MicroPython Cheat Sheet

1. **Mind the spaces**; code blocks are indented using spaces.
2. **Python is case sensitive**. It matters for variables, functions and any keyword in general.
3. **Don't Forget the Colons** (:) at the end the first line of an if, while, for:

|  |  |
| --- | --- |
| REPL Interaction  #ctr  print("hello",name,"); | Adding Comments / Annotations  # Single Line Comment  """  Multi-line comment  """ |
| Printing  #use Parentheses (), can print multiple things  name = 'jos'  print("hello",name,"how are you ?")  #print with no newline  print( 'Processing...', end='')  #Print with separator  print( 'one','two','three', sep=' | ' ) | General Help  #get help  help()  #List available modules  help('modules')  #overview of functions and constants  help(module/class/variable)  #List variables and loaded modules  dir() |
| Variable Assignment  myInteger = 10  myInteger += 5  myString = 'Hello World'  myList = [ "John", 'James', 'Laura'] | Basic Arithmetic  i = a + b  i = a – b  i = a / b  i = a \* b  i = a % b (Modulus/Remainder) |
| Loops  #Mind the spaces, Don't Forget the Colons  for i in range(1,10): #using a range  print i    for i in [1, 3, 5, 7]: #using a list  print i  #to exit the loop, use the instruction:  break  #Use for Loops Instead of while or range  #in Python, simple is good.  S = "lumberjack"  for c in S:  print(c) | IF Statements  #Mind the spaces, Don't Forget the Colons  if i == 7:  print("seven")  else:  print("Not seven!")    if i == 7:  print("seven")  elif i == 8:  print("eight")  elif i == 9:  print("nine")  else:  print "Not seven, eight or nine" |
| Define Functions  #Mind the spaces, Don't Forget the Colons  def myFunction(arg1, arg2):  #Code goes here  myValue = 42  return myValue  def mySumFunction( x, y, z=0 ):  sum = x + y + z  return sum  print( mySumFunction ( 1,3,5 ) )  # this would display 9 | Convert between types  #To Integer  int(2.345) #2  #To string  str(2.345) #'2.345'#To character  chr(65) # 'A'#to binary string  bin(42) #'0b101010'  #to hexadecimal string  hex(42) #'0x20' |
| time.Sleep functions  import time  #Sleep for the given number of seconds  time.sleep(seconds)  #Delay for given number of milliseconds  time.sleep\_ms(ms)  #Delay for given number of microseconds  time.sleep\_us(us)  ***NOTE:***Do not confuse time.sleep() and machine.sleep()  >>>machine.sleep() ***suspends*** the device. | Working with modules  **#mymodule.py**  def myfunc(): print("Hello!")  #Import a module  import mymodule  mymodule.myfunc()*Hello!*  #import all functions from the module  from mymodule import \* myfunc() *Hello!*  **#List current loaded modules**  import sys; sys.modules |
| String and output Formatting  # use a string with {} for placeholders  # and .format( 1,2,3) to supply values  "buy {} apples get {} free".format(3,1)  var**=**"Temperature{:5.2f} Pressure{:06d}"**.**format (temp, press) | machine module: Control Outputs using Pin  from machine import Pin  #low level pin access  **from** machine **import** Pin  *# create an output pin on pin #0*  p0 **=** Pin(0, Pin**.**OUT)  *# set the value low then high*  p0**.**value(0)  p0**.**value(1) |
| Error handling  #define risky function  import urandom  def takeaRisk():  return (3/urandom.getrandbits(2))  #deal with the risk  try:  x = takeaRisk()  except ZeroDivisionError:  print("We hit a division error!")  else:  print("The result was :",x)  finally:  print("and this always runs")  #List of exceptions  [http://docs.micropython.org/en/latest /esp8266/library/builtins.html#exceptions](http://docs.micropython.org/en/latest/esp8266/library/builtins.html#exceptions) | Strings and ByteStrings  my\_str = "Hello MicroPython"  #unicode string to bytes  str.encode(my\_str)  # Show it in byte representation  print(bytes,type(bytes))  #bytes to unicode string  my\_decoded\_str = bytes.decode(bytes)  # show it in string representation  print(my\_decoded\_str, type(my\_decoded\_str))**#print byte-**  **#string as hex string**  import binascii  b"abcde".decode("utf-8")  >>> binascii.hexlify(bytearray(array\_alpha))  '8535eaf1' |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Left** | **Right** | **Top** | **Bottom** | **ESP32** | **Device** | **Function** | **Devices** | **Free/Usable** |
| 1 | T0 |  |  | GPIO 01 | Serial/USB | TXD1 | UART0 | Conflict with USB |
| 2 | G2 |  |  | GPIO 02 |  |  |  | Yes, Touch |
| 3 | R0 |  |  | GPIO 03 | Serial/USB | RXD1 | UART0 | Conflict with USB |
|  |  |  |  | GPIO 04 | SPI - SD Card | CS - SD Card | TFCARD | n/a |
| 5 | G5 |  |  | GPIO 05 |  |  |  | Yes |
|  |  |  |  | GPIO 14 | Display | CS - Display | ILI9341 | n/a |
| 16 | R2 |  |  | GPIO 16 | SPI-RAM |  |  | Not with SPI RAM |
| 17 | T2 |  |  | GPIO 17 | SPI-RAM |  |  | Not with SPI RAM |
|  |  | 18 | SCK | GPIO 18 | SPI - SD Card, Display | SCK | TFCARD, ILI9341 | Conflict SPI bus |
|  |  | 19 | MI | GPIO 19 | SPI - SD Card, Display | MISO | TFCARD, ILI9341 | Conflict SPI bus |
|  |  | 21 | SDA | GPIO 21 | I2C Bus,MPU9250, Grove | Data SDA | MPU9250 / GROVE | for I2C bus |
|  |  | 22 | SCL | GPIO 22 | I2C Bus,MPU9250, Grove | Clock SCL | MPU9250 / GROVE | for I2C bus |
|  |  | 23 | MO | GPIO 23 | SPI - SD Card, Display | MOSI | TFCARD, ILI9341 | Conflict SPI bus |
| 25 | DA |  |  | GPIO 25 | Internal Speaker | DAC1 | Digital -> Analog | Usable as Input |
| 26 | DA |  |  | GPIO 26 |  | DAC2 | Digital -> Analog | Yes |
|  |  |  |  | GPIO 27 | Display | Data/Command | ILI9341 | Conflict with Display |
|  |  |  |  | GPIO 32 | Display | Backlight | ILI9341 | Conflict with Display |
|  |  |  |  | GPIO 33 | Display | Reset | ILI9341 | Conflict with Display |
| 35 | AD |  |  | GPIO 35 | Analog -> Digital | ADC1 | Analog -> Digital | Input only |
| 36 | AD |  |  | GPIO 36 | Analog -> Digital | ADC2 | Analog -> Digital | Input only |
| 3V3 | 3V3 | 3V3 | 3V3 |  |  | 3.3 volt |  |  |
| 5V | 5V | 5V | 5V |  |  | 5 volt |  |  |
| G | G | G | G |  |  | Ground |  |  |
| BAT | BAT |  |  |  |  | Battery |  |  |