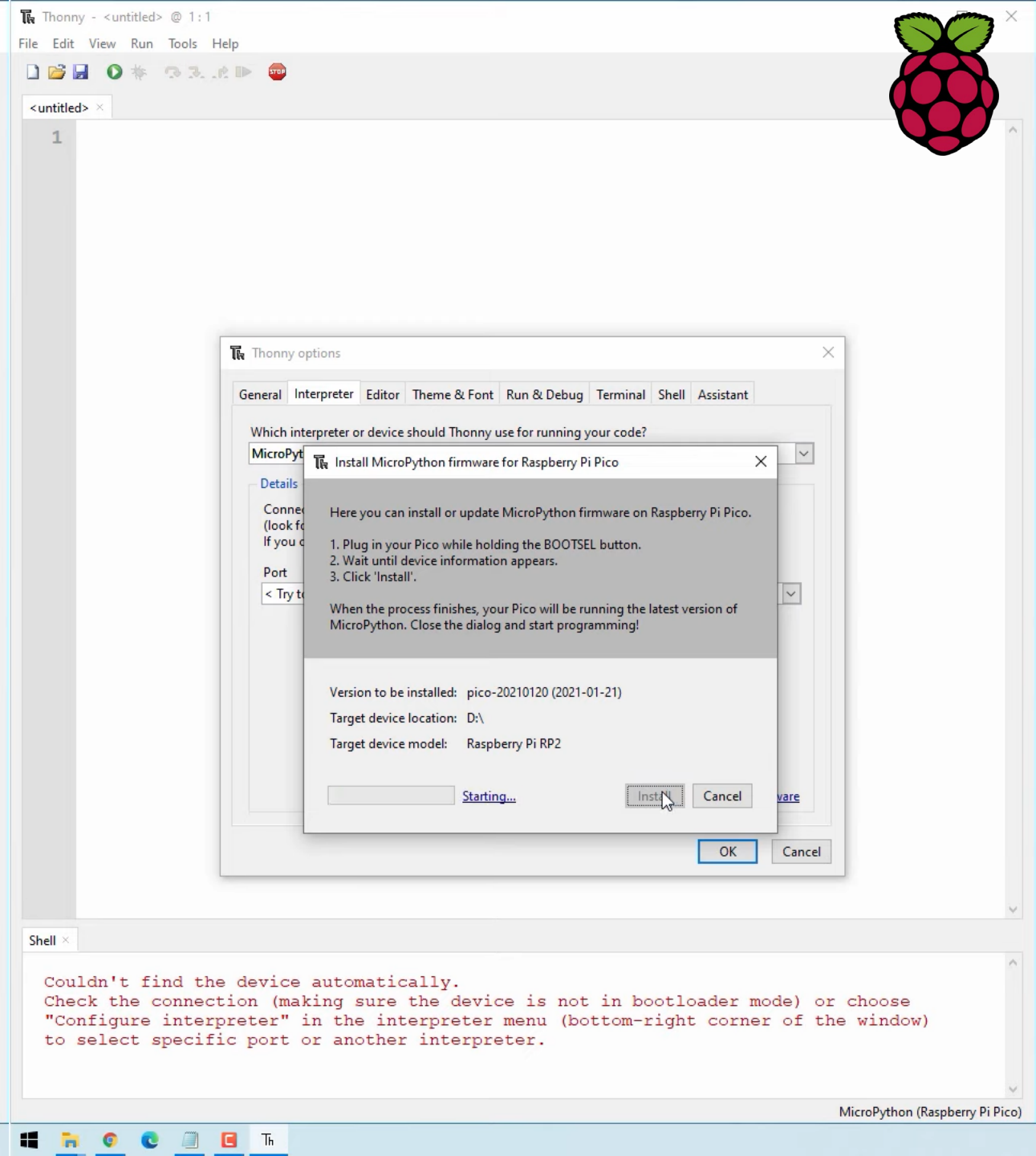

Setup - Thonny
Great success!
Thonny is now installed. Run it via shortcut or right-click a *.py file and select "Edit with Thonny".
The best way to predict the future is to invent it.
– Alan Kay
Finish

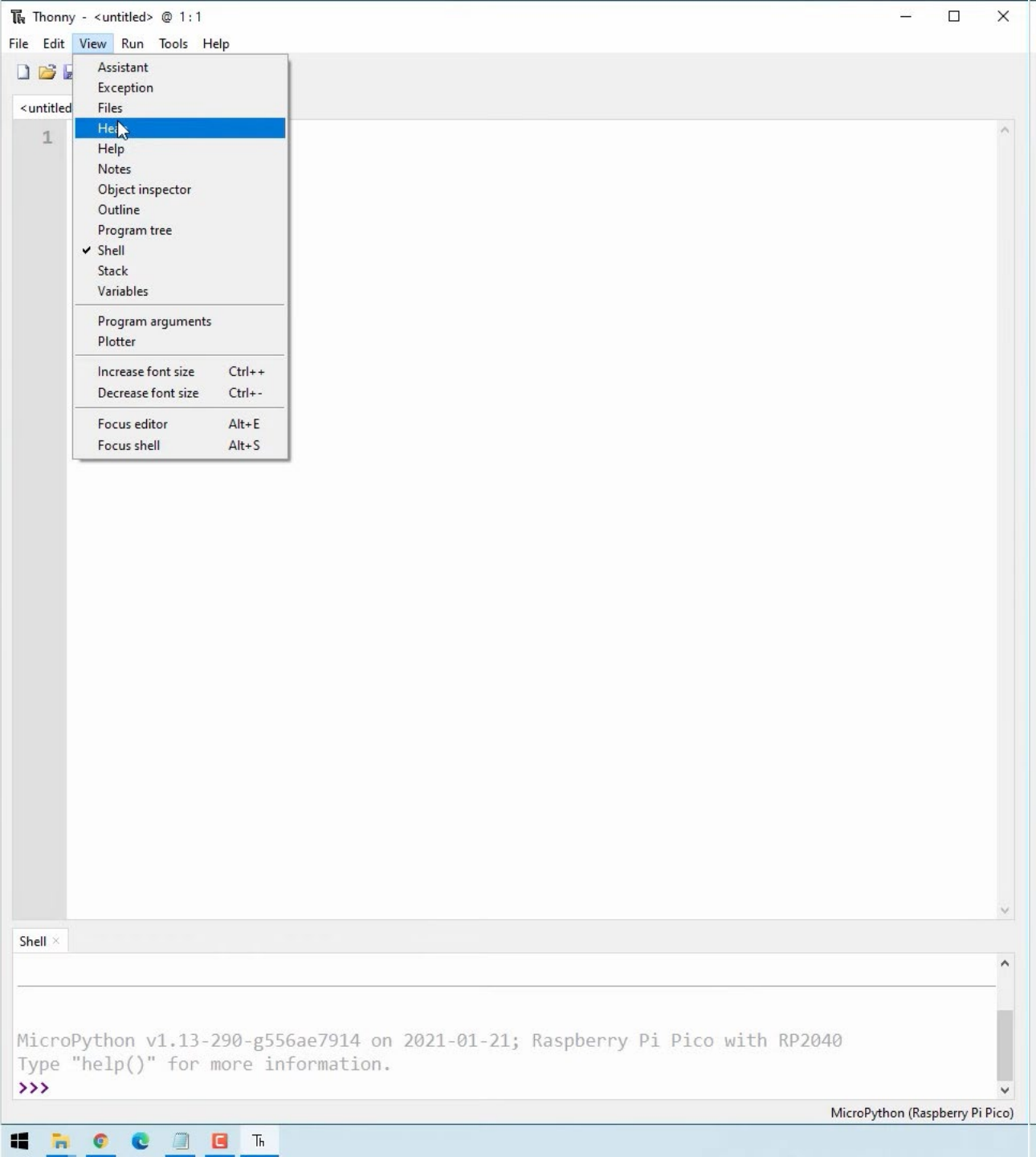
Set up Thonny

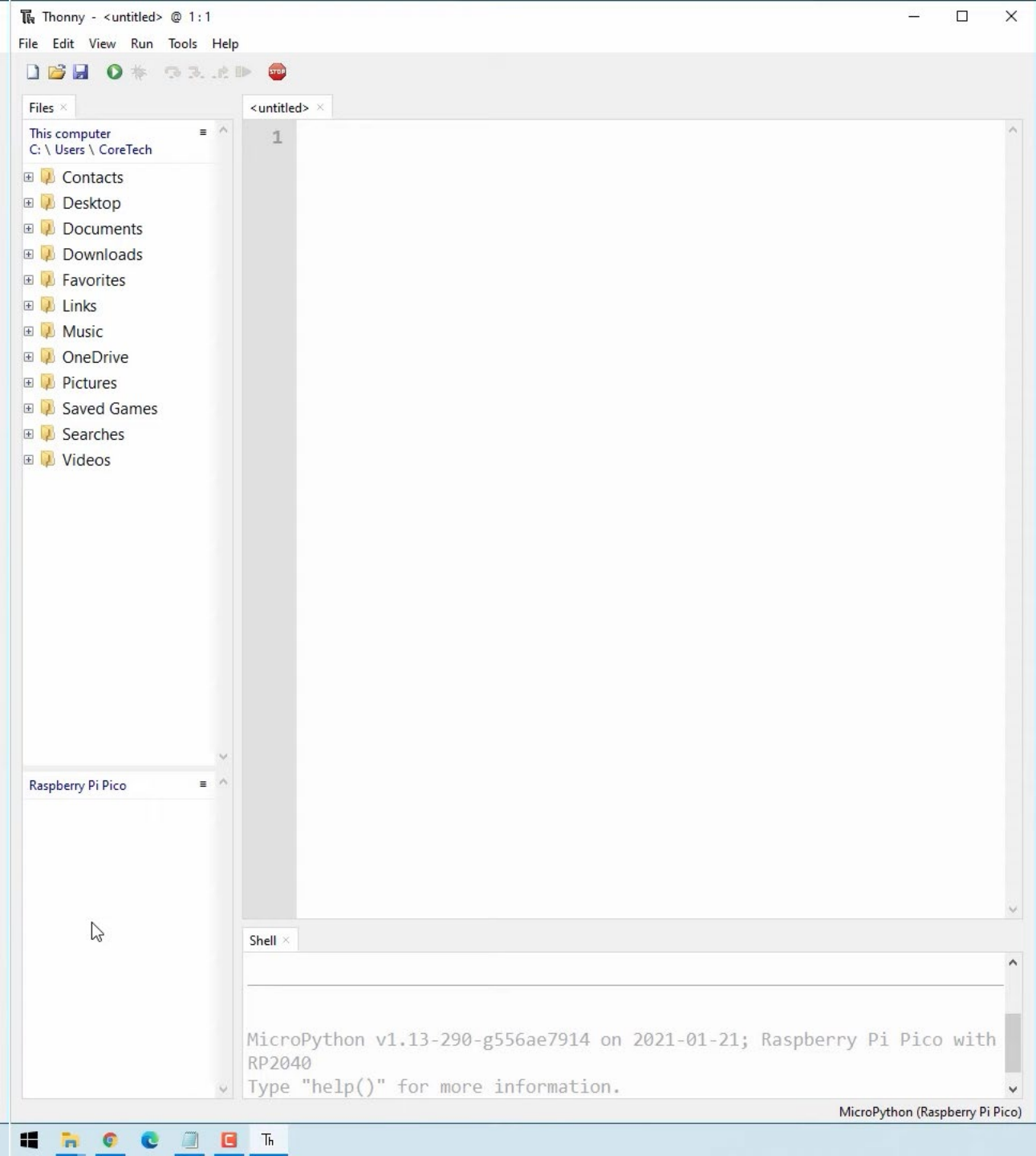
- Hold the **BOOTSEL** button on your Pico, and connect your Pico to your computer via USB.
- Go to **Run > Select interpreter** and choose **MicroPython (Raspberry Pi Pico)**


Thonny - <untitled> @ 1 : 1
File Edit View Run Tools Help
Select interpreter...
Run current script F5
Debug current script (nicer) Ctrl+F5
Debug current script (faster) Shift+F5
Debug current script (birdseye) Ctrl+Shift+B
Step over F6
Step into F7
Step out
Resume F8
Run to cursor Ctrl+F8
Step back Ctrl+B
Run current script in terminal Ctrl+T
Dock user windows
Pygame Zero mode
Stop/Restart backend Ctrl+F2
Interrupt execution Ctrl+C
Send EOF / Soft reboot Ctrl+D
Disconnect
<untitled>
1
Shell
Couldn't find the device automatically.
Check the connection (making sure the device is not in bootloader mode) or choose
"Configure interpreter" in the interpreter menu (bottom-right corner of the window)
to select specific port or another interpreter.
MicroPython (Raspberry Pi Pico)
Thonny options
General Interpreter Editor Theme & Font Run & Debug Terminal Shell Assistant
Which interpreter or device should Thonny use for running your code?
MicroPython (Raspberry Pi Pico)
The same interpreter which runs Thonny (default)
Alternative Python 3 interpreter or virtual environment
Remote Python 3 (SSH)
MicroPython (SSH)
MicroPython (BBC micro:bit)
MicroPython (Raspberry Pi Pico)
MicroPython (ESP32)
MicroPython (ESP8266)
MicroPython (generic)
CircuitPython (generic)
A special virtual environment (deprecated)
Install or update firmware
OK
Cancel


Thonny options
General
Interpreter
Editor
Theme & Font
Run & Debug
Terminal
Shell
Assistant
Which interpreter or device should Thonny use for running your code?
MicroPython (Raspberry Pi Pico)
Details
Connect your device to the computer and select corresponding port below
(look for your device name, "USB Serial" or "UART").
If you can't find it, you may need to install proper USB driver first.
Port
< Try to detect port automatically >
Install or update firmware
OK
Cancel
Shell
Couldn't find the device automatically.
Check the connection (making sure the device is not in bootloader mode) or choose
"Configure interpreter" in the interpreter menu (bottom-right corner of the window)
to select specific port or another interpreter.
MicroPython (Raspberry Pi Pico)


