

Intro to the Raspberry Pi (RPi) & The Internet of Things (IoT)

Class 2 – 2018 / 03 / 18

Raspberry Pi Name Origin



... from last class
Thanks for this question!

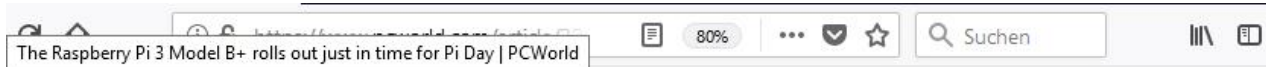


TS: Where does the name Raspberry Pi come from?

Raspberry is a reference to a fruit naming tradition in the old days of microcomputers. A lot of computer companies were named after fruit. There's Tangerine Computer Systems, Apricot Computers, and the old British company Acorn, which is a family of fruit.

Pi is because originally we were going to produce a computer that could only really run Python. So the Pi in there is for Python. Now you can run Python on the Raspberry Pi but the design we ended up going with is much more capable than the original we thought of, so it's kind of outlived its name a little bit.

PCWorld on „Pi Day“ 3 / 14 – New RPi Model 3 B+!



NEWS

The Raspberry Pi 3 Model B+ rolls out just in time for Pi Day

For \$35 you can pick-up the Raspberry Pi 3 Model B+ with a faster processor and dual-band Wi-Fi.



By [Ian Paul](#)

Contributor, PCWorld | MAR 14, 2018 9:03 AM PT

Microsoft



Get the latest version of
Excel with **Office 365**.

Now only
\$10.00 user/mo.
[CLICK HERE](#) →



MORE LIKE THIS



Raspberry Pi: Everything you need to know



Best Raspberry Pi 3 cases: 10 great options to dress up your mini PC



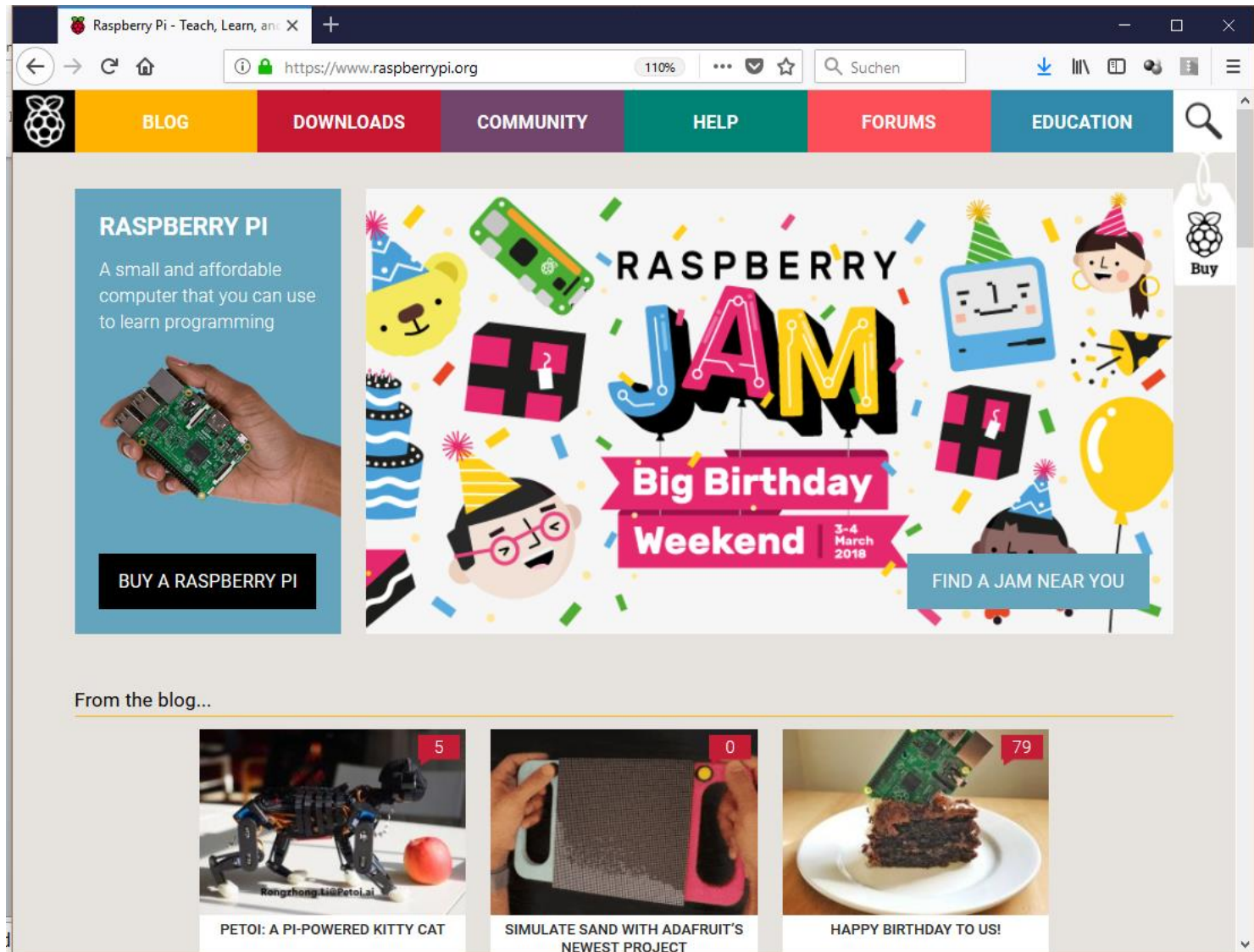
Best Raspberry Pi 3 kits: 8 options for beginners and experienced makers



VIDEO

Kyle Bennett from HardOCP: What's up with Nvidia's GeForce Partner Program? [...]

Check out <http://www.raspberrypi.org>



AstroPi

https://www.youtube.com/watch?v=yjll_4JY98g

MISSION ZEROMISSION SPACE LABABOUTPAST MISSIONS

SEARCHING FOR THE NEXT GENERATION OF SPACE SCIENTISTS!

YOU COULD

- Design a science experiment that explores life on earth or life in space!
- Take control of a space computer for two orbits of the Earth!
- Win awesome space related prizes!



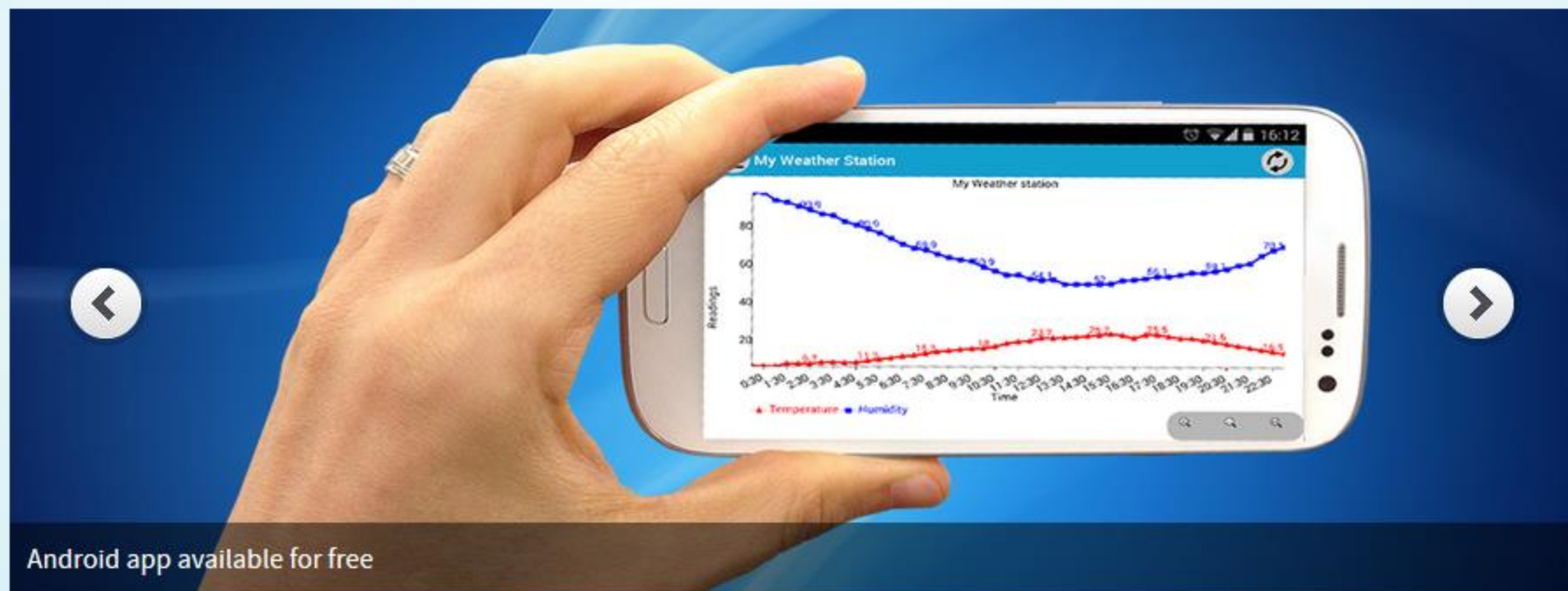
WHAT IS ASTRO PI?

