

Python Cheat Sheet



If, else if, else

```
# if / ifelse / else
foo = 3

if foo == 3:
    print('foo is 3')
elif foo == 2:
    print('foo is 2')
else:
    print('foo is neither 3 nor 2')
```



Basic Input / Output

```
# Print a message
print("Hello World")

# Print multiple values (of different types)
foo = "cruel"
print('Goodbye', foo, 'World') # => Goodbye cruel World

# Asking the user for a string
name = input('What is your name?')
```



Loops

```
# For
for i in range(10):
    print(i)

# While
while True:
    print("Hello")
```



Imports / time

```
# Import Libs
import time

# Print current time
print(time.time())
```



Lists and Dictionaries

```
# Lists
new_list = ['list0', 'list1', 'list2', 'list3']
new_list.append("list4") # Adds 'list4' to the end of the List
print(new_list[3]) # Prints: 'list2'

# Dictionaries
new_dict = {'a': 0, 'b': 1, 'c': 2}
new_dict['d'] = 3 # Creates a new key 'd' with value 4
print(new_dict['d']) # Prints: '3'
```



GPIO: Events

```
# Import Libs
import RPI.GPIO as GPIO

# Use BCM Mode
GPIO.setmode(GPIO.BCM)

# Set TASTER to Variable
TASTER = 24

# Define Function to run when Taster gets clicked
def on_taster(TASTER_GPIO):
    print('You pressed Taster')

# Add Event Detector
GPIO.add_event_detect(TASTER, GPIO.RISING, callback=on_taster)

try:
    while True:
        time.sleep(0.2)
except KeyboardInterrupt: # Check if Script is exited using CTRL + C
    print("Goodbye!")
    GPIO.cleanup() # Clear all pins (set all pins back to INPUT)
finally:
    GPIO.cleanup
```



Comparing Values

```
foo = 3

# Are two values equal?
if foo == 3:
    pass

# Are two values not equal?
if foo != 3:
    pass

# Less than another?
if foo < 3:
    pass

# Greater than another?
if foo > 3:
    pass

# Less than or equal to?
if foo <= 3:
    pass

# Greater than or equal to?
if foo >= 3:
    pass

# Boolean: True or False
```



GPIO: LEDs

```
# Import Libs
import RPI.GPIO as GPIO
import time

# Use BCM Mode
GPIO.setmode(GPIO.BCM)

# Set LED_ROT to Variable
LED_ROT = 17

# Interact with PIN
GPIO.output(LED_ROT, GPIO.HIGH) # Turn LED_ROT on
time.sleep(1) # sleep for 1 second
GPIO.output(LED_ROT, GPIO.LOW) # Turn LED_ROT off
```