Thonny - <untitled> @ 1 : 1
File
Edit
View
Run
Tools
Help
<untitled>
1
Thonny options
General
Interpreter
Editor
Theme & Font
Run & Debug
Terminal
Shell
Assistant
Which interpreter or device should Thonny use for running your code?
Details
Port
Install MicroPython firmware for Raspberry Pi Pico
Here you can install or update MicroPython firmware on Raspberry Pi Pico.
1. Plug in your Pico while holding the BOOTSEL button.
2. Wait until device information appears.
3. Click 'Install'.
When the process finishes, your Pico will be running the latest version of
MicroPython. Close the dialog and start programming!
Version to be installed: pico-20210120 (2021-01-21)
Target device location: D:\
Target device model: Raspberry Pi RP2
Starting...
Install
Cancel
OK
Cancel
Shell
Couldn't find the device automatically.
Check the connection (making sure the device is not in bootloader mode) or choose
"Configure interpreter" in the interpreter menu (bottom-right corner of the window)
to select specific port or another interpreter.
MicroPython (Raspberry Pi Pico)


Thonny - <untitled> @ 1 : 1
File Edit View Run Tools Help
Assistant
Exception
Files
Help
Help
Notes
Object inspector
Outline
Program tree
Shell
Stack
Variables
Program arguments
Plotter
Increase font size Ctrl++
Decrease font size Ctrl+-
Focus editor Alt+E
Focus shell Alt+S
<untitled
1
Shell
MicroPython v1.13-290-g556ae7914 on 2021-01-21; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
MicroPython (Raspberry Pi Pico)


Thonny - <untitled> @ 1 : 1
File Edit View Run Tools Help
Files
This computer
C: \ Users \ CoreTech
Contacts
Desktop
Documents
Downloads
Favorites
Links
Music
OneDrive
Pictures
Saved Games
Searches
Videos
Raspberry Pi Pico
<untitled>
1
Shell
MicroPython v1.13-290-g556ae7914 on 2021-01-21; Raspberry Pi Pico with RP2040
Type "help()" for more information.
MicroPython (Raspberry Pi Pico)

Thonny - <untitled> @ 8 : 14

File Edit View Run Tools Help

Save (Ctrl+S)

Files

This computer
C: \ Users \ CoreTech

- Contacts
- Desktop
- Documents
- Downloads
- Favorites
- Links
- Music
- OneDrive
- Pictures
- Saved Games
- Searches
- Videos

Raspberry Pi Pico

<untitled> *

```
1 from machine import Pin
2 from time import sleep
3
4 led = Pin(25, Pin.OUT)
5
6 while True:
7     led.toggle()
8     sleep(0.5)
```

Shell

```
ode) or choose
"Configure interpreter" in the interpreter menu (bottom-right corne
r of the window)
to select specific port or another interpreter.

MicroPython v1.13-290-g556ae7914 on 2021-01-21; Raspberry Pi Pico with
RP2040
Type "help()" for more information.
>>> print('hello world')
 hello world
>>>
```

MicroPython (Raspberry Pi Pico)

Thonny - <untitled> @ 8 : 14

File Edit View Run Tools Help

Files

This computer
C: \ Users \ CoreTech

- Contacts
- Desktop
- Documents
- Downloads
- Favorites
- Links
- Music
- OneDrive
- Pictures
- Saved Games
- Searches
- Videos

Raspberry Pi Pico

<untitled> *

```
1 from machine import Pin
2 from time import sleep
3
4 led = Pin(25, Pin.OUT)
5
6 while True:
7     led.toggle()
8     sleep(0.5)
```

Shell

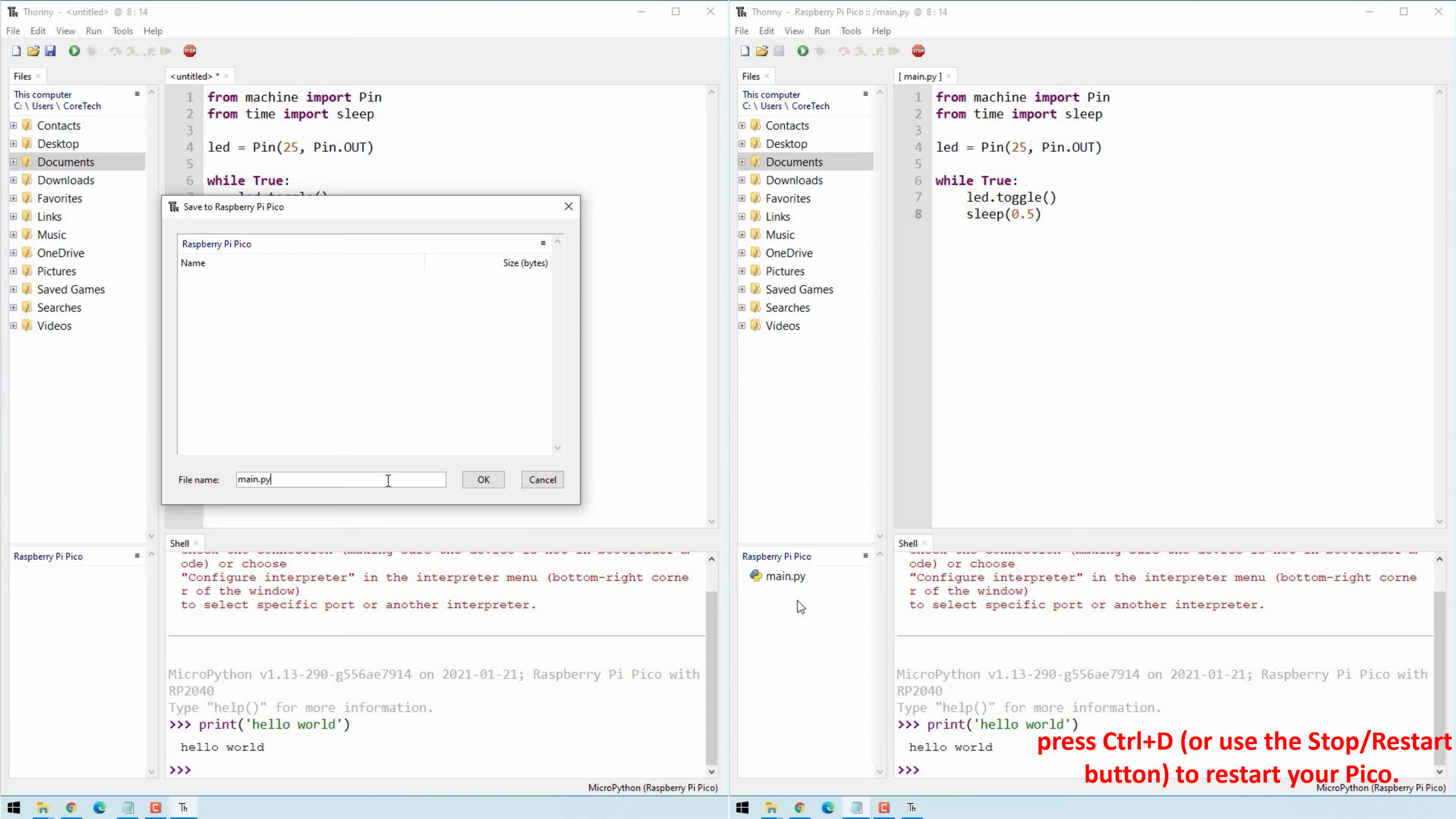
```
ode) or choose
"Configure interpreter" in the interpreter menu (bottom-right corne
r of the window)
to select specific port or another interpreter.

MicroPython v1.13-290-g556ae7914 on 2021-01-21; Raspberry Pi Pico with
RP2040
Type "help()" for more information.
>>> print('hello world')
 hello world
>>>
```

MicroPython (Raspberry Pi Pico)

Where to save to?