Astro Pi is an annual science and coding competition where student-written code is run on the International Space Station! Watch the [promo video](#).



Raspberry Weather

search...

[Home](#)[Software set up](#)[Hardware set up](#)[Optional](#)[Additional info](#)

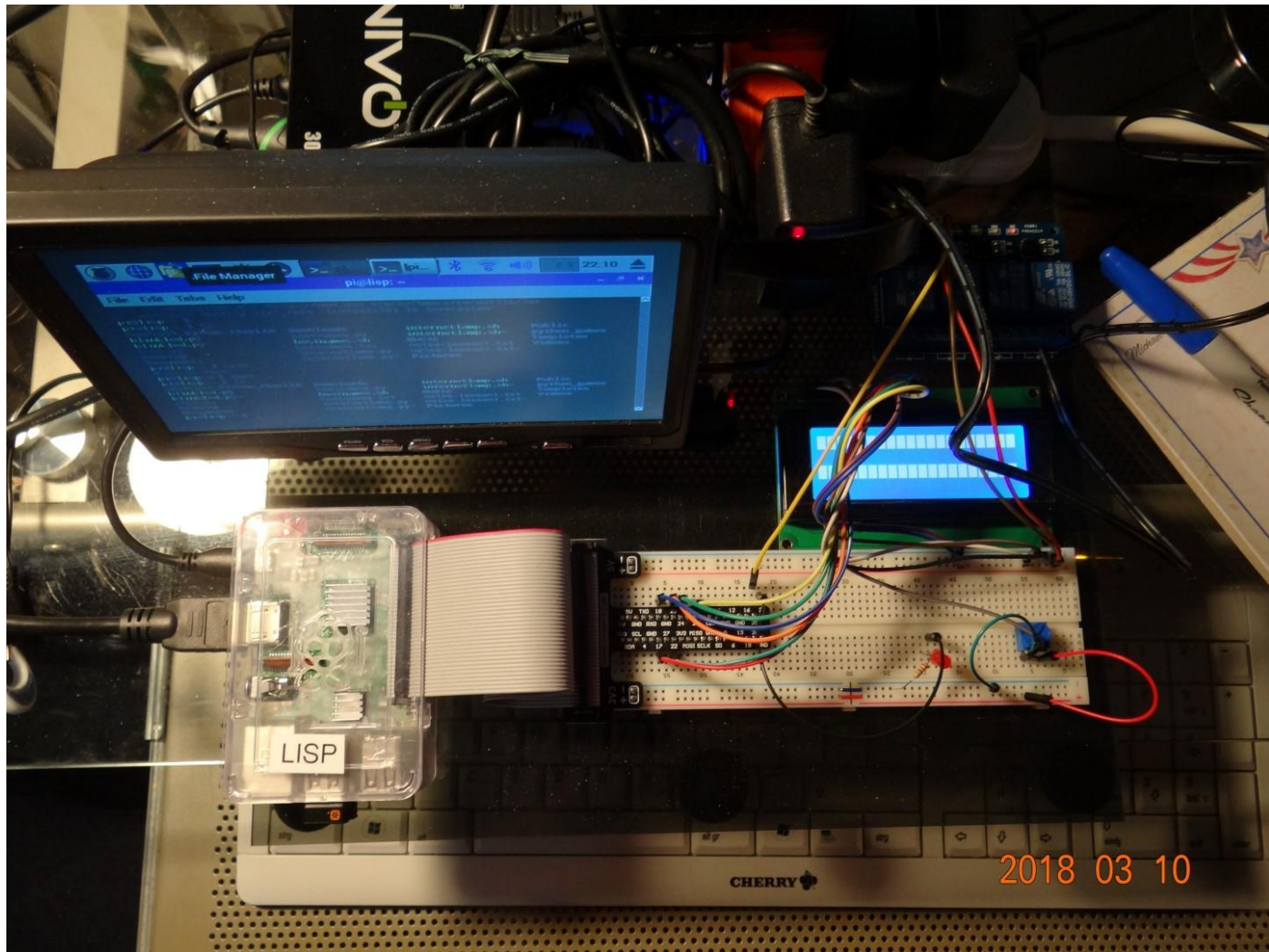
Optional - „RPi Workstation“ for Easy Transport

- Get an acrylic sheet from Osh!



- Double Sided *Really Strong* Tape, e.g., M3 Mounting Tape (not simple transparent Scotch double sided)
- Get a magnifier glass if your eyes are as bad as mine (GPIO labels are small)
- Notice that monitor has glued tape underneath already!

