

PYTHON

BASIC COURSE

ELECTRONICS

BASIC COURSE



Teachers

GABRIELA OCHOA-DĄDERSKA
KACPER KOTLEWSKI

Python : programming language

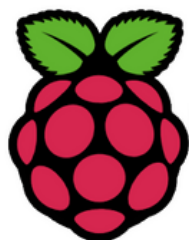
Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Electronics: Raspberry Pi Pico CircuitPython

Raspberry Pi Pico is a tiny, fast, and versatile board built using RP2040, a brand new microcontroller chip designed by Raspberry Pi in the UK. CircuitPython is a programming language designed to simplify experimenting and learning to code on low-cost microcontroller boards.



CIRCUITPYTHON



RaspberryPi



python™

PYTHON

Python code

= symbols set variable value

then we separate in `print` the sting with `"Your name is "`

variable name that is called `name` in this case

as a value string

`"Kristian"`

and `add` to this string a value of variable called `name`

```

name = "Kristian"
print("Your name is " + name)
Your name is Kristian
    
```

as you see output is `Your name is Kristian`

```

name = "Kristian"
age = "35"
print("There once was a man named " + name + ",")
print("he was " + age + " years old.")
print("He really liked the name", name + ",")
print("but didn't like being", age + ".")
There once was a man named Kristian,
he was 35 years old.
He really liked the name Kristian,
but didn't like being 35.
    
```

In case of adding strings in `print`, works similar to `+` except `+` don't create space automatically, but `,` creates one

spaced not spaced

PYTHON

Python code

```
[ ] a=2
    b=5
    if a>b:
        print("a is greater than b")
    else:
        print("not true")
```

not true

```
[ ] a=3
    b=9
    if a>b:
        print("a is greater than b")
    elif a == b:
        print("a and b are equal")
    else:
        print("hello word")
```

hello word

Python supports the usual logical conditions from mathematics:

- Equals: `a == b`
- Not Equals: `a != b`
- Less than: `a < b`
- Less than or equal to: `a <= b`
- Greater than: `a > b`
- Greater than or equal to: `a >= b`

These conditions can be used in several ways, most commonly in "if statements" and loops.

PYTHON

In Python, Using a for loop with `range()`, we can repeat an action a specific number of times. For example, let's see how to use the `range()` function of Python 3 to produce the first five numbers

```
▶ for i in range(5):  
    print(i)
```

```
↳ 0  
   1  
   2  
   3  
   4
```

Error in Python

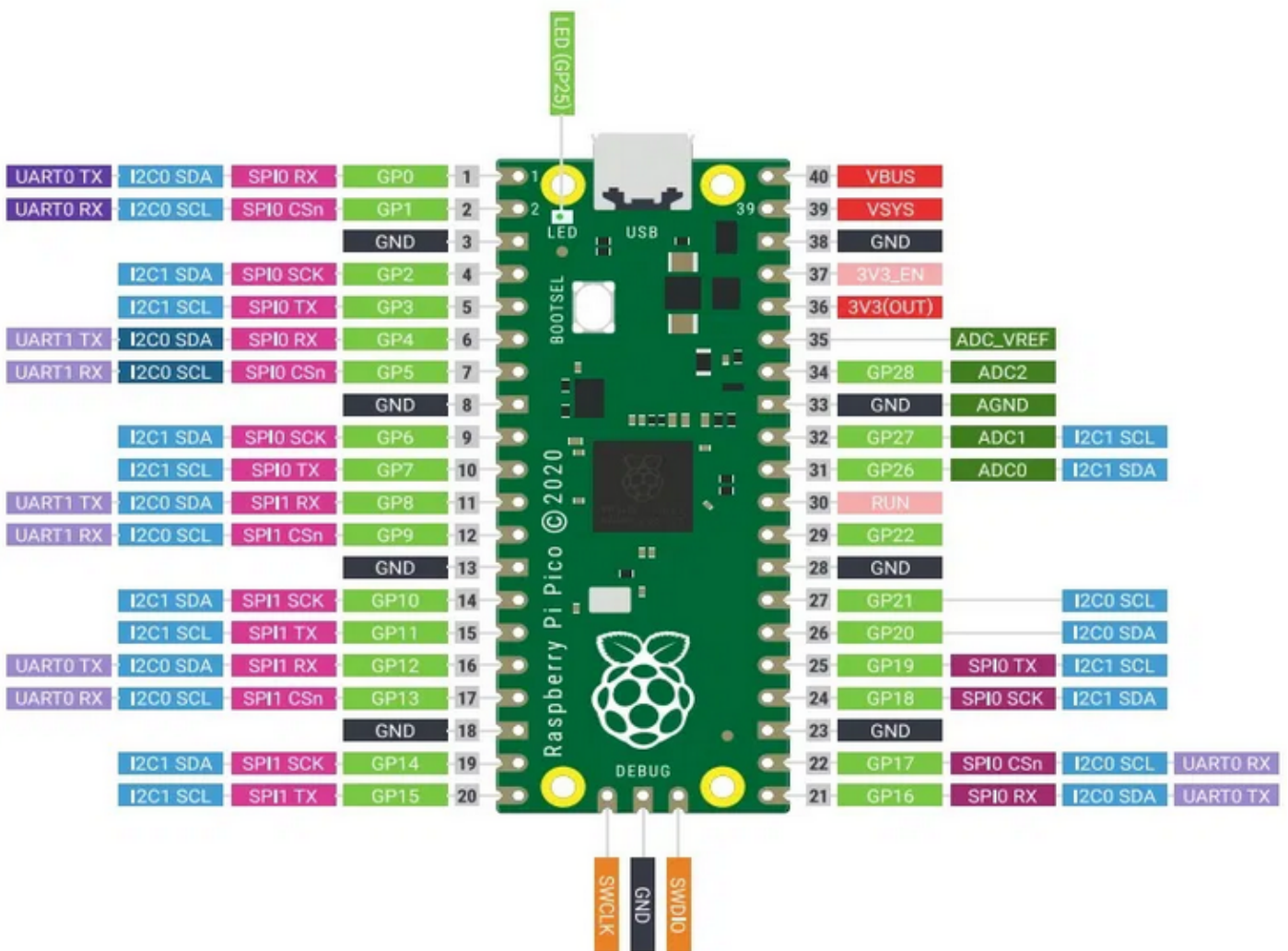
```
▶ for i in range("HaLO"):  
    print(i)
```

```
-----  
TypeError                                 Traceback (most recent call last)  
<ipython-input-9-1a3d3a55b9dc> in <module>()  
----> 1 for i in range("HaLO"):  
      2     print(i)
```

```
TypeError: 'str' object cannot be interpreted as an integer
```