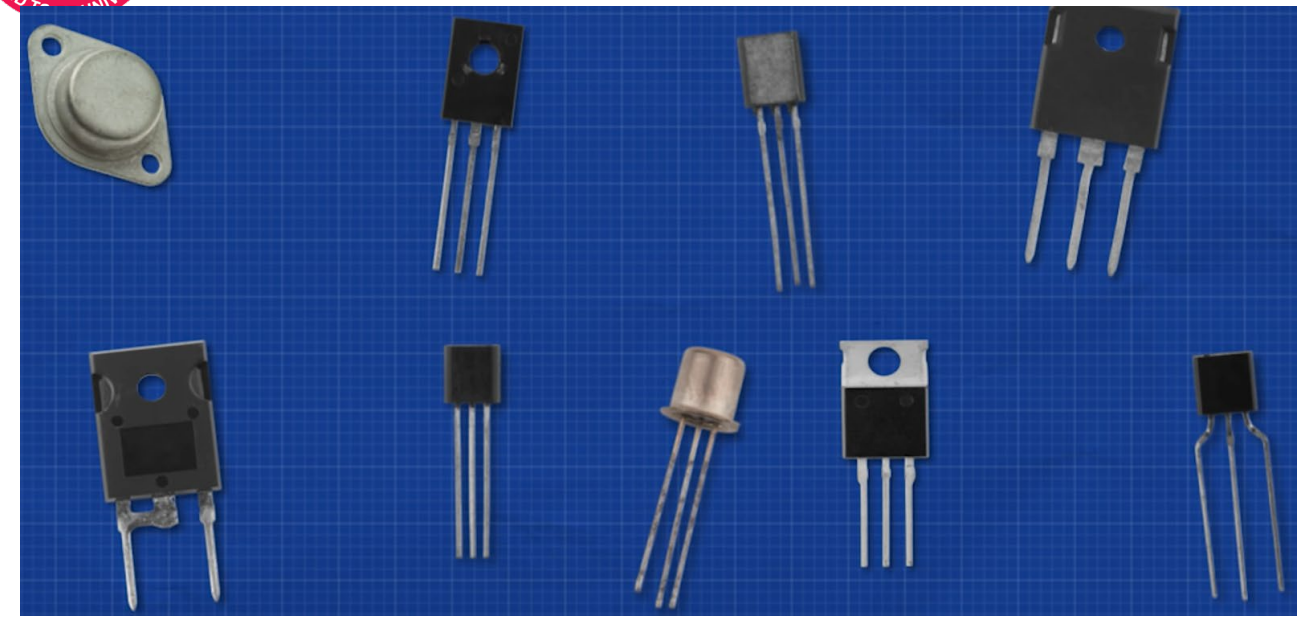
This computer

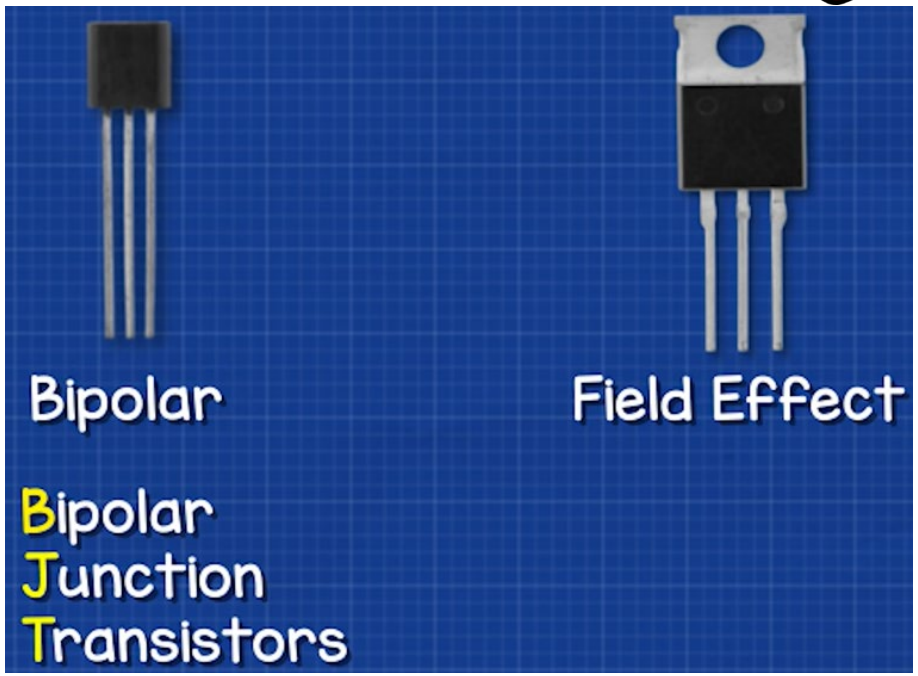
Raspberry Pi Pico

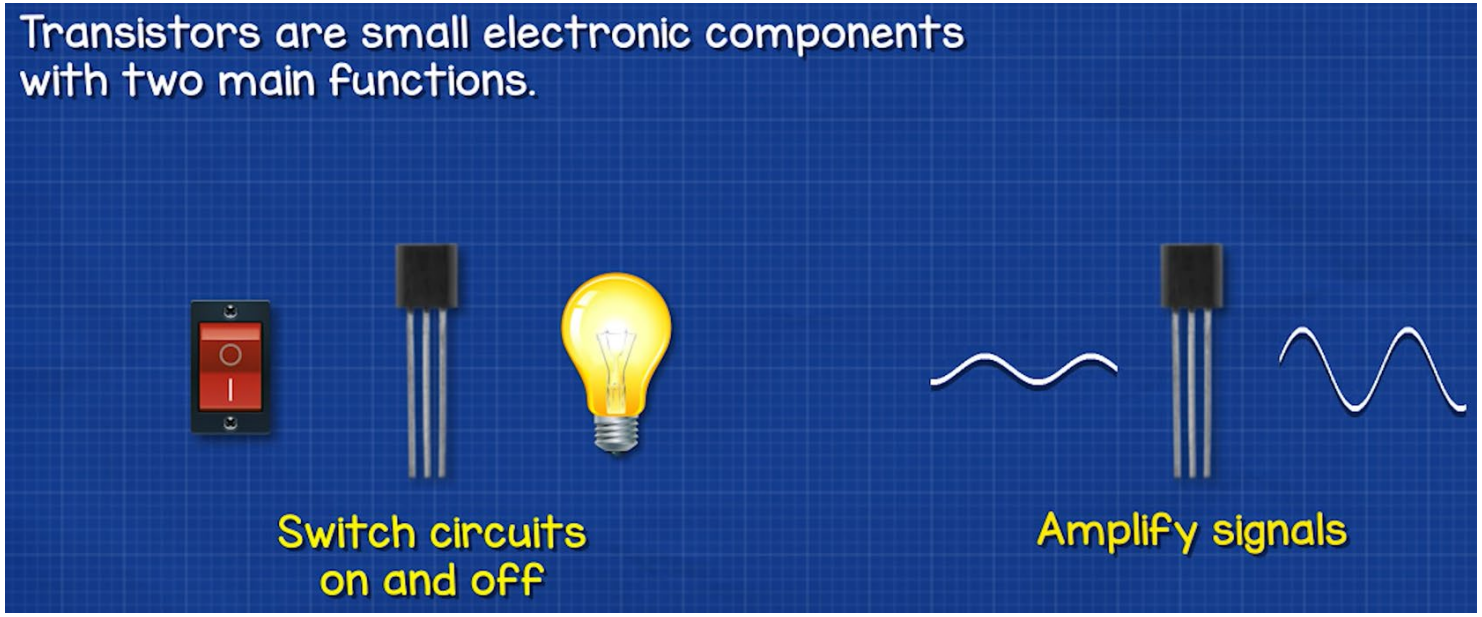

Thonny - <untitled> @ 8 : 14
File Edit View Run Tools Help
Files
<untitled> *
This computer
C: \ Users \ CoreTech
Contacts
Desktop
Documents
Downloads
Favorites
Links
Music
OneDrive
Pictures
Saved Games
Searches
Videos
1 from machine import Pin
2 from time import sleep
3
4 led = Pin(25, Pin.OUT)
5
6 while True:
Save to Raspberry Pi Pico
Raspberry Pi Pico
Name
Size (bytes)
File name: main.py
OK
Cancel
Shell
ode) or choose
"Configure interpreter" in the interpreter menu (bottom-right corne
r of the window)
to select specific port or another interpreter.
MicroPython v1.13-290-g556ae7914 on 2021-01-21; Raspberry Pi Pico with
RP2040
Type "help()" for more information.
>>> print('hello world')
hello world
>>>
MicroPython (Raspberry Pi Pico)
Thonny - Raspberry Pi Pico :: /main.py @ 8 : 14
File Edit View Run Tools Help
Files
[main.py]
This computer
C: \ Users \ CoreTech
Contacts
Desktop
Documents
Downloads
Favorites
Links
Music
OneDrive
Pictures
Saved Games
Searches
Videos
1 from machine import Pin
2 from time import sleep
3
4 led = Pin(25, Pin.OUT)
5
6 while True:
7 led.toggle()
8 sleep(0.5)
Raspberry Pi Pico
main.py
Shell
ode) or choose
"Configure interpreter" in the interpreter menu (bottom-right corne
r of the window)
to select specific port or another interpreter.
MicroPython v1.13-290-g556ae7914 on 2021-01-21; Raspberry Pi Pico with
RP2040
Type "help()" for more information.
>>> print('hello world')
hello world
>>>
press Ctrl+D (or use the Stop/Restart
button) to restart your Pico.
MicroPython (Raspberry Pi Pico)