Idea for Setup

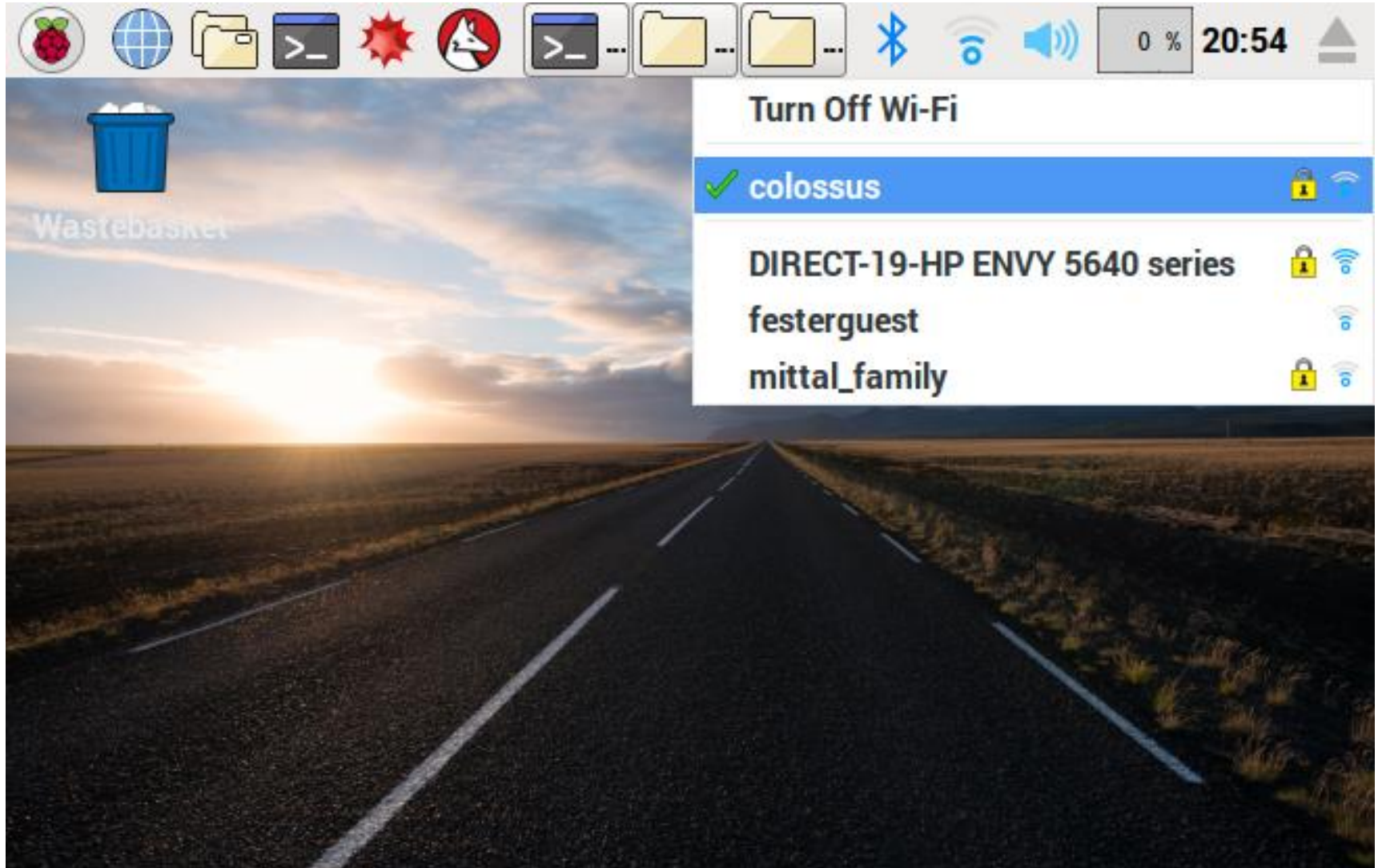


Configuration

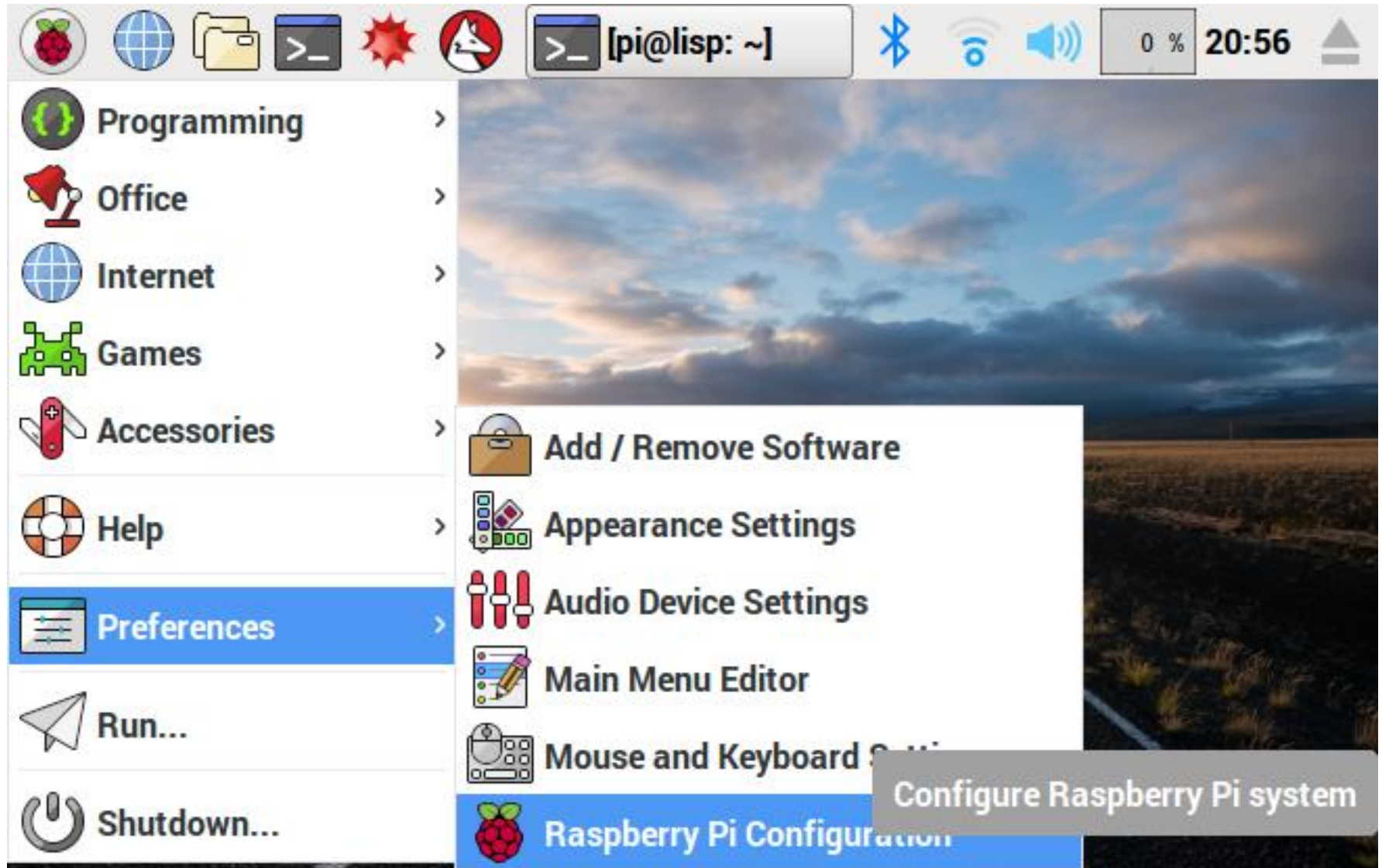
- Screen resolution
- Internet WiFi
- Password
 - Default user: **pi** (no need to change)
 - Default password: **raspberry** – let's change that!
 - Super user login disabled, but **sudo**
- Set Hostname from Lottery
 - Use clear tape for stickers
 - 1 on RPi case, 1 under keyboard
 - Leave the keyboard dongles plugged in all the time to avoid confusion