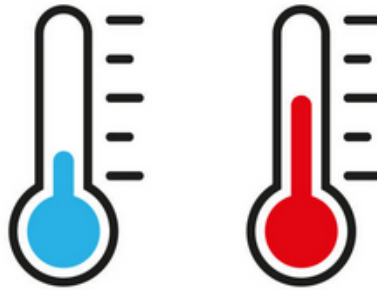
Electronics

Raspberry Pi Pico



Electronics

Read temperature



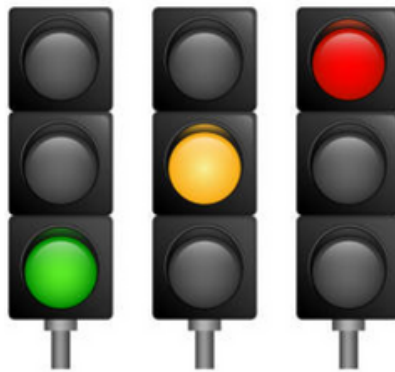
```
File Edit View Run Tools Help
[ temperature.py ]
1 import microcontroller
2 temp = microcontroller.cpu.temperature
3 print(temp)

Shell
>>> %Run -c $EDITOR_CONTENT
>>> %Run -c $EDITOR_CONTENT
24.7977
>>>
```

CircuitPython (generic)

Electronics

Trafic light



```
automated traffic lights.py
1 import board
2 import digitalio
3 import time
4
5 led_red = digitalio.DigitalInOut(board.GP13)
6 led_red.direction = digitalio.Direction.OUTPUT
7
8 led_yellow = digitalio.DigitalInOut(board.GP12)
9 led_yellow.direction = digitalio.Direction.OUTPUT
10
11 led_green = digitalio.DigitalInOut(board.GP11)
12 led_green.direction = digitalio.Direction.OUTPUT
13
14
15 led_red.value = False
16 led_yellow.value = False
17 led_green.value = False
18
19 while True:
20     led_yellow.value = False
21     led_red.value = True
22     time.sleep(1)
23
24     led_yellow.value = True
25     time.sleep(0.3)
26
27     led_red.value = False
28     led_yellow.value = False
29     led_green.value = True
30     time.sleep(1)
31
```


Electronics

Button and LED



8 button and led.py * x

```
1 import board
2 import digitalio
3
4 led = digitalio.DigitalInOut(board.GP15)
5 led.direction = digitalio.Direction.OUTPUT
6 button = digitalio.DigitalInOut(board.GP0)
7 button.switch_to_input(pull=digitalio.Pull.DOWN)
8
9 while True:
10     led.value = button.value
```

Electronics

Blink external LED



```
5 blink led gpio15.py ×  
1 import time  
2 import board  
3 import digitalio  
4  
5 led = digitalio.DigitalInOut(board.GP15)  
6 led.direction = digitalio.Direction.OUTPUT  
7  
8 while True:  
9     led.value = not led.value  
10    time.sleep(0.5)
```

INSTYTUT BADAŃ I INNOWACJI W EDUKACJI

```
▶ print(" |--| ")  
print("  ..  ")  
print("  \__/" )  
print("inbie.pl")
```

```
↳ |--|  
  ..  
  \__/  
inbie.pl
```

Thank you for participating
in the INBIE course

```
▶ print("info@inbie.pl")
```

```
↳ info@inbie.pl
```