TRANSISTORS

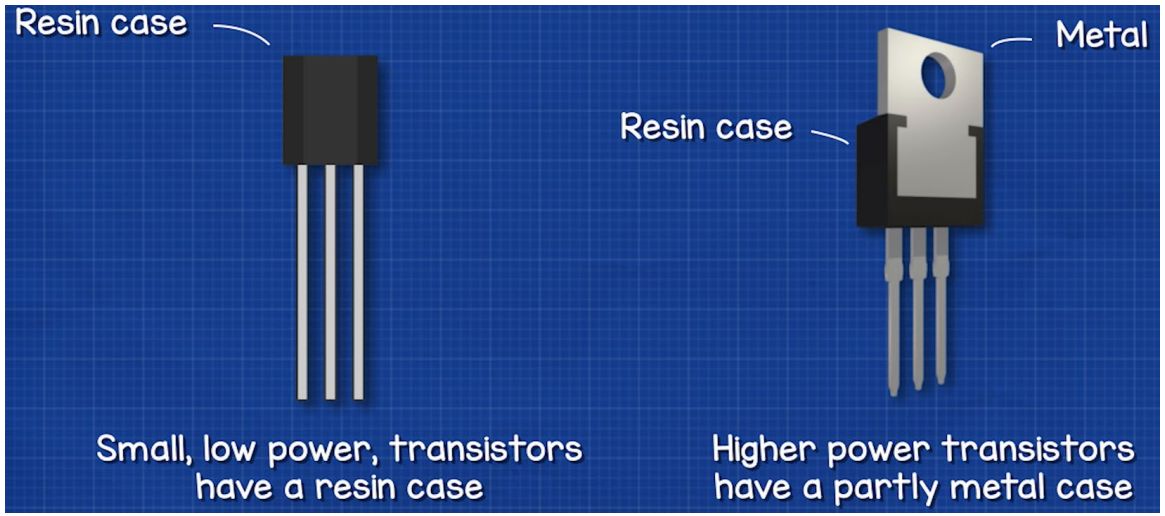
Transistors

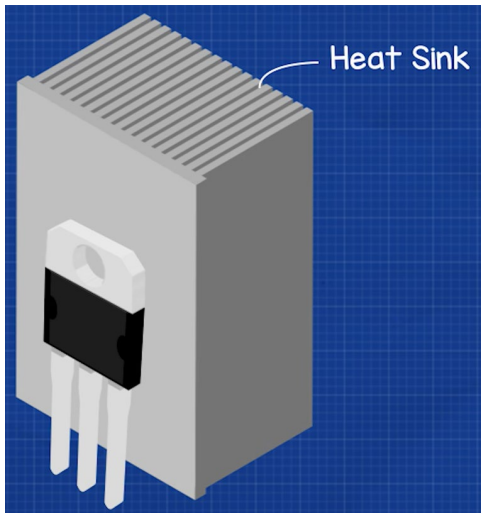


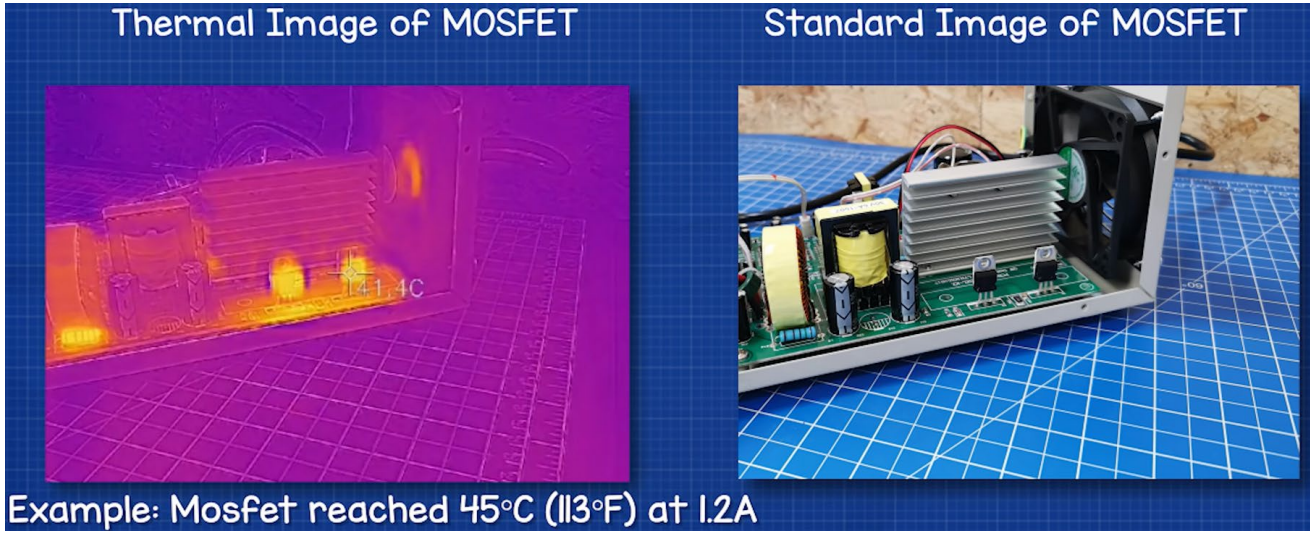


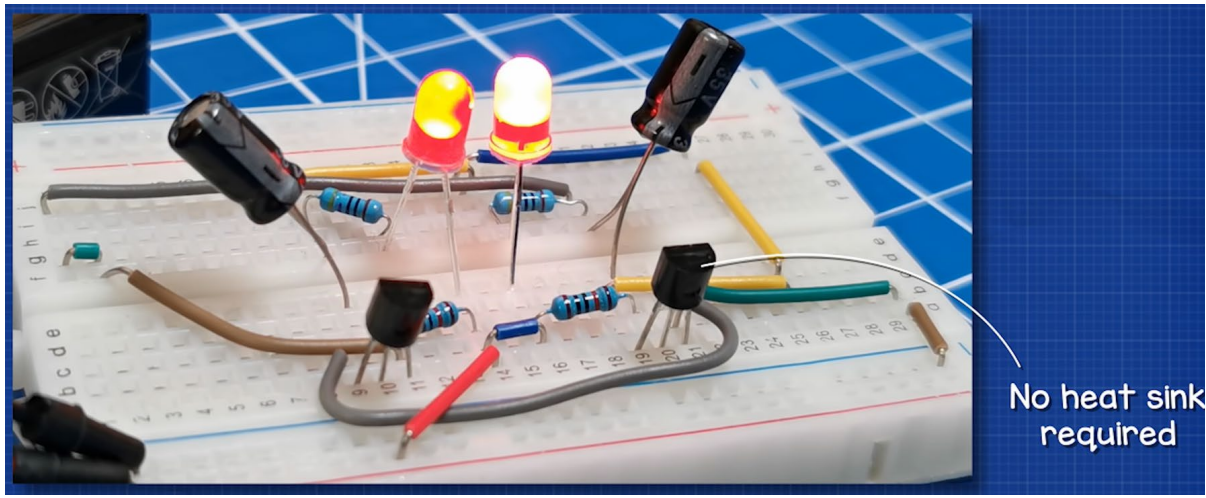


Transistors

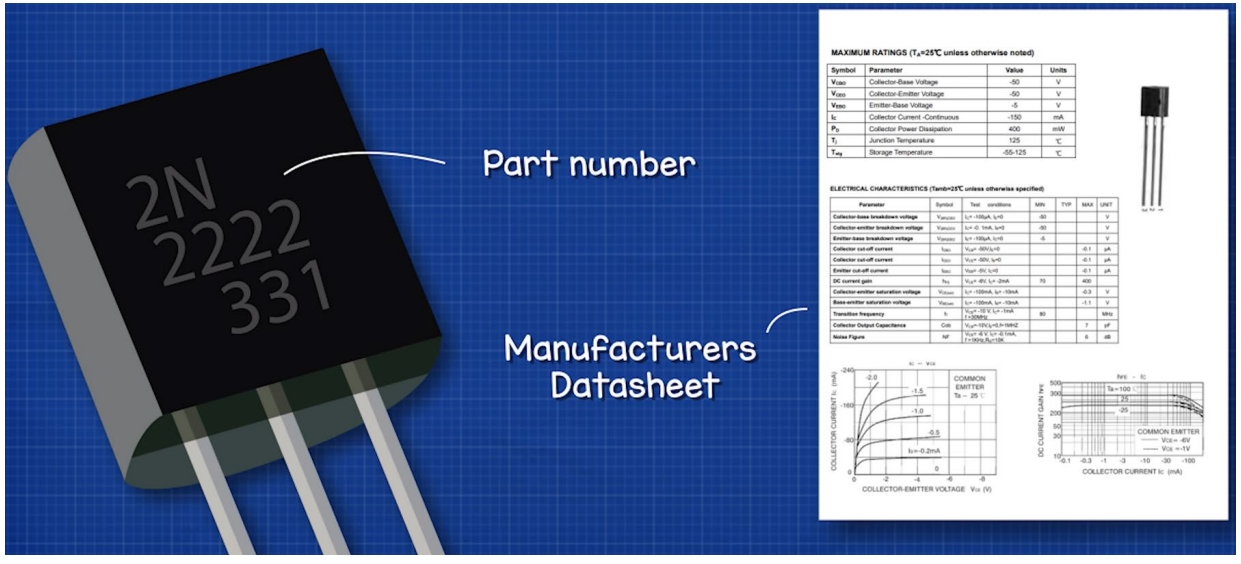


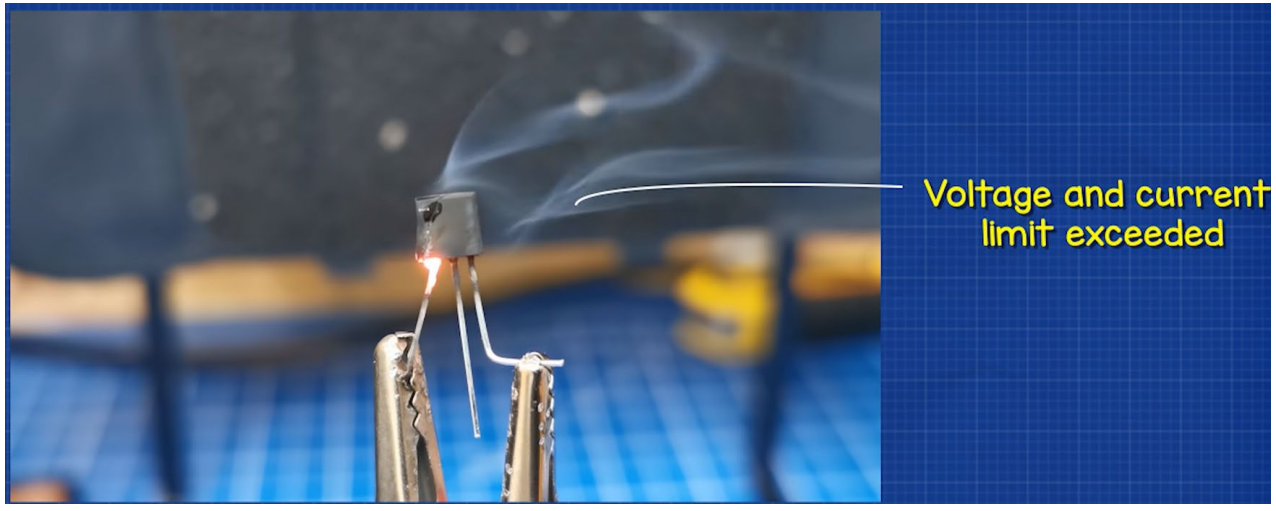


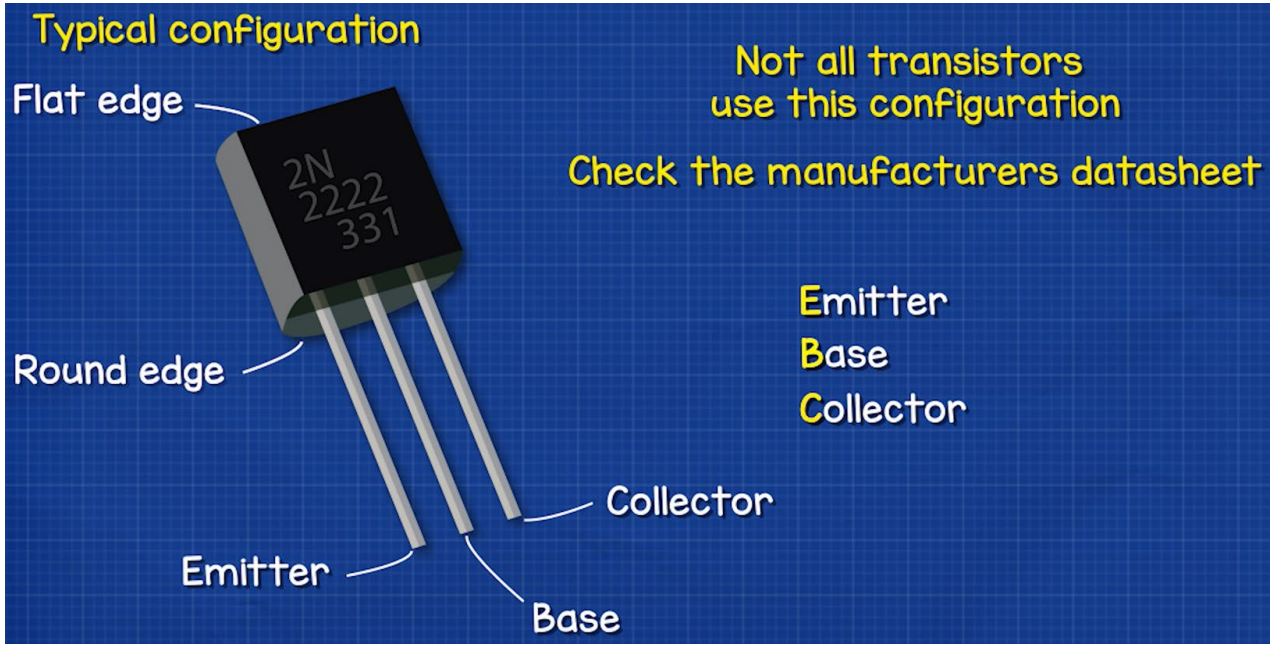




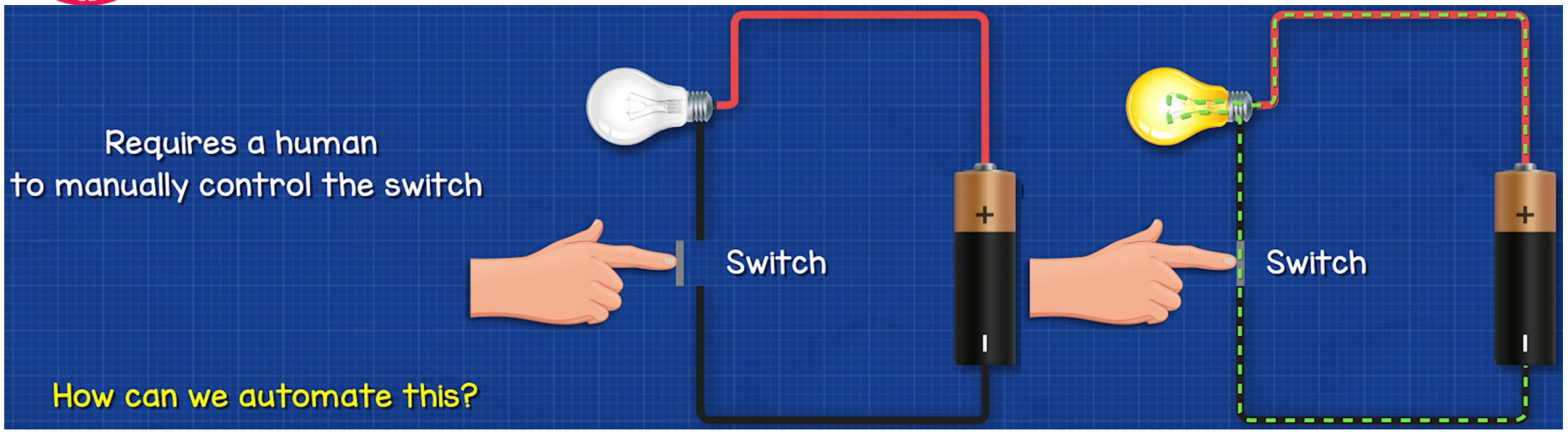
Transistors

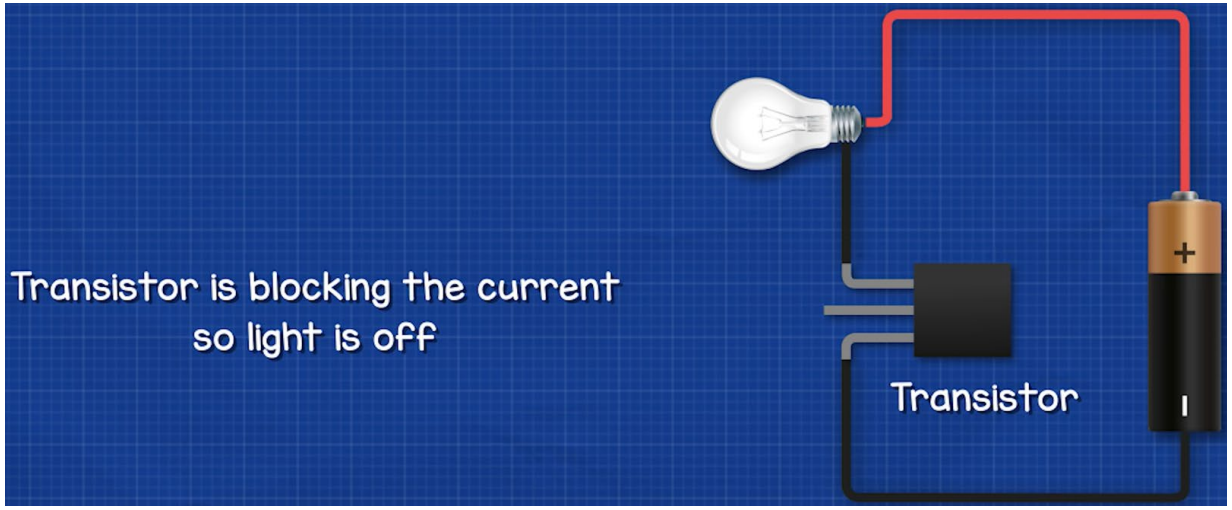