Wifi Configuration

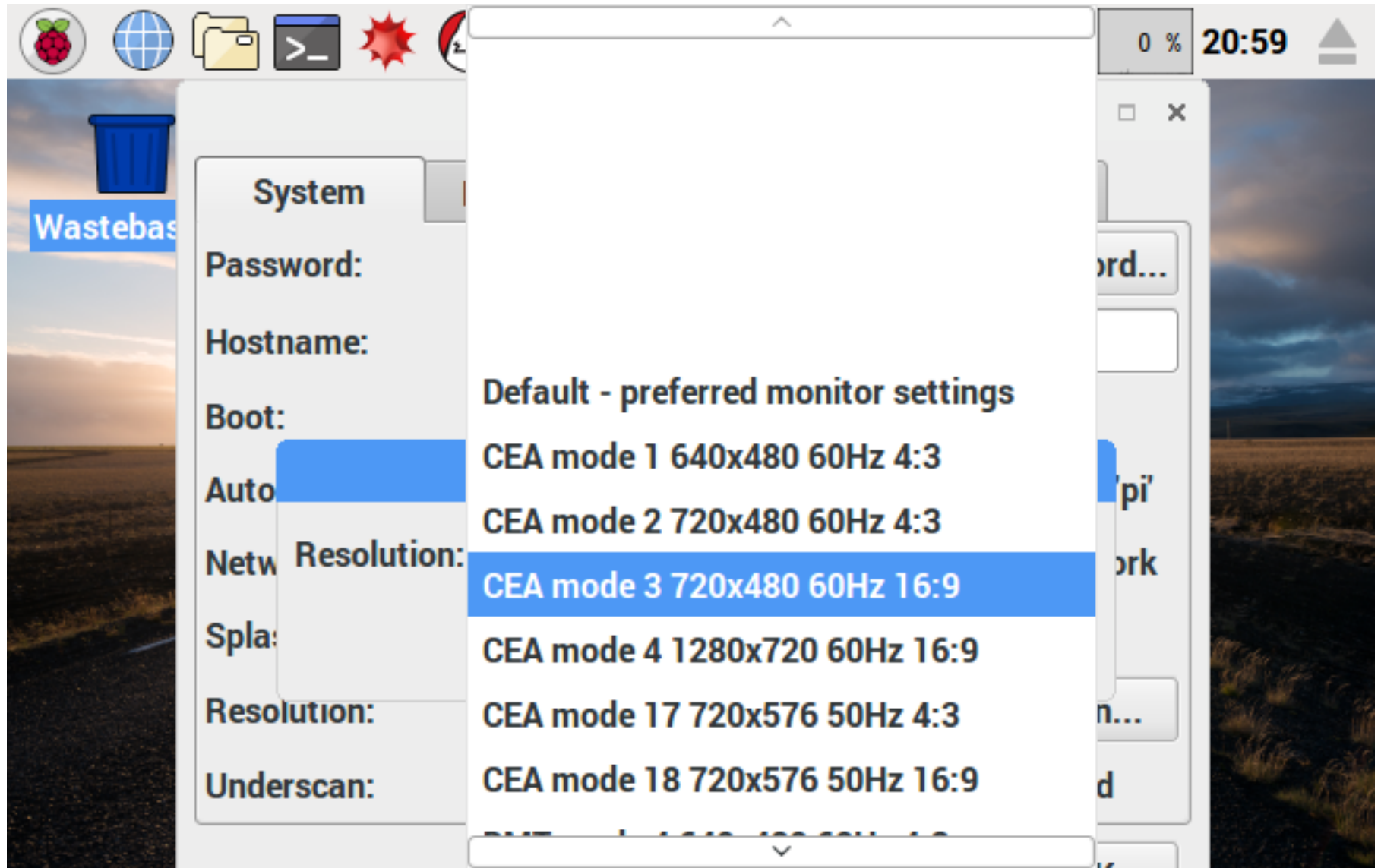
Wifi Password is



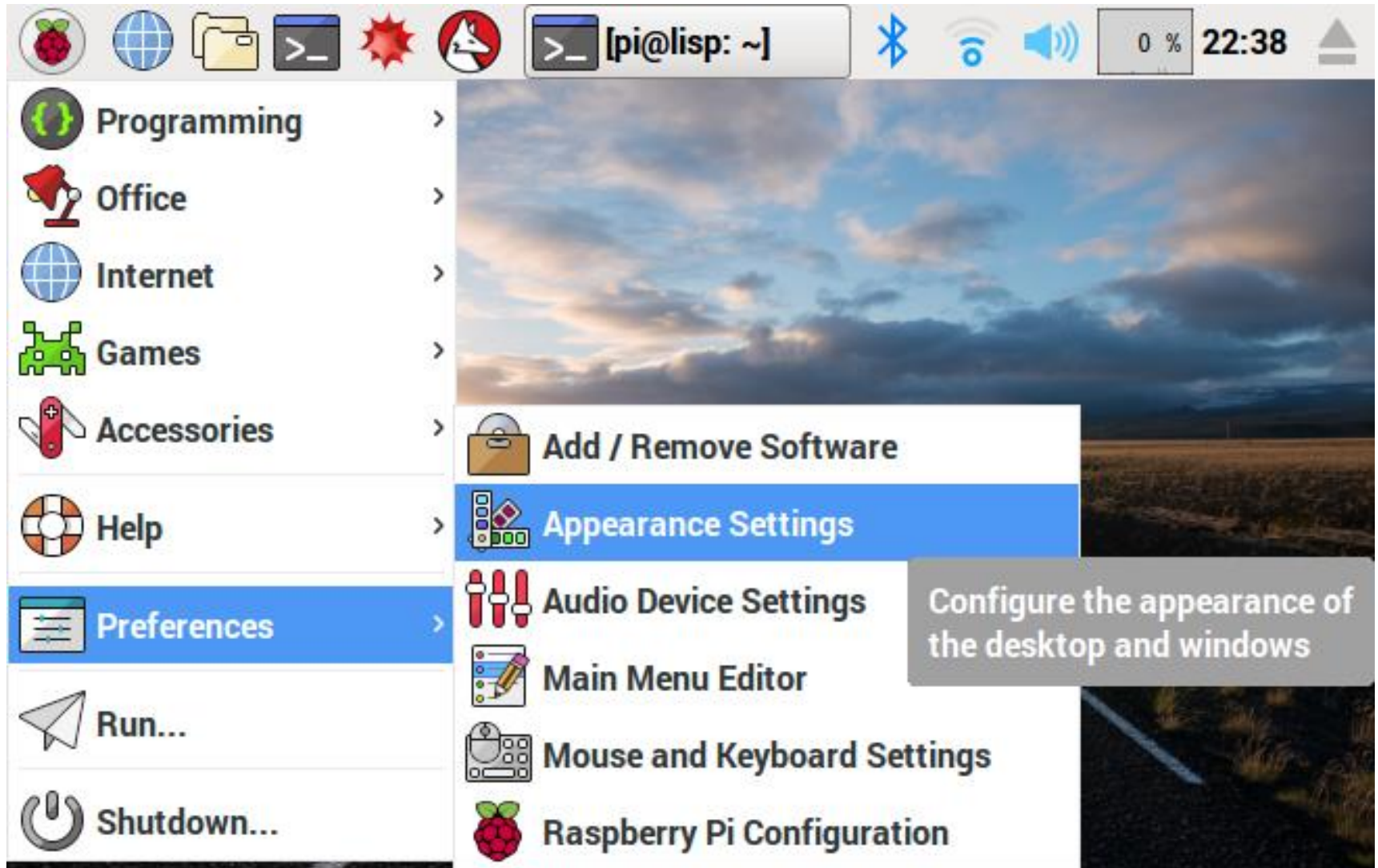
Screen Resolution



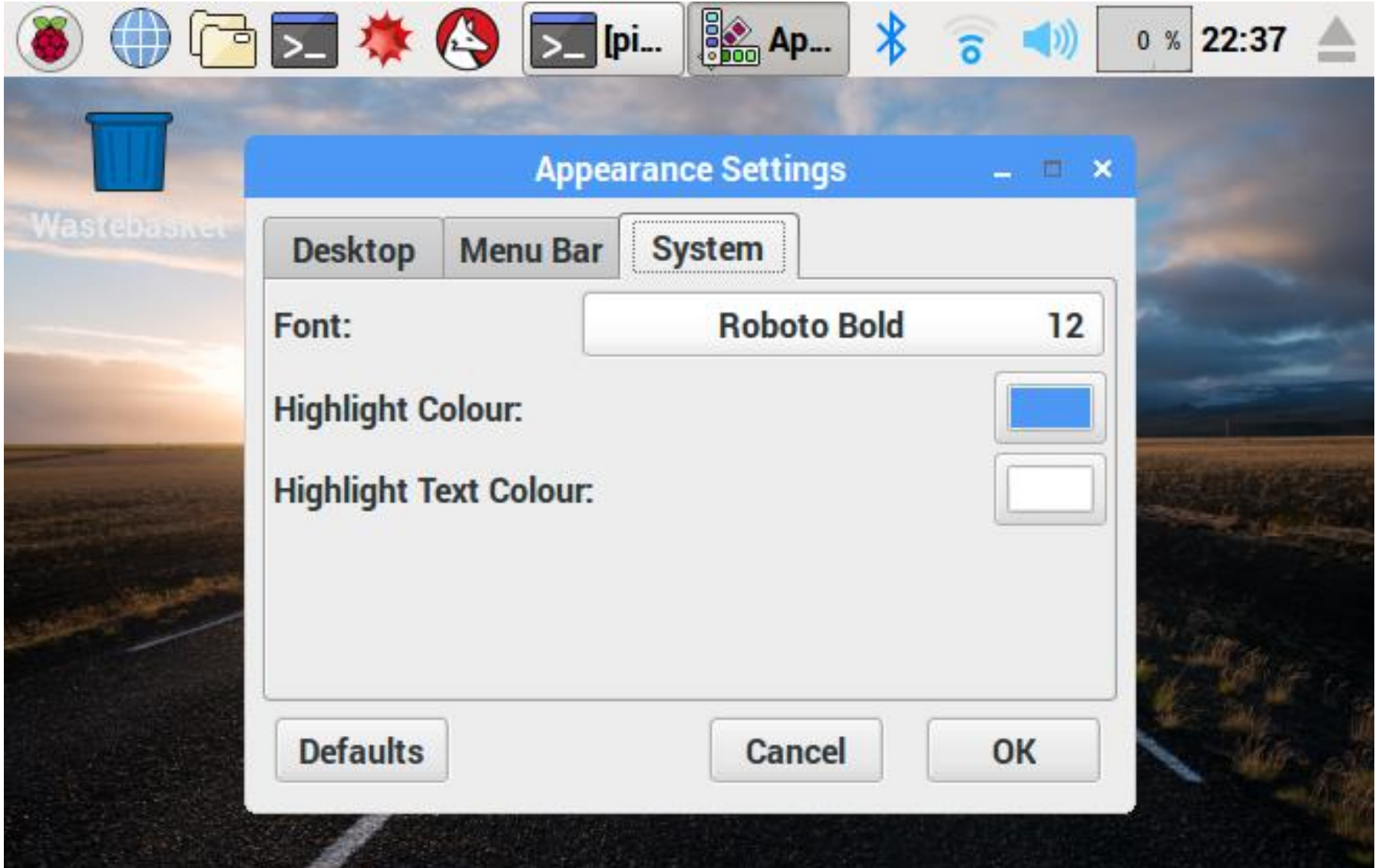
Screen Resolution



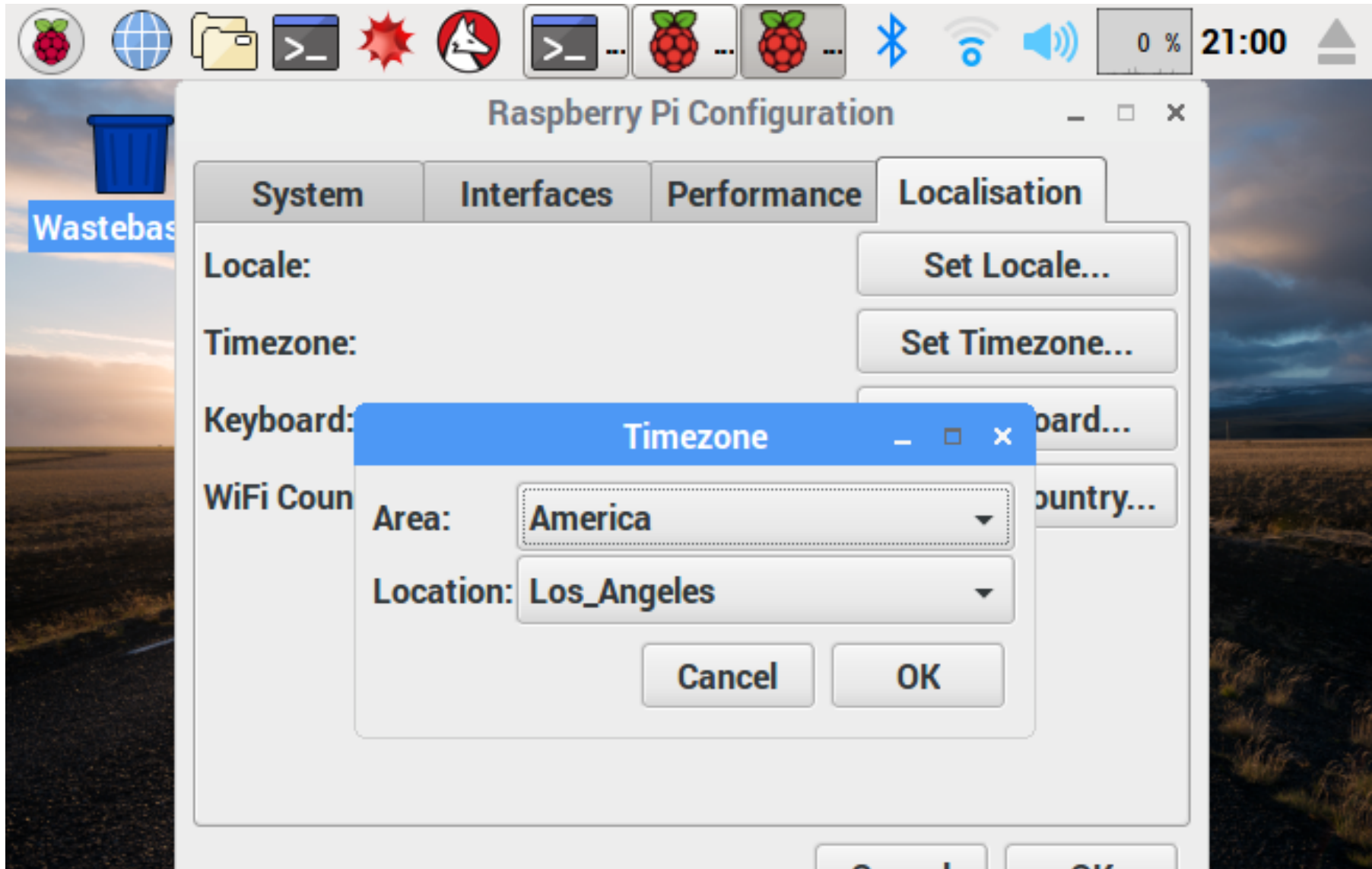
Increase Font Size



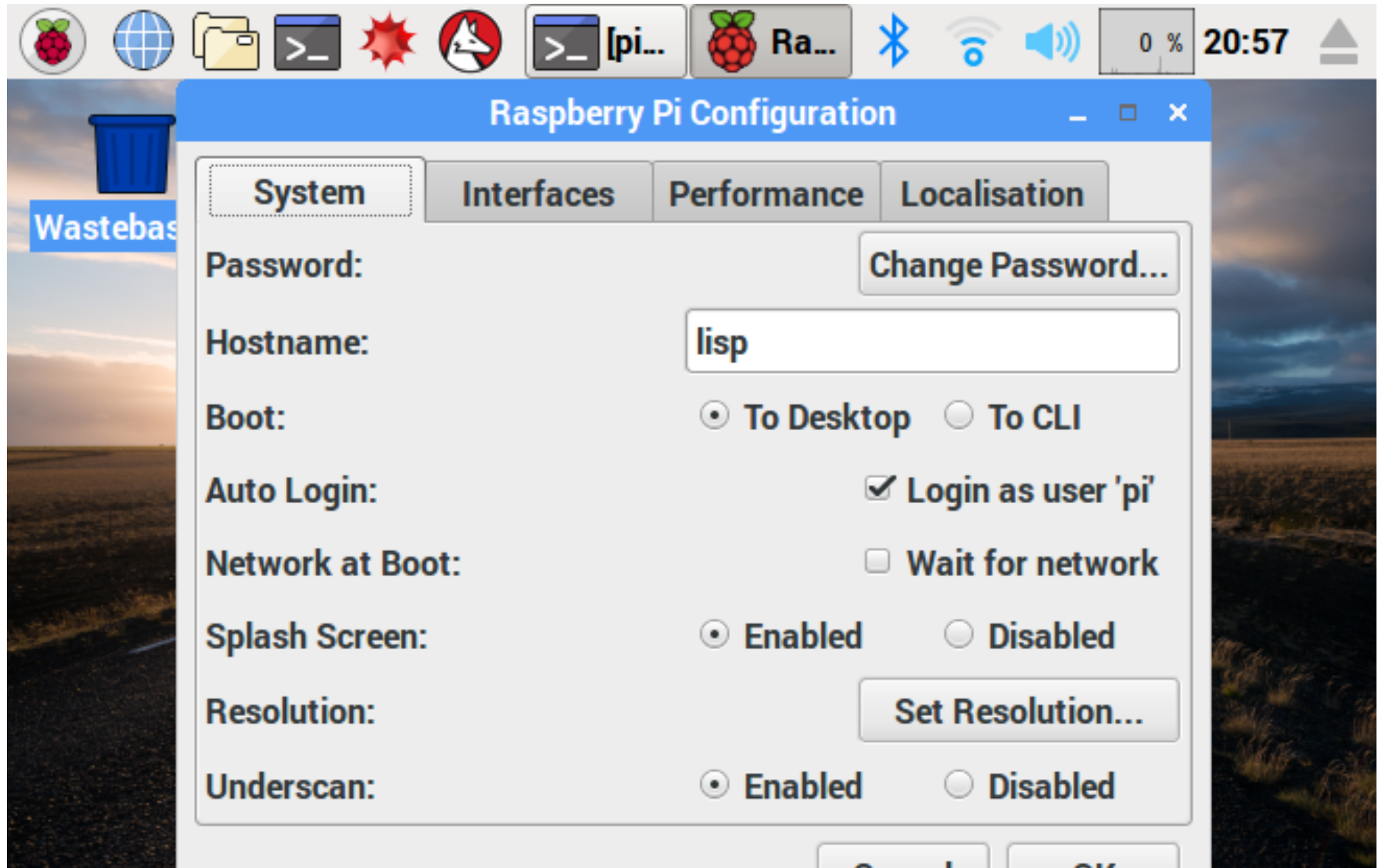
The screenshot shows the Raspberry Pi desktop environment. At the top is a dock with various application icons including a Raspberry Pi logo, a globe, a folder, a terminal, a red star, a cat icon, a terminal window titled 'ipi...', an application window titled 'Ap...', and system status icons for Bluetooth, Wi-Fi, and audio. The desktop background is a scenic image of a sunset over a field. On the left side, there is a 'Wastebasket' icon. In the center, the 'Appearance Settings' dialog box is open. It has three tabs: 'Desktop', 'Menu Bar', and 'System', with 'System' currently selected. The 'System' tab contains settings for the font, which is set to 'Roboto Bold' in size '12'. Below the font settings are two color selection boxes: 'Highlight Colour' (set to blue) and 'Highlight Text Colour' (set to white). At the bottom of the dialog are three buttons: 'Defaults', 'Cancel', and 'OK'.



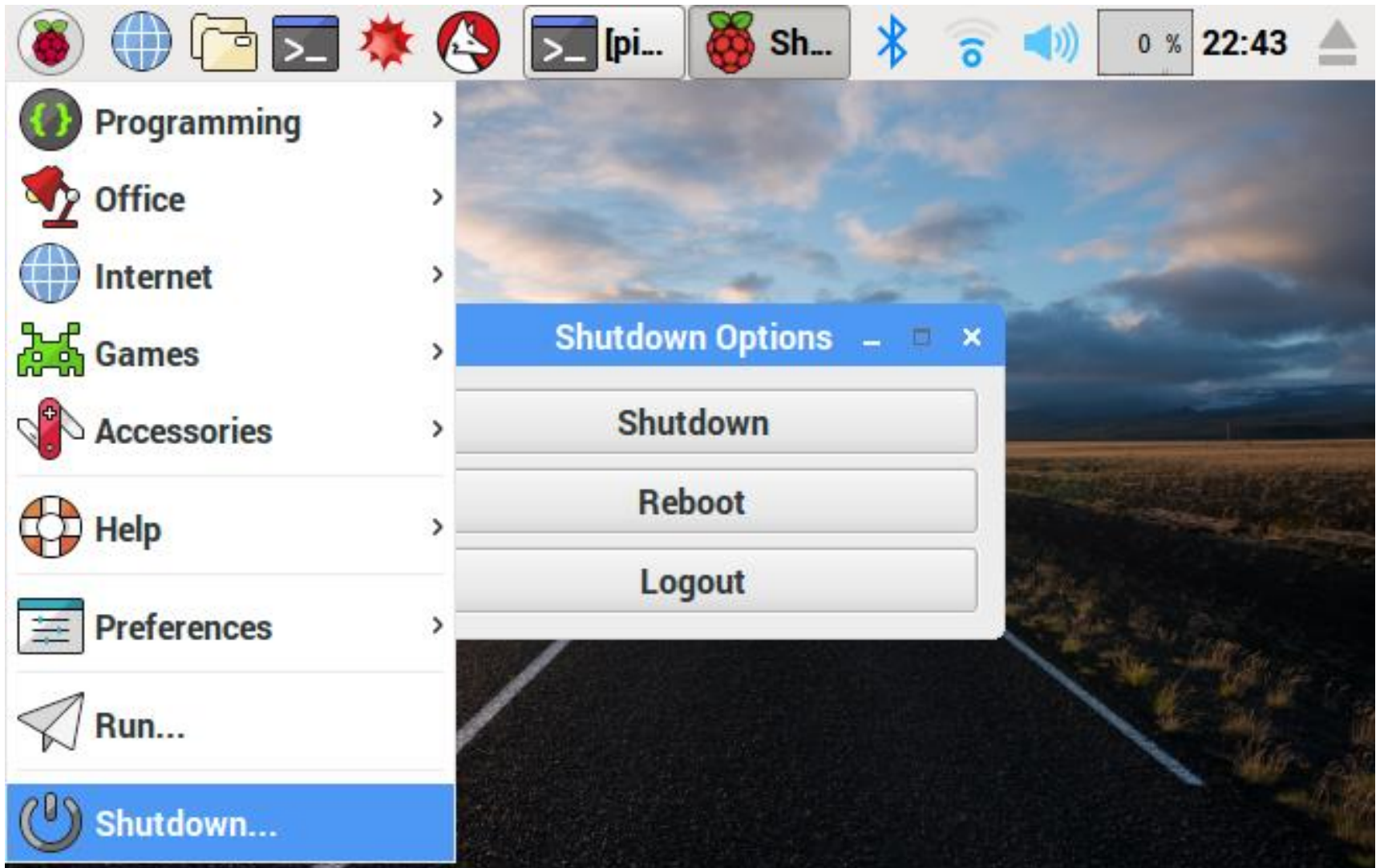
TimeZone / Localization for Clock



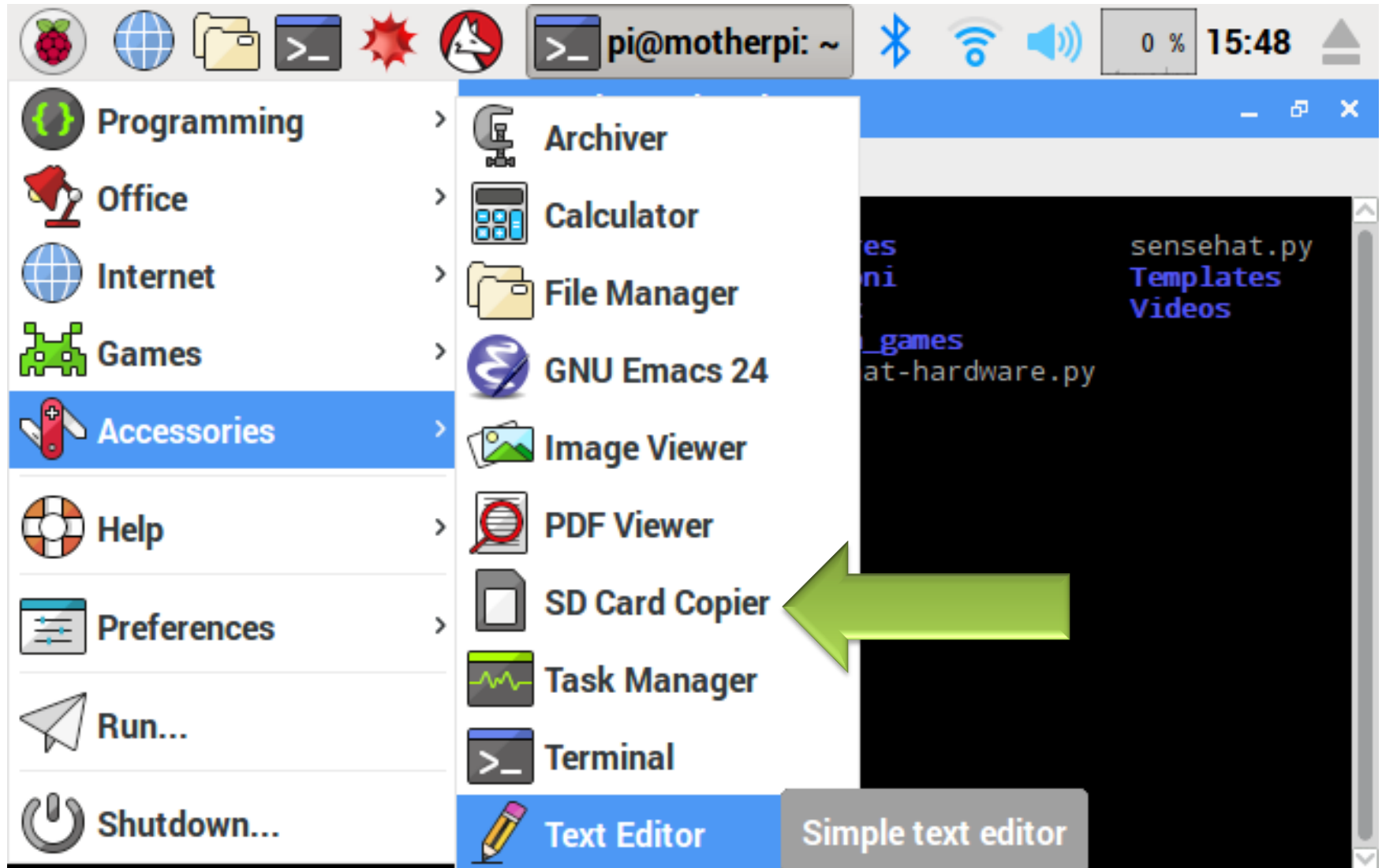
Password & Hostname from Lottery



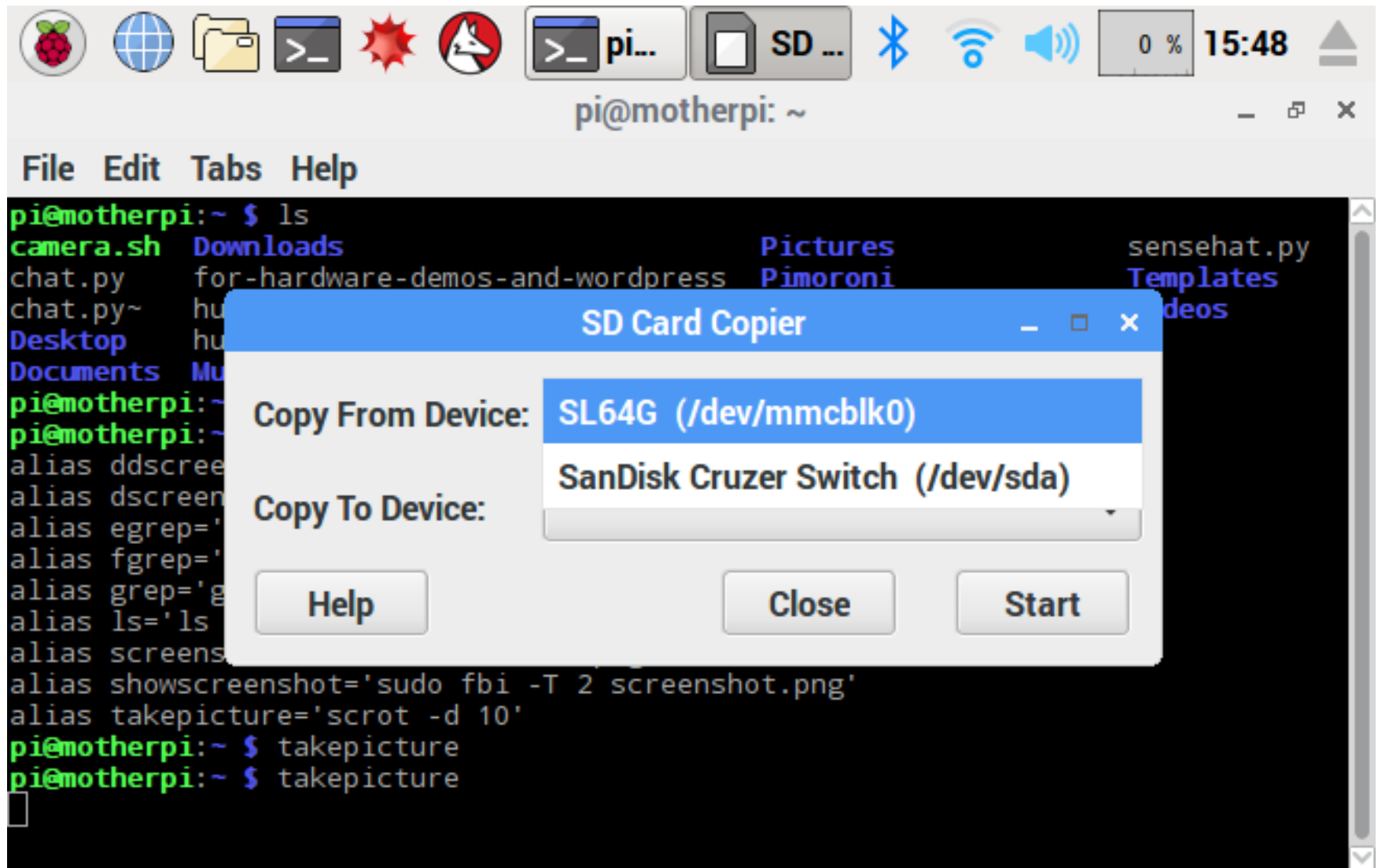
Reboot - Never Just Unplugg Your RPi!



MAKE A BACKUP!!



Use the USB SDCard Adapter



Adapter USB – Micro SDCard



Exporing PIXEL – Taking Notes in Class

- PIXEL =
'Pi Improved XWindow Environment, Lightweight'
- Show File Browser
- **Take screenshots during class of your work**
 - Keep a record of what you did
 - **FN + Delete = „Print Screen“**
- TouchPad
 - Use 2 Finger Scrolling
 - Right Mouse Button for Mac Users 😊
- Browser
 - Chromium
 - Flash
 - YouTube works also!