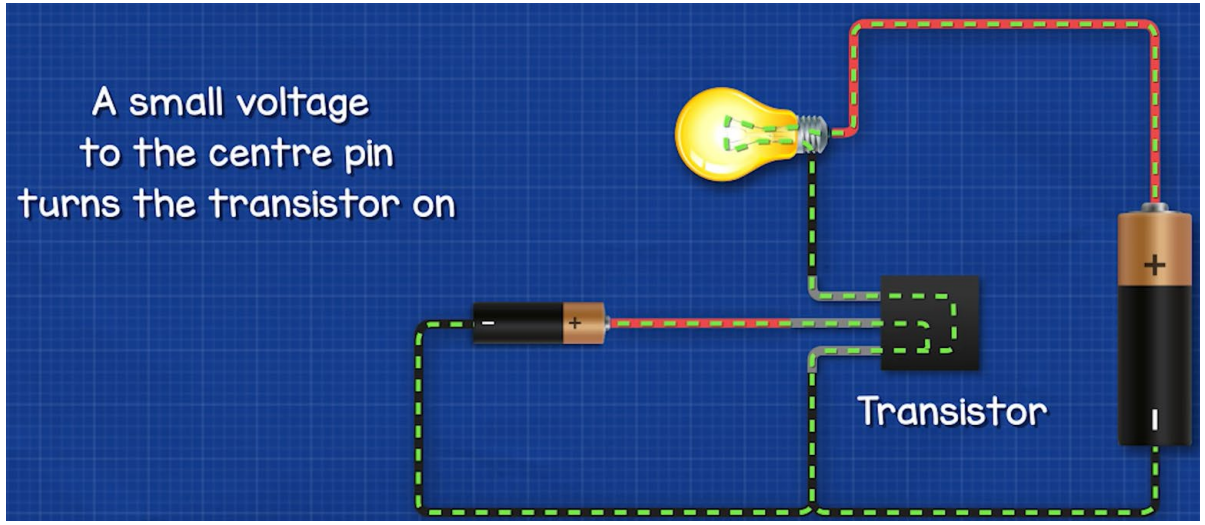




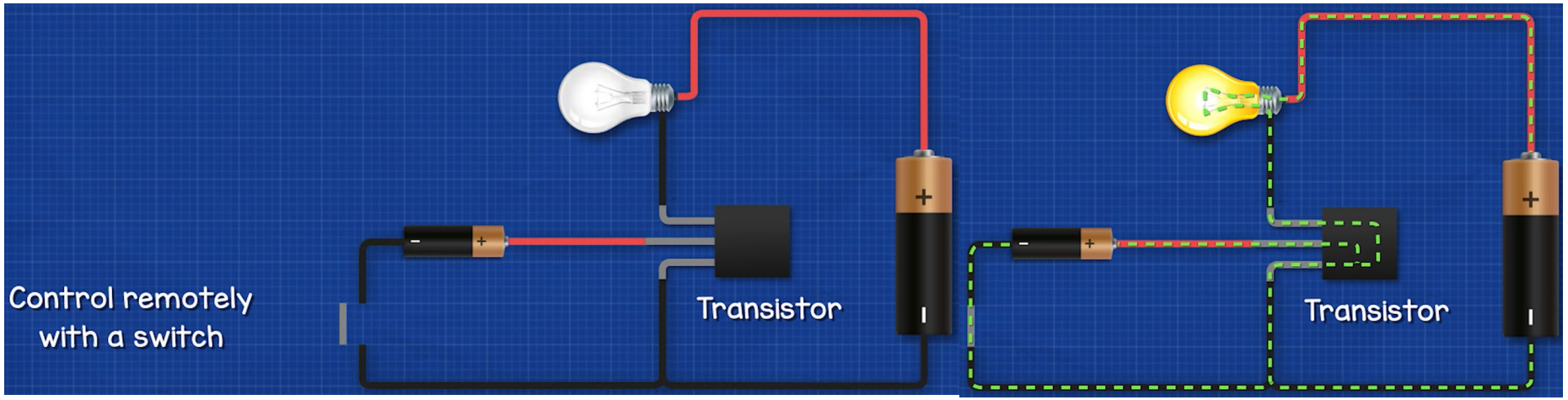
Transistors Application

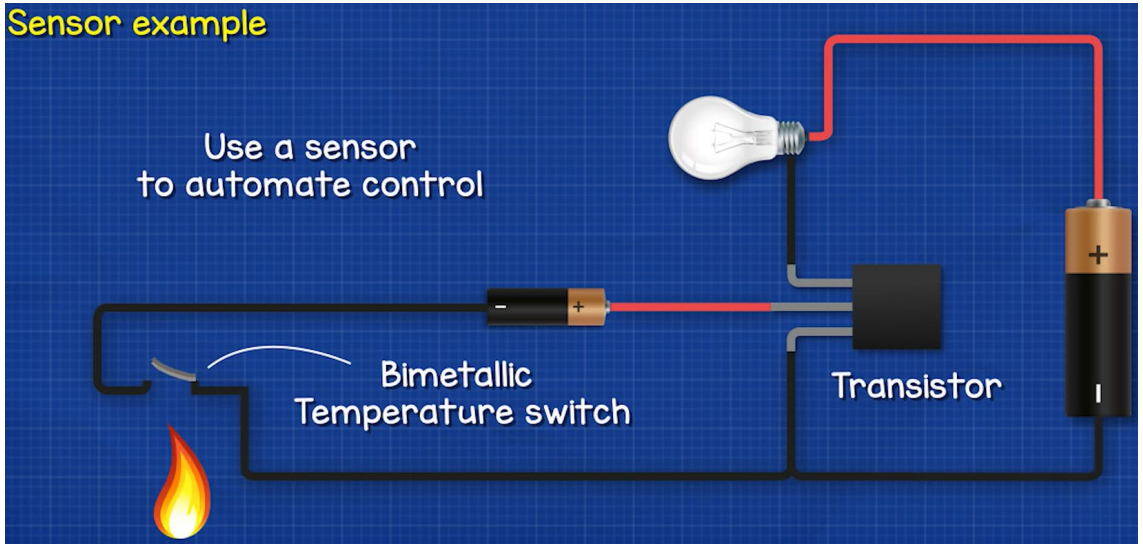


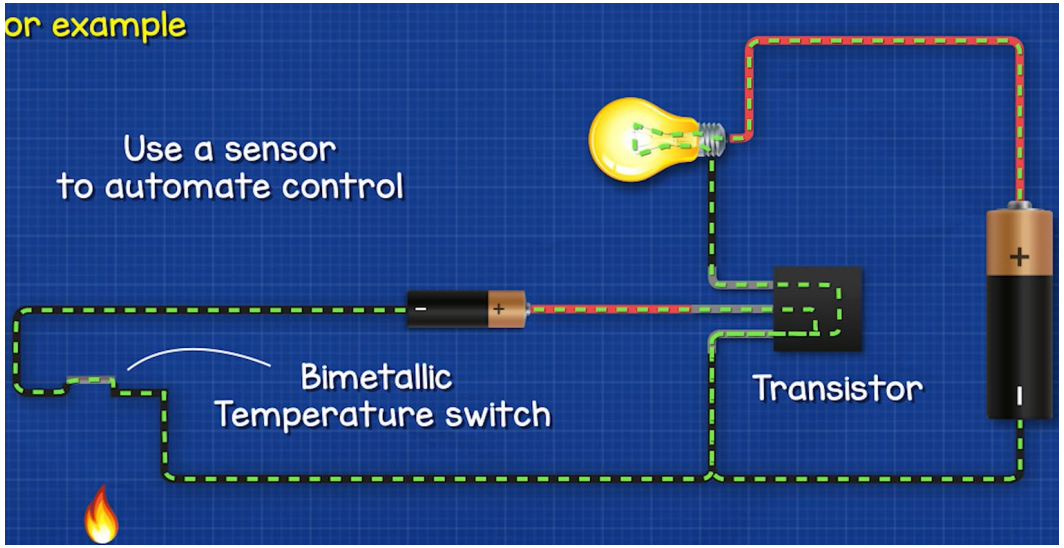




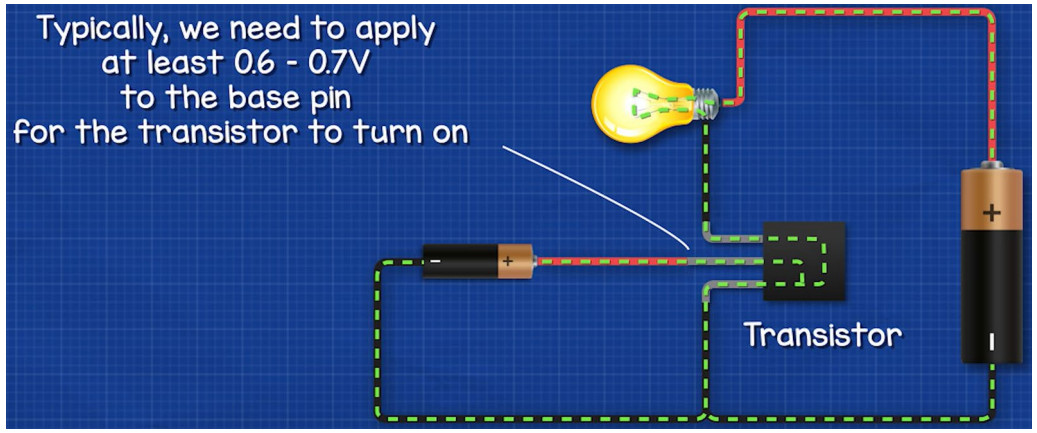
Transistors Application

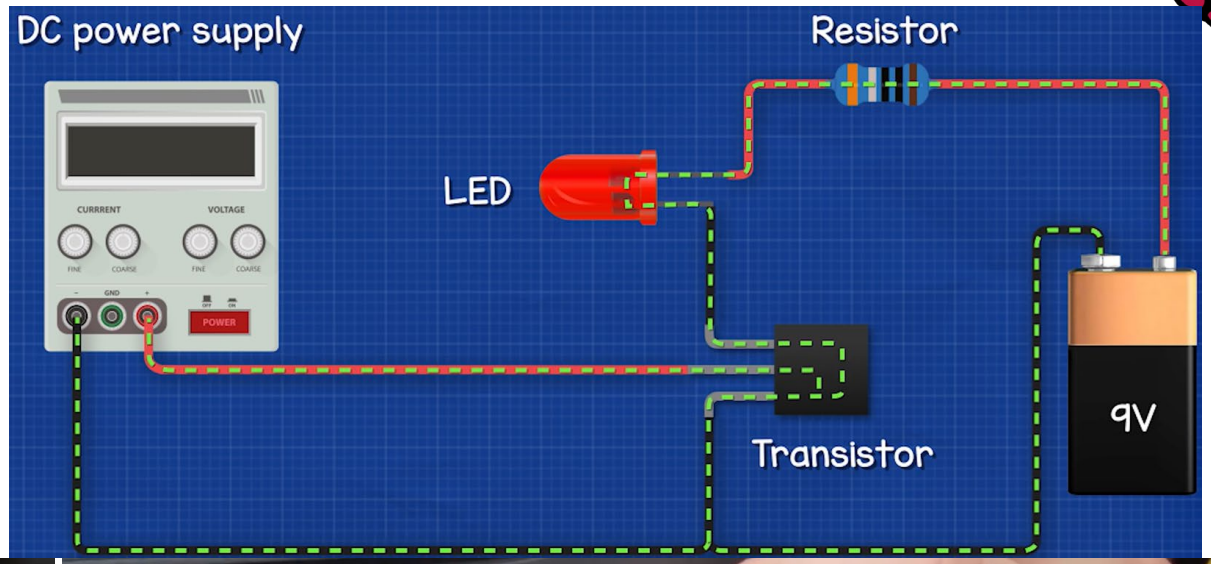


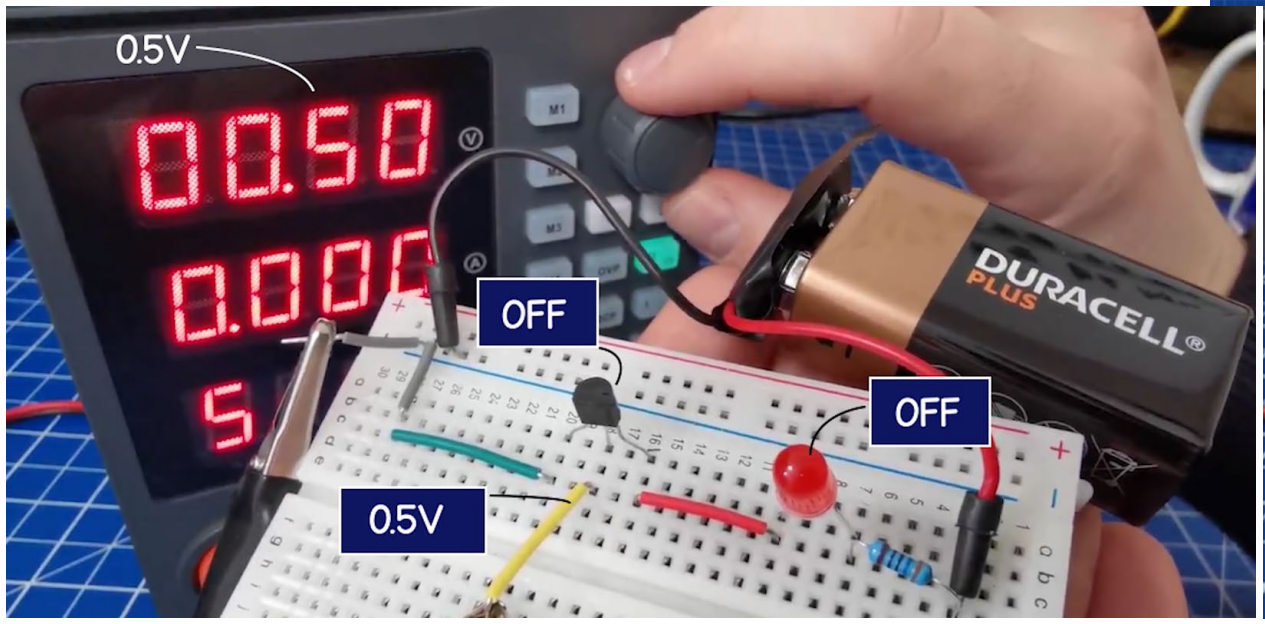


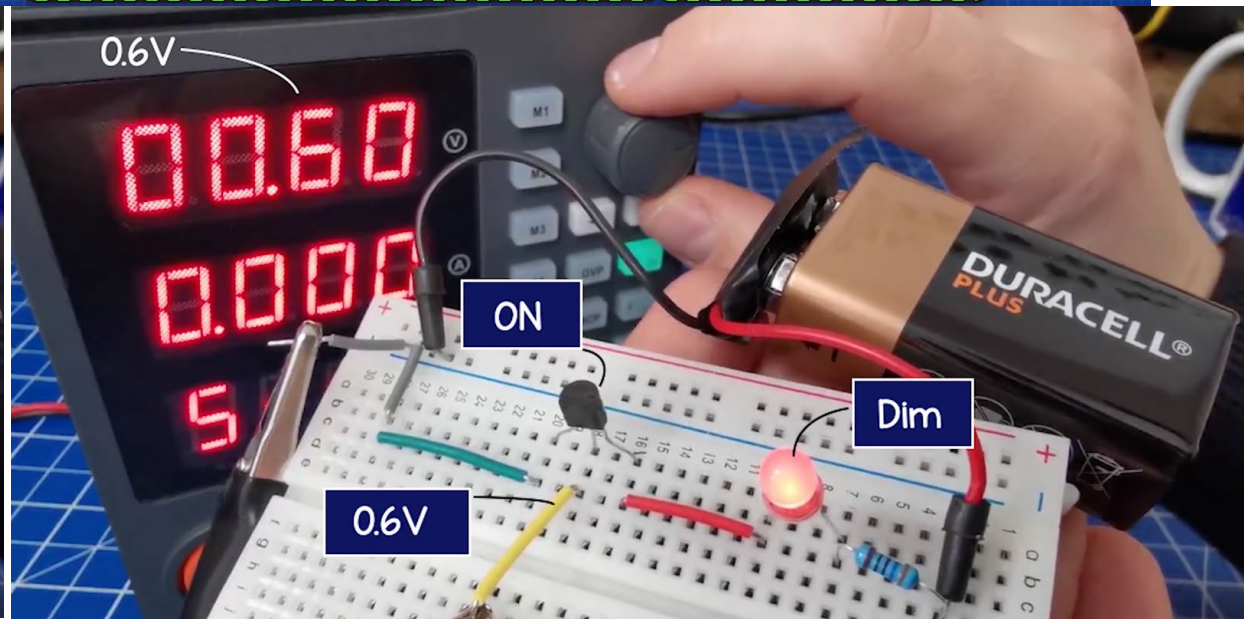


Transistors Application