PIXEL Keys / Taking Notes in Class

- Window not showing all content?
Use maximize button!
 - Also minimize, close
- **Switch / cycle between windows: Alt-tab**
 - Open the Text Editor
 - Take notes using Alt-tab switching!!
 - Between Consol / Text Editor
- Close window Ctrl-w
- Use Ctrl-Esc for „Start Menu“

Linux Basics

- Some History
- Command Shell / BASH / „Terminal“ / Console
 - the „command post“ of every Unix computer
- Why not do everything from the desktop?
 - The shell is much more powerful and versatile
 - Allows „scripting“
 - Most things can ONLY be done from the shell
 - True hackers don't use the desktop 😊

Influential Unix Hackers

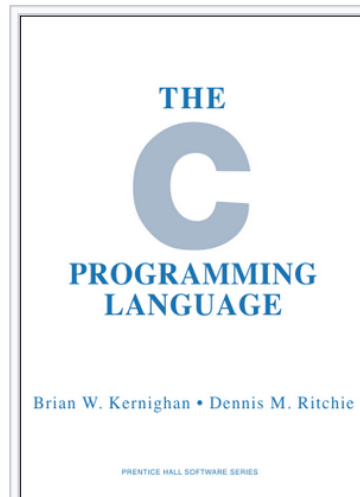


Ken Thompson and Dennis Ritchie, 
principal developers of Research Unix



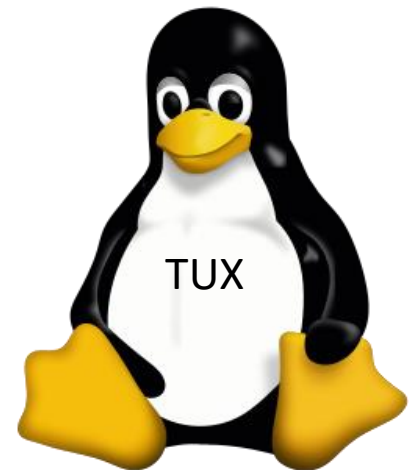
Ken Thompson (sitting) and Dennis 
Ritchie working together at a PDP-11

K&R C –
„The C Bible“



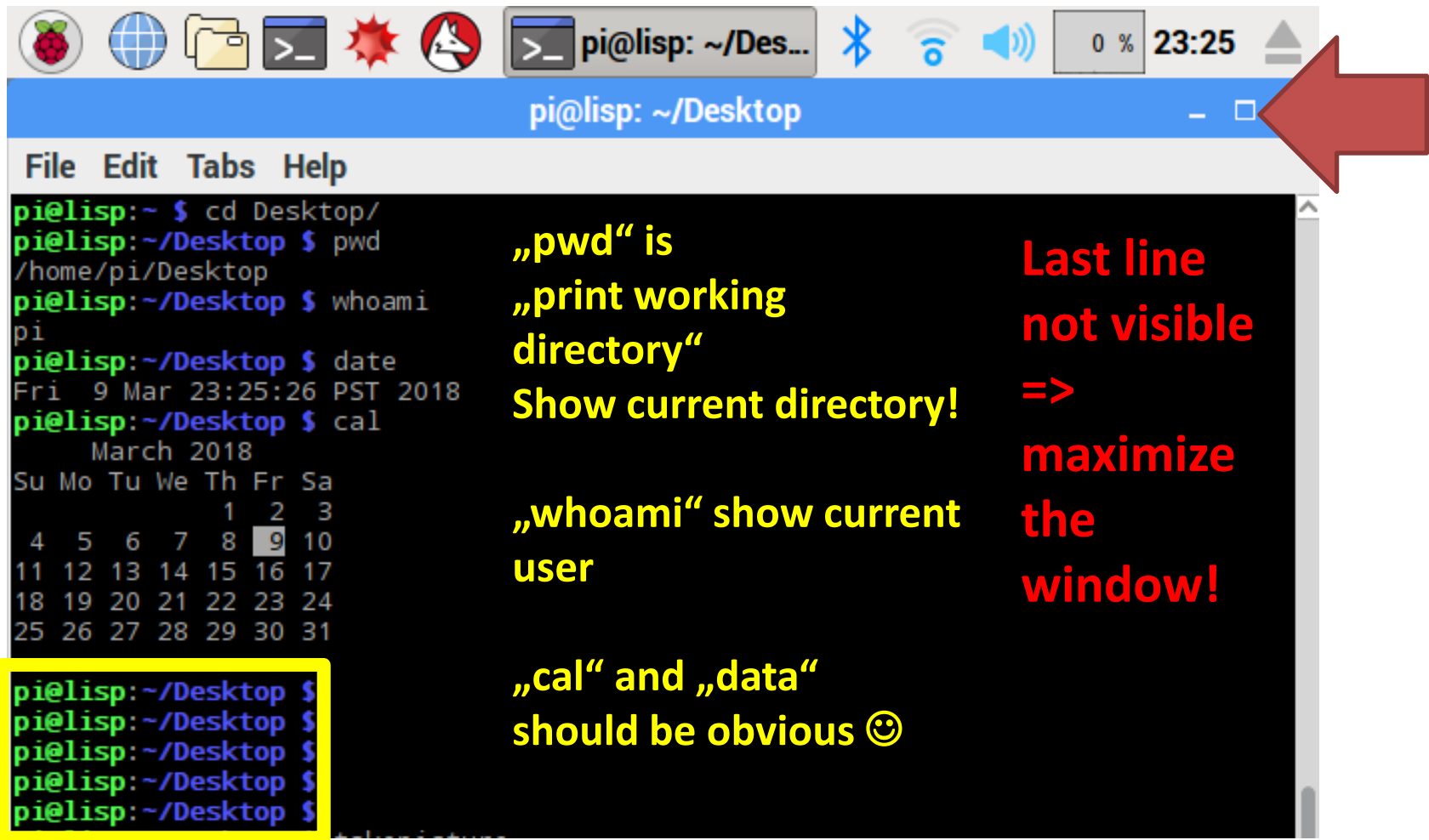
The cover of the book, *The* 
C Programming Language,
first edition by Brian
Kernighan and Dennis Ritchie

Linus Torvalds
Linux = Linus' Unix



Command Shell = BASH = Terminal =
Console = Command Line Interface (CLI) ...

PROMPT = <username>@<hostname>: **<current directory>**



The screenshot shows a terminal window titled "pi@lisp: ~/Desktop". The window has a menu bar with "File", "Edit", "Tabs", and "Help". The terminal content is as follows:

```
pi@lisp:~ $ cd Desktop/
pi@lisp:~/Desktop $ pwd
/home/pi/Desktop
pi@lisp:~/Desktop $ whoami
pi
pi@lisp:~/Desktop $ date
Fri 9 Mar 23:25:26 PST 2018
pi@lisp:~/Desktop $ cal
March 2018
Su Mo Tu We Th Fr Sa
        1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

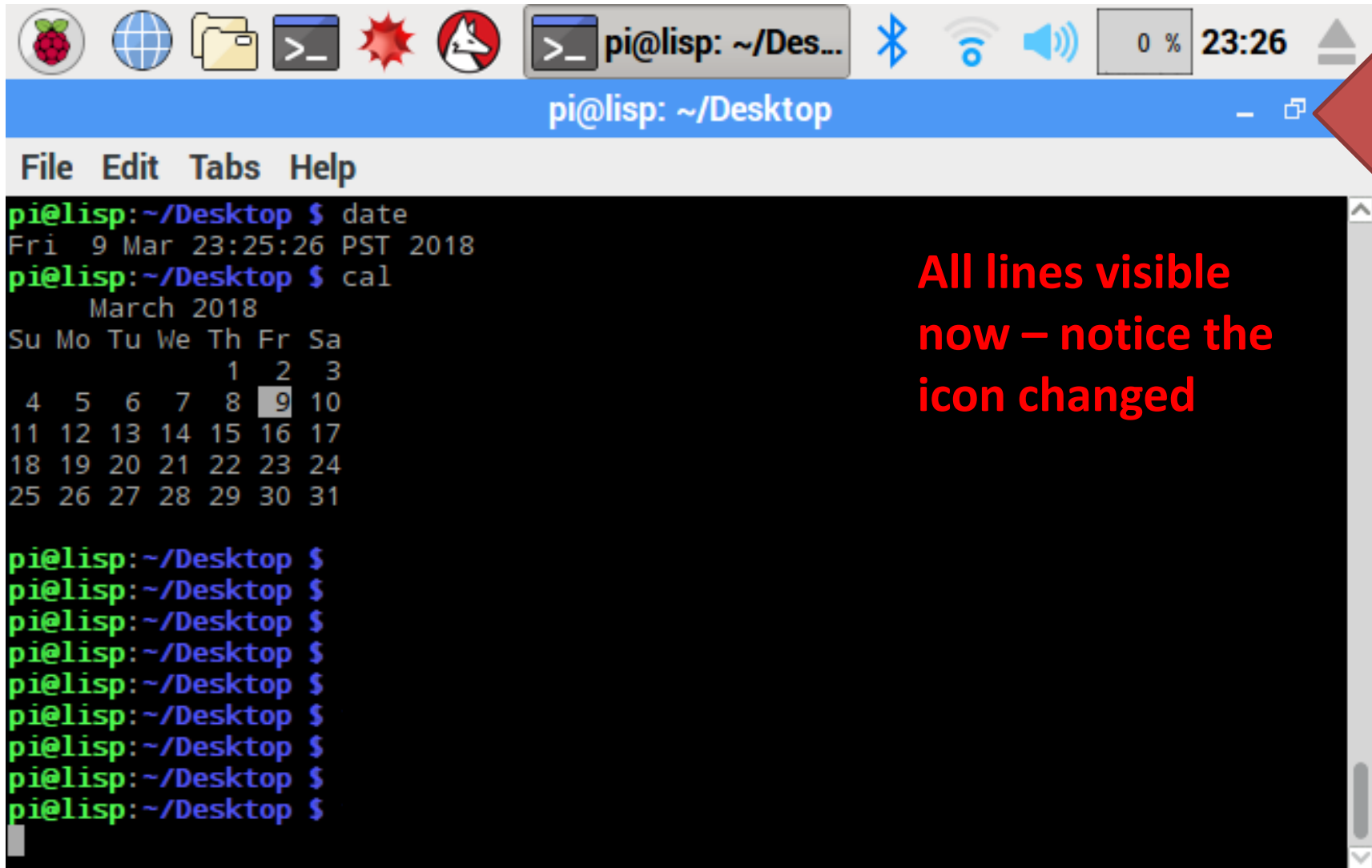
pi@lisp:~/Desktop $
pi@lisp:~/Desktop $
pi@lisp:~/Desktop $
pi@lisp:~/Desktop $
pi@lisp:~/Desktop $
```