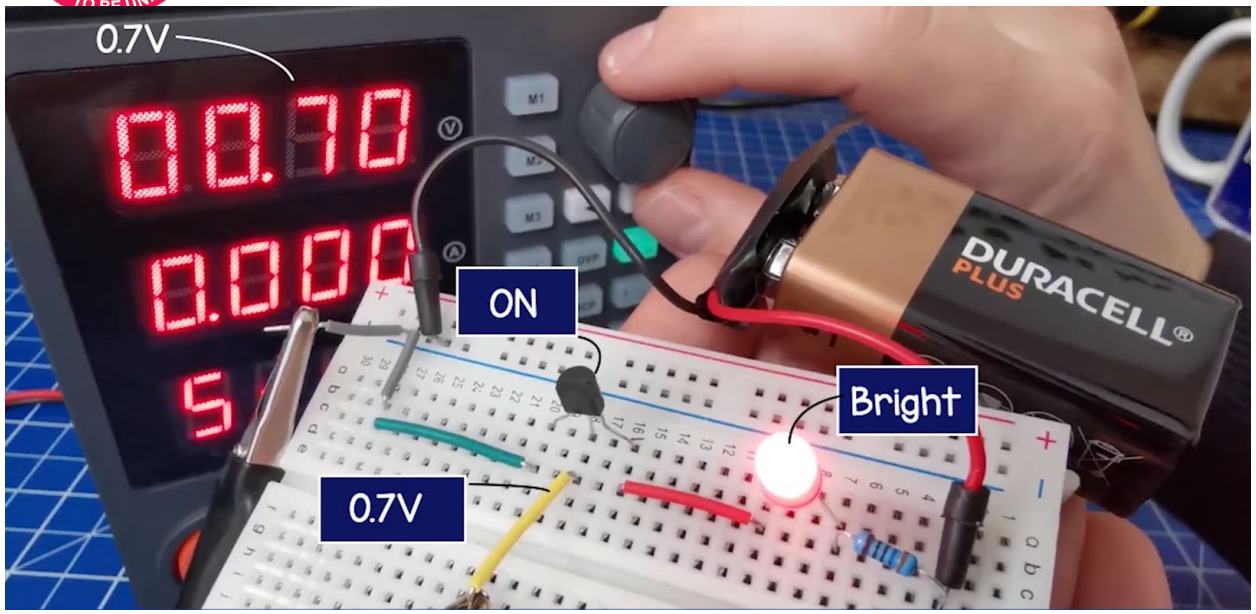


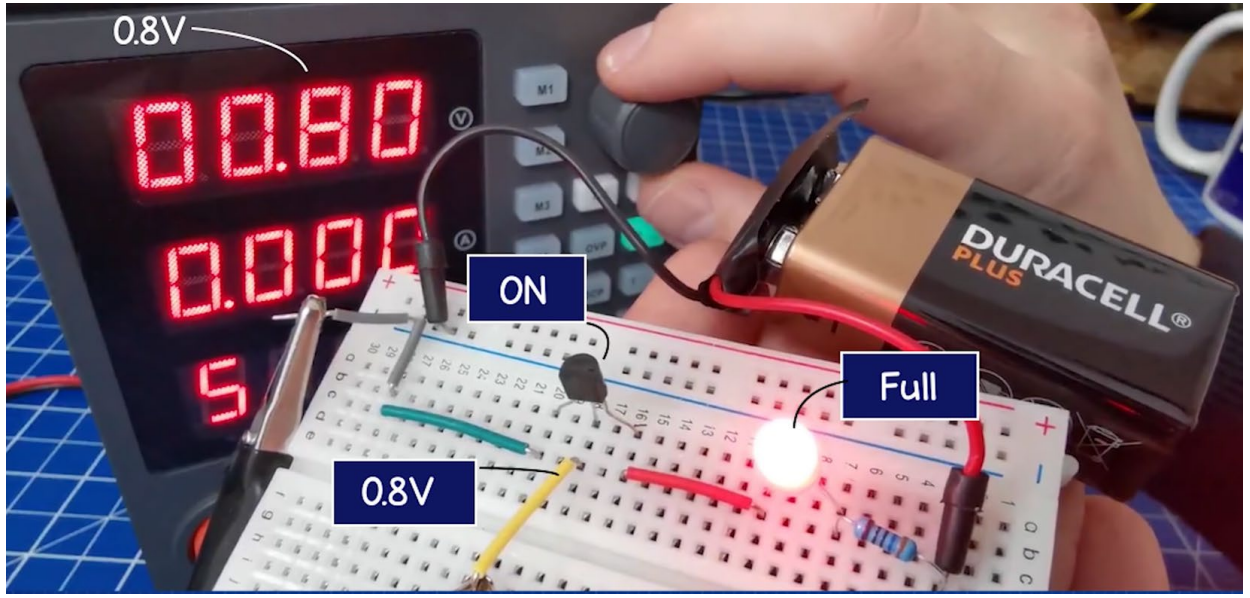


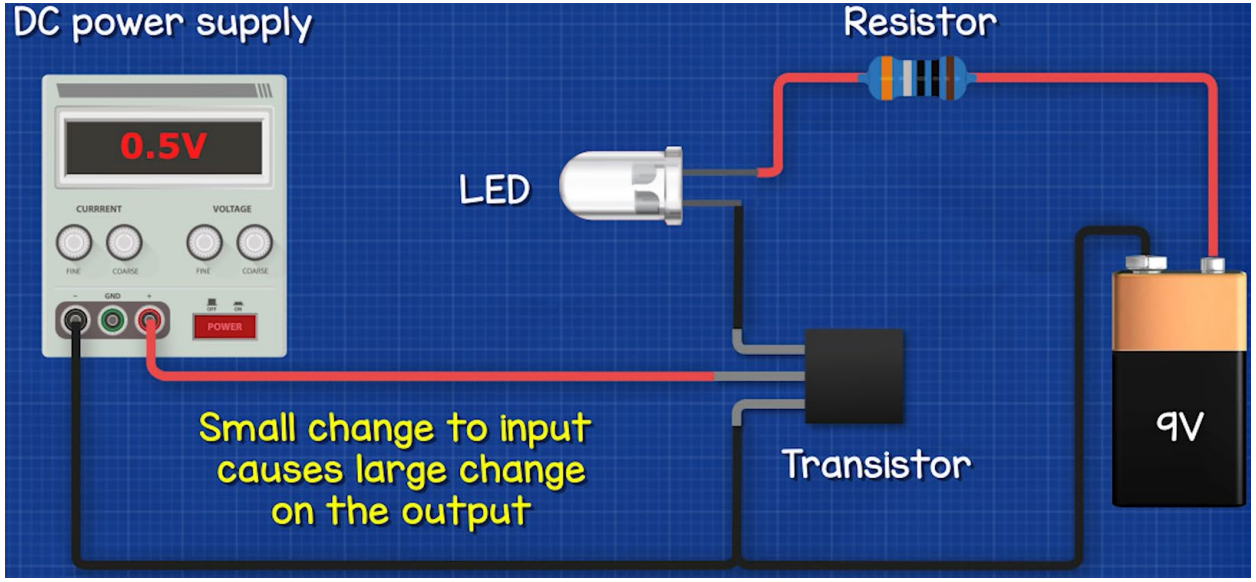


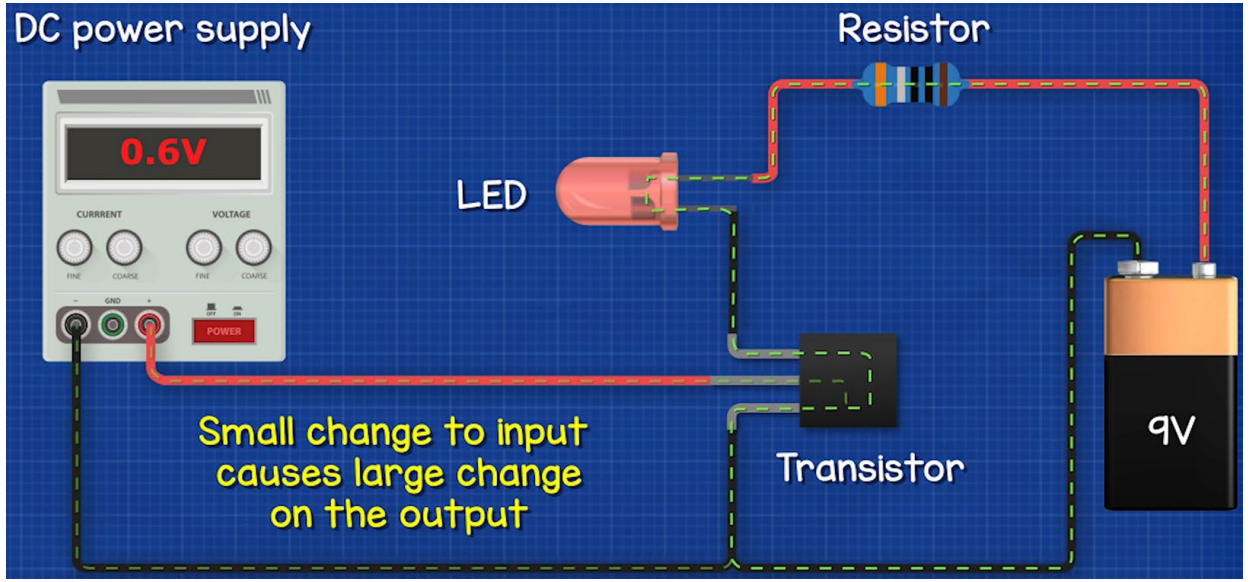


Transistors Application

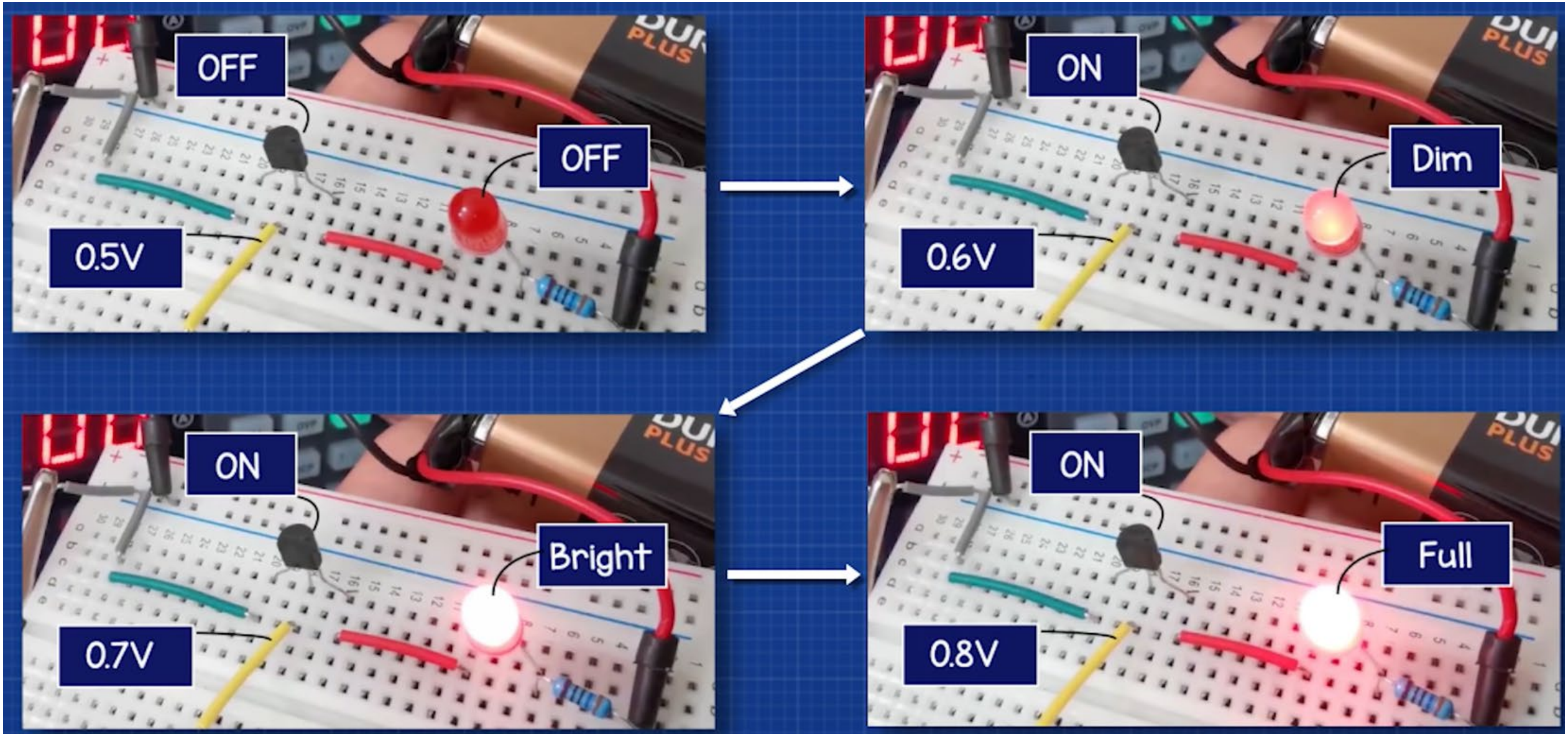




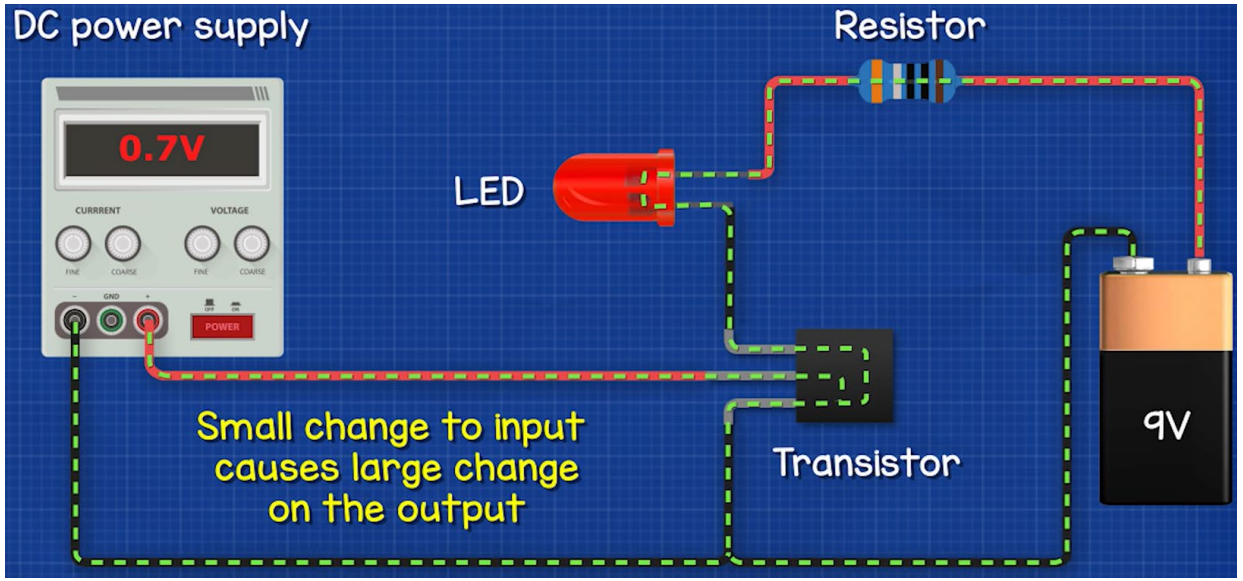


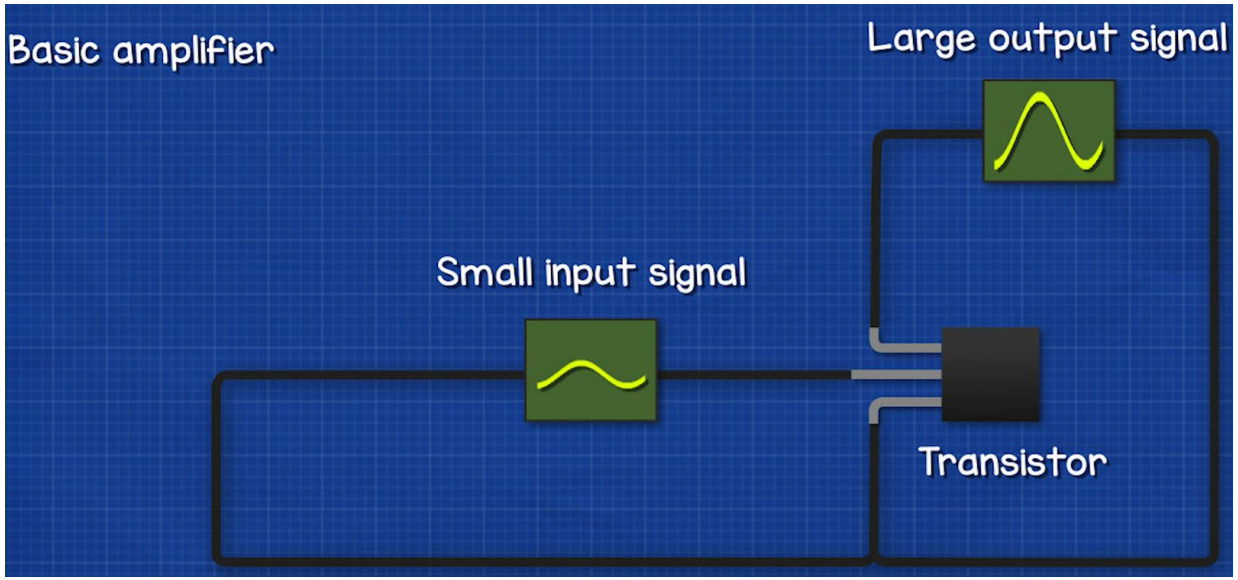


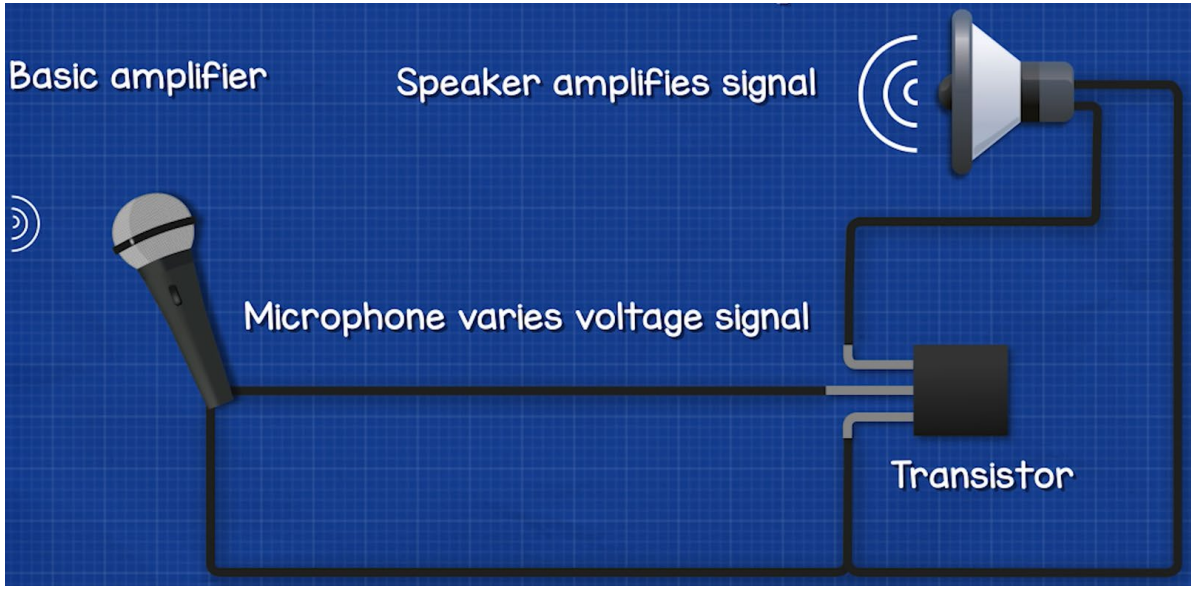
Transistors Application



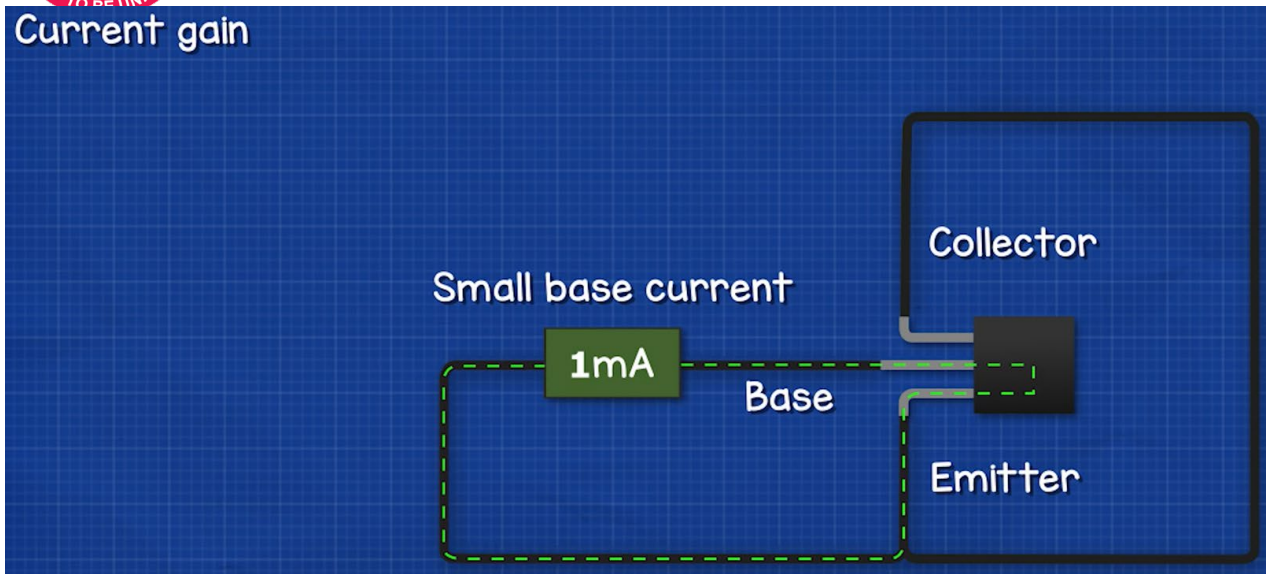
Transistors Application

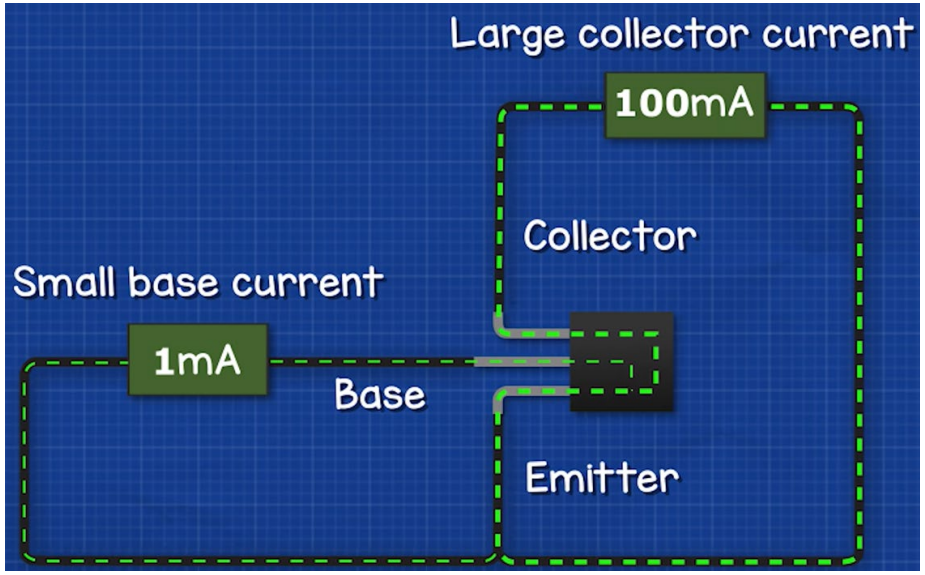






Transistors Current Gain

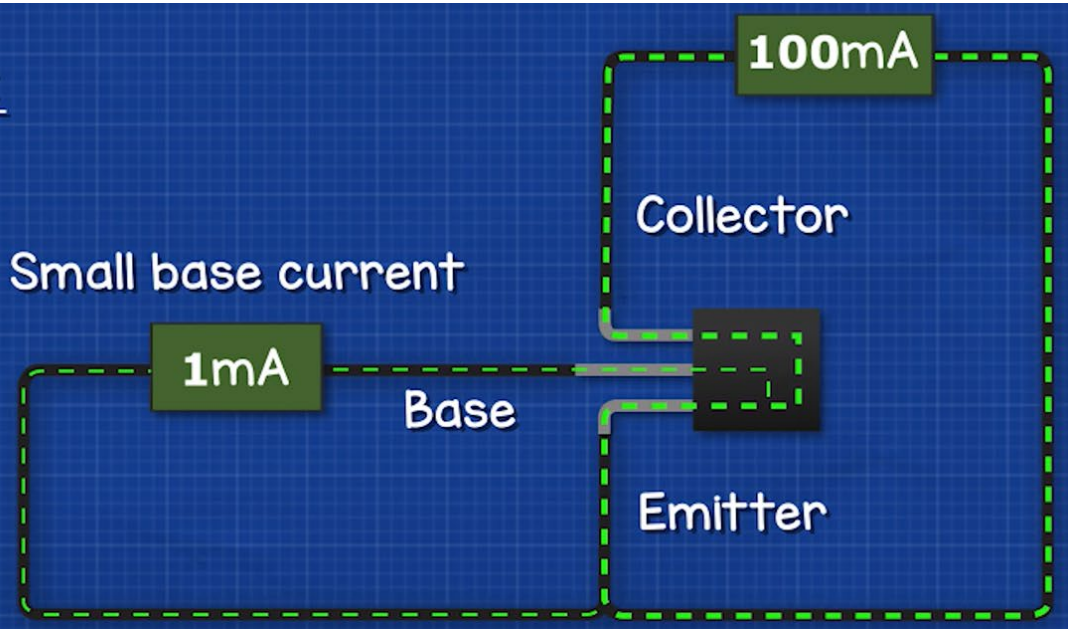




$$\beta = \frac{\text{Collector current}}{\text{Base current}}$$

$$\beta = \frac{100\text{mA}}{1\text{mA}}$$

$$\beta = 100$$

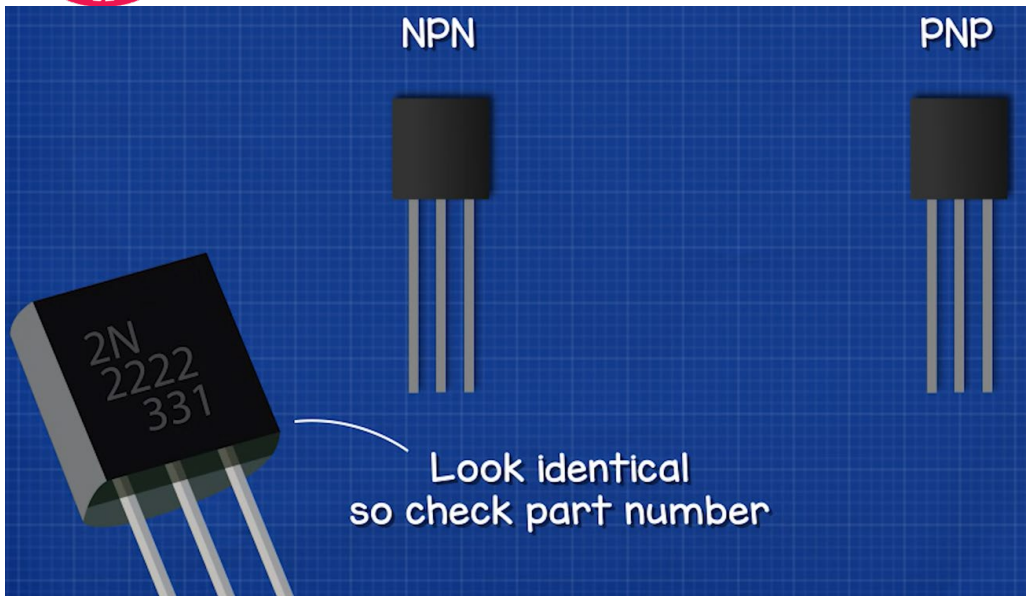


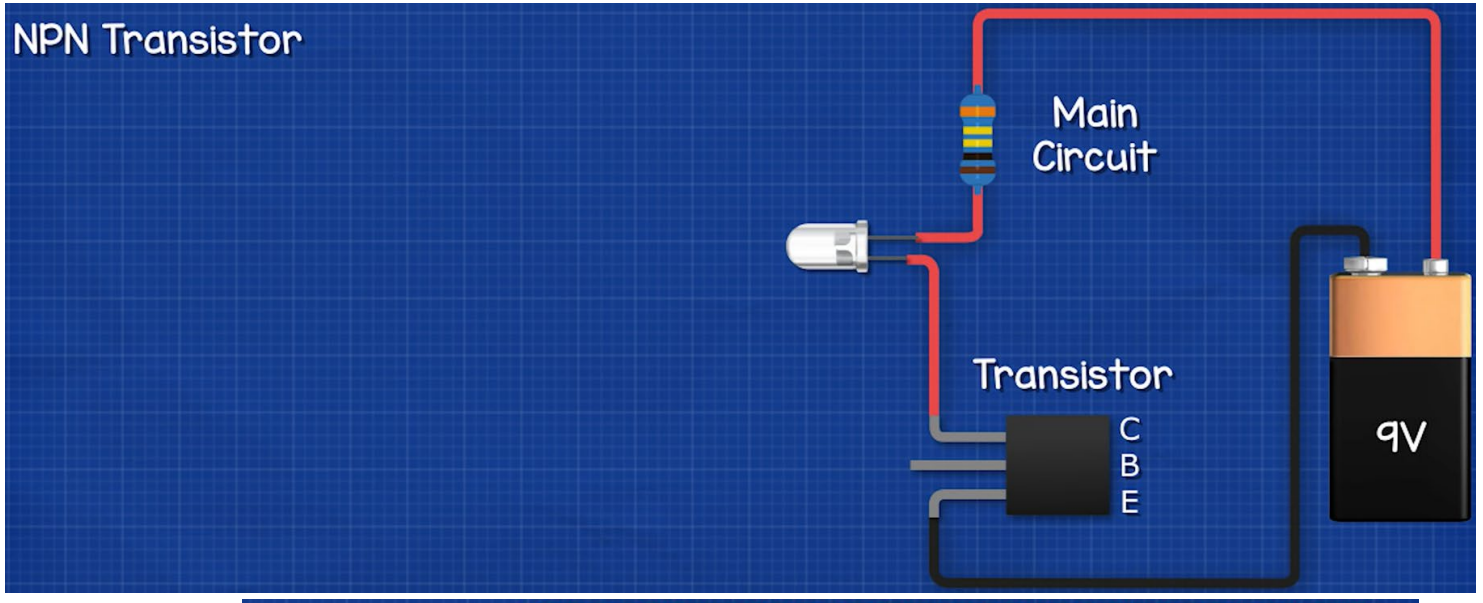
$$\beta = \frac{\text{Collector current}}{\text{Base current}}$$

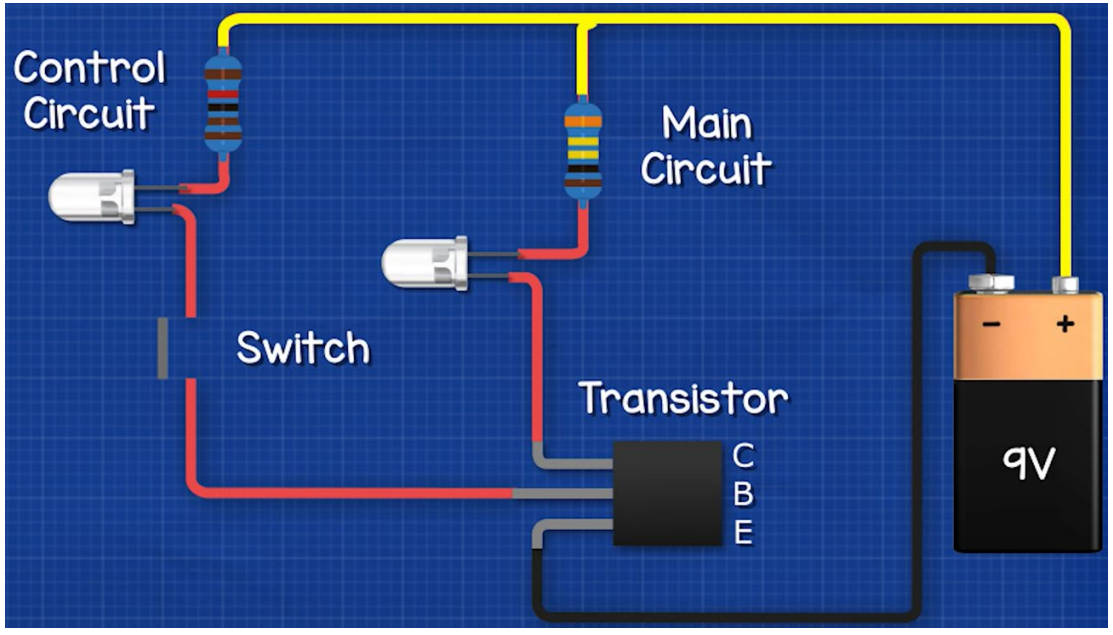
$$\text{Collector current} = \beta \times \text{Base current}$$

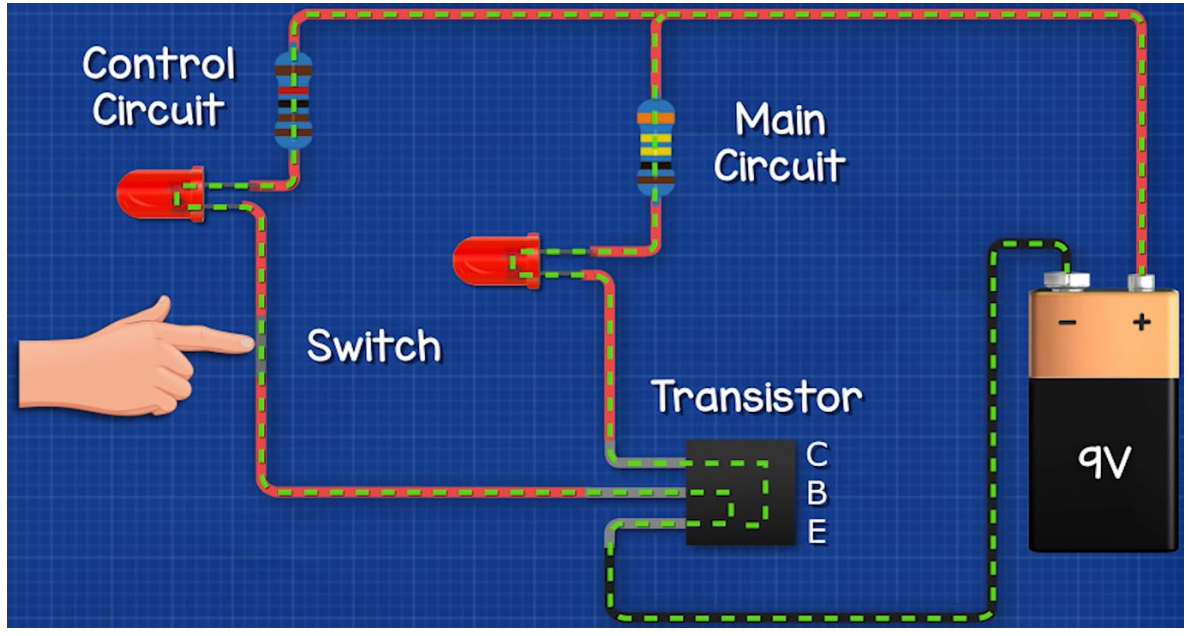
$$\text{Base current} = \frac{\text{Collector current}}{\beta}$$

NPN and PNP Transistor

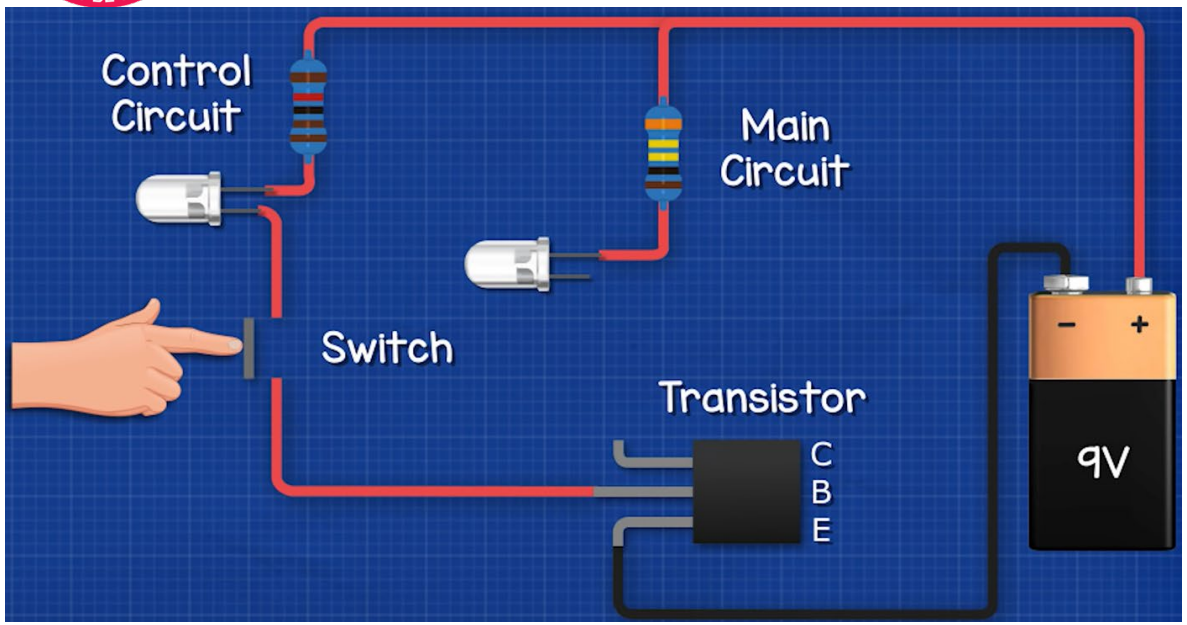


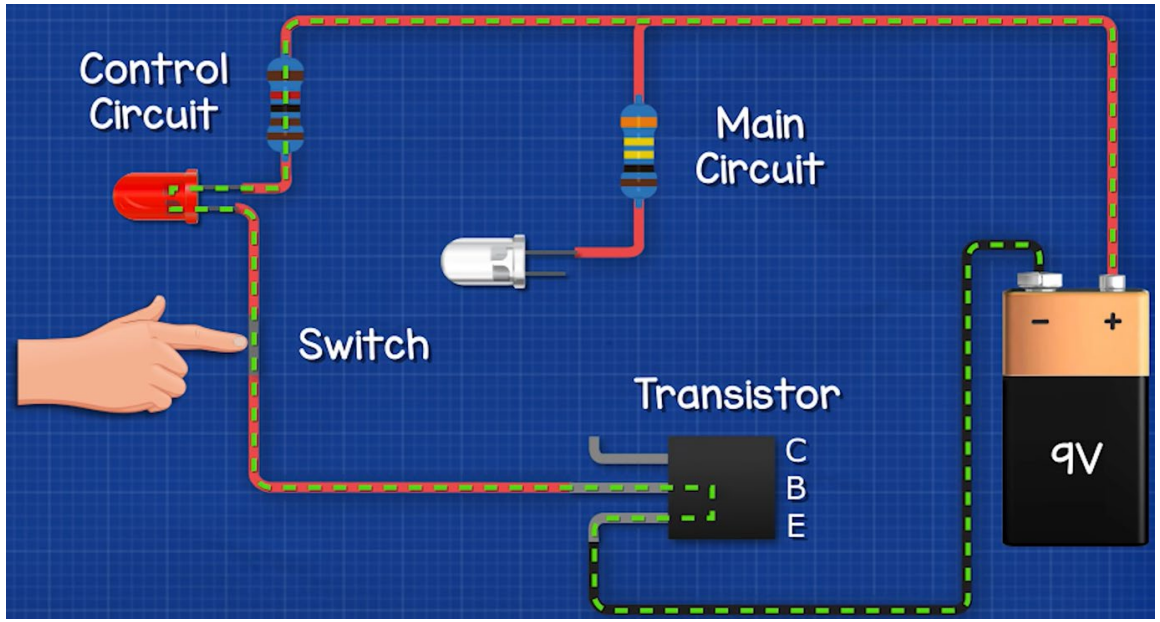


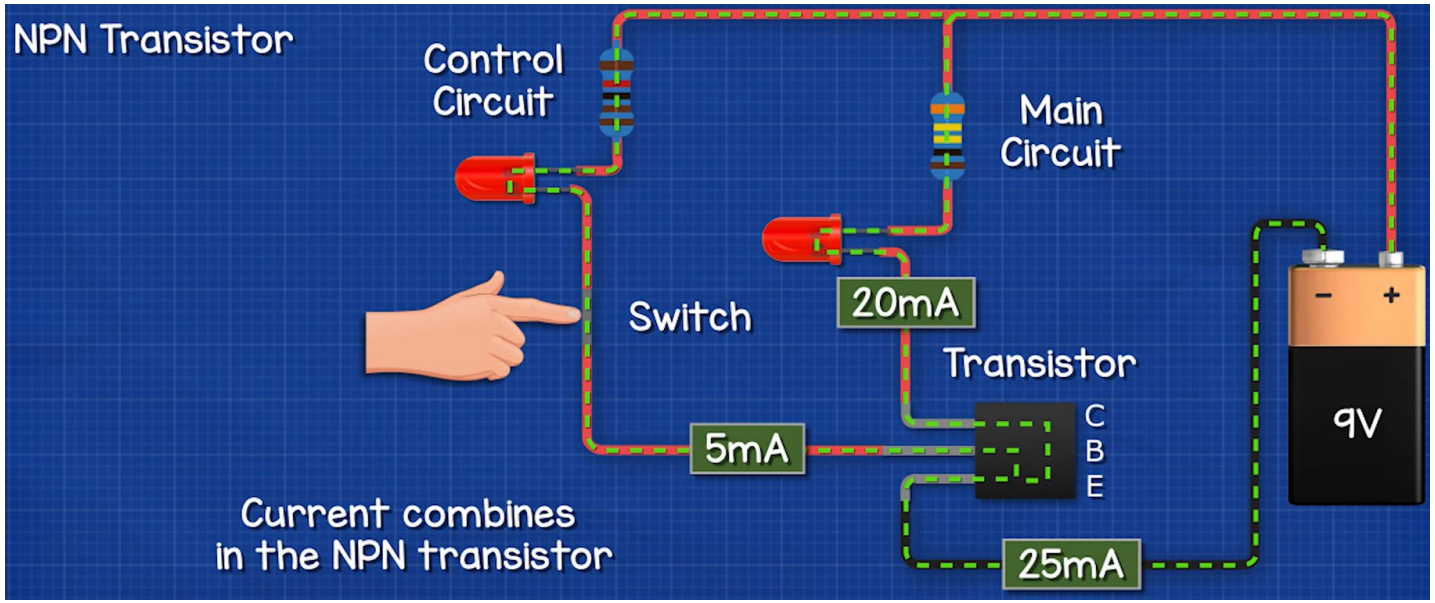




NPN and PNP Transistor







PNP Transistor