Annotations on the right side of the terminal window:

- „pwd“ is „print working directory“
- Show current directory!
- „whoami“ show current user
- „cal“ and „data“ should be obvious 😊
- Last line not visible => maximize the window!

A red arrow points to the window title bar, and a yellow box highlights the prompt area.

Windows: Minimize, Maximize, Close



The screenshot shows a Raspberry Pi desktop environment. At the top is a taskbar with various icons including a Raspberry Pi logo, a globe, a folder, a terminal icon, a red star, a cat icon, and a window icon. The window icon is highlighted with a red arrow. Below the taskbar is a blue title bar with the text "pi@lisp: ~/Desktop" and window control buttons (minimize, maximize, close). The main window is a terminal with a menu bar (File, Edit, Tabs, Help) and a black background. The terminal displays the output of the 'date' and 'cal' commands, showing the current date and a calendar for March 2018. The terminal text is as follows:

```
pi@lisp:~/Desktop $ date
Fri 9 Mar 23:25:26 PST 2018
pi@lisp:~/Desktop $ cal
March 2018
Su Mo Tu We Th Fr Sa
      1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

pi@lisp:~/Desktop $
pi@lisp:~/Desktop $
pi@lisp:~/Desktop $
pi@lisp:~/Desktop $
pi@lisp:~/Desktop $
pi@lisp:~/Desktop $
pi@lisp:~/Desktop $
pi@lisp:~/Desktop $
pi@lisp:~/Desktop $
```

All lines visible now – notice the icon changed

BASH Basic Keyboard Shortcuts

- Use „clear“ in the console to clear content
 - However, HISTORY is still available
 - And you can also scroll back
- Make a new „tab“ with Ctrl-Shift-t
- Close a tab with with Ctrl-Shift-w
- Close a shell / BASH with Ctrl-Shift-q
- To get control back over the BASH, use Ctrl-c to interrupt / cancel any currently running command
 - Show „top“ process managment program
 - „q“ or „Ctrl-c“ to quit it

BASH Navigation

- Use tab key to complete filenames in current directory!
 - type: `cd ~`
 - type: `cd De<tab>` -> completes to `cd Desktop/`
- Use up/down arrows for history of commands
- Use Ctrl-r for „reverse search“, enter search string
 - Use Ctrl-c to cancel
 - Enter to re-use that command from history

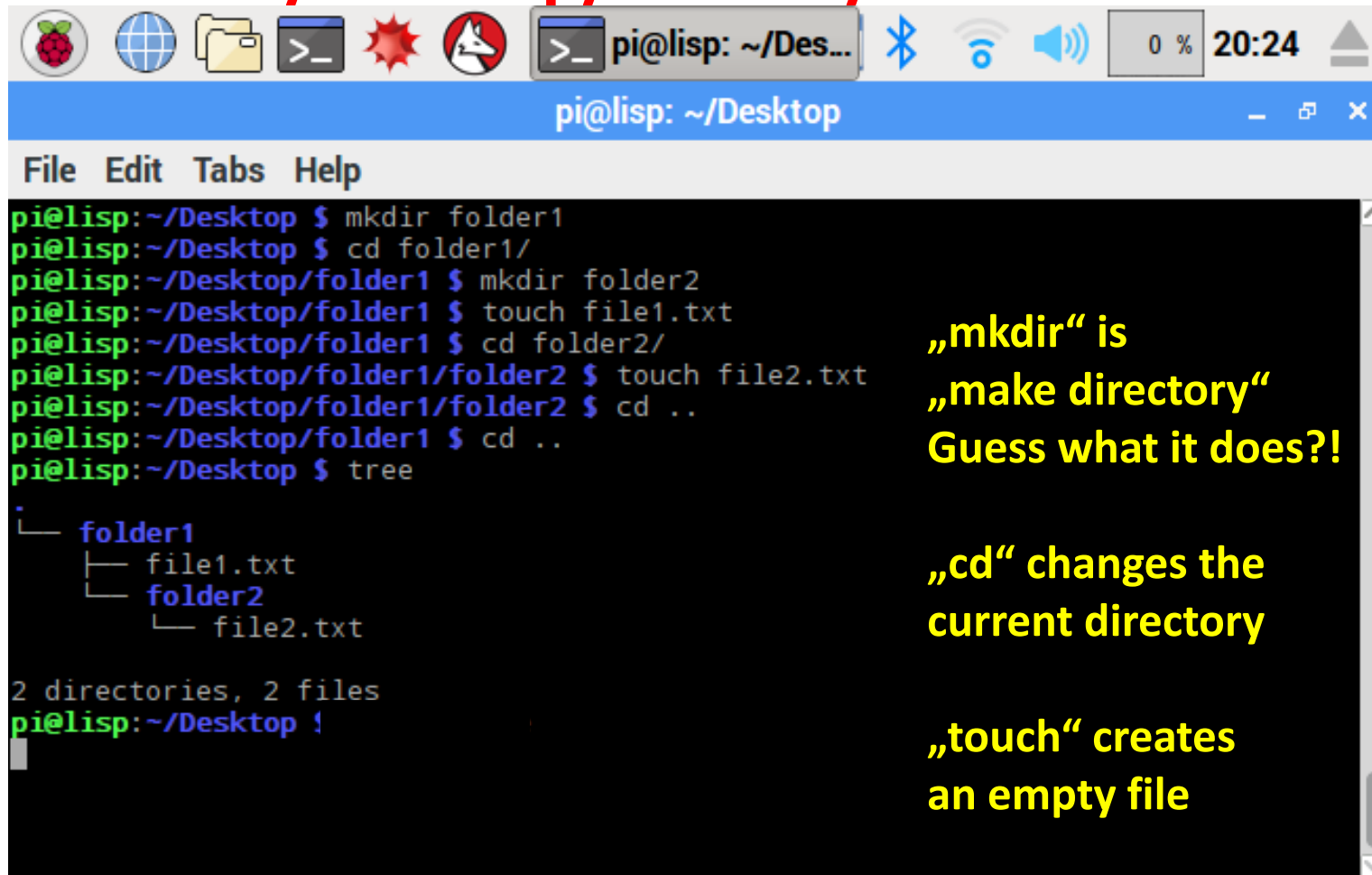
Files, Directories, and Trees

- Files = clear, right? Like in MacOS, Windows, ...
 - Files can be „hidden“
(start with a „period“, eg., „.alias“)
 - Files CAN have an extension, eg., „.txt“ for „text file“
 - Files are OWNED by a user („pi“ or „root“):
 - Privacy, Security (Access Rights)
 - You cannot delete or rename or move files from owner „root“, for examples
 - Files can be created with the File Manager
 - Or from the BASH shell!
- Directories = like folders
 - Can also be hidden, have owners, ...
 - Directories can contain other directories
 - > „Tree structure“

Files, Directories, and Trees - 2

- Each file is contained in SOME directory

~/Desktop/folder1/file1.txt = FILE PATH



The screenshot shows a terminal window on a Raspberry Pi. The window title is 'pi@lisp: ~/Desktop'. The terminal output shows the following commands and their results:

```
pi@lisp:~/Desktop $ mkdir folder1
pi@lisp:~/Desktop $ cd folder1/
pi@lisp:~/Desktop/folder1 $ mkdir folder2
pi@lisp:~/Desktop/folder1 $ touch file1.txt
pi@lisp:~/Desktop/folder1 $ cd folder2/
pi@lisp:~/Desktop/folder1/folder2 $ touch file2.txt
pi@lisp:~/Desktop/folder1/folder2 $ cd ..
pi@lisp:~/Desktop/folder1 $ cd ..
pi@lisp:~/Desktop $ tree
```

The output of the 'tree' command shows a directory structure:

```
.
├── folder1
│   ├── file1.txt
│   └── folder2
│       └── file2.txt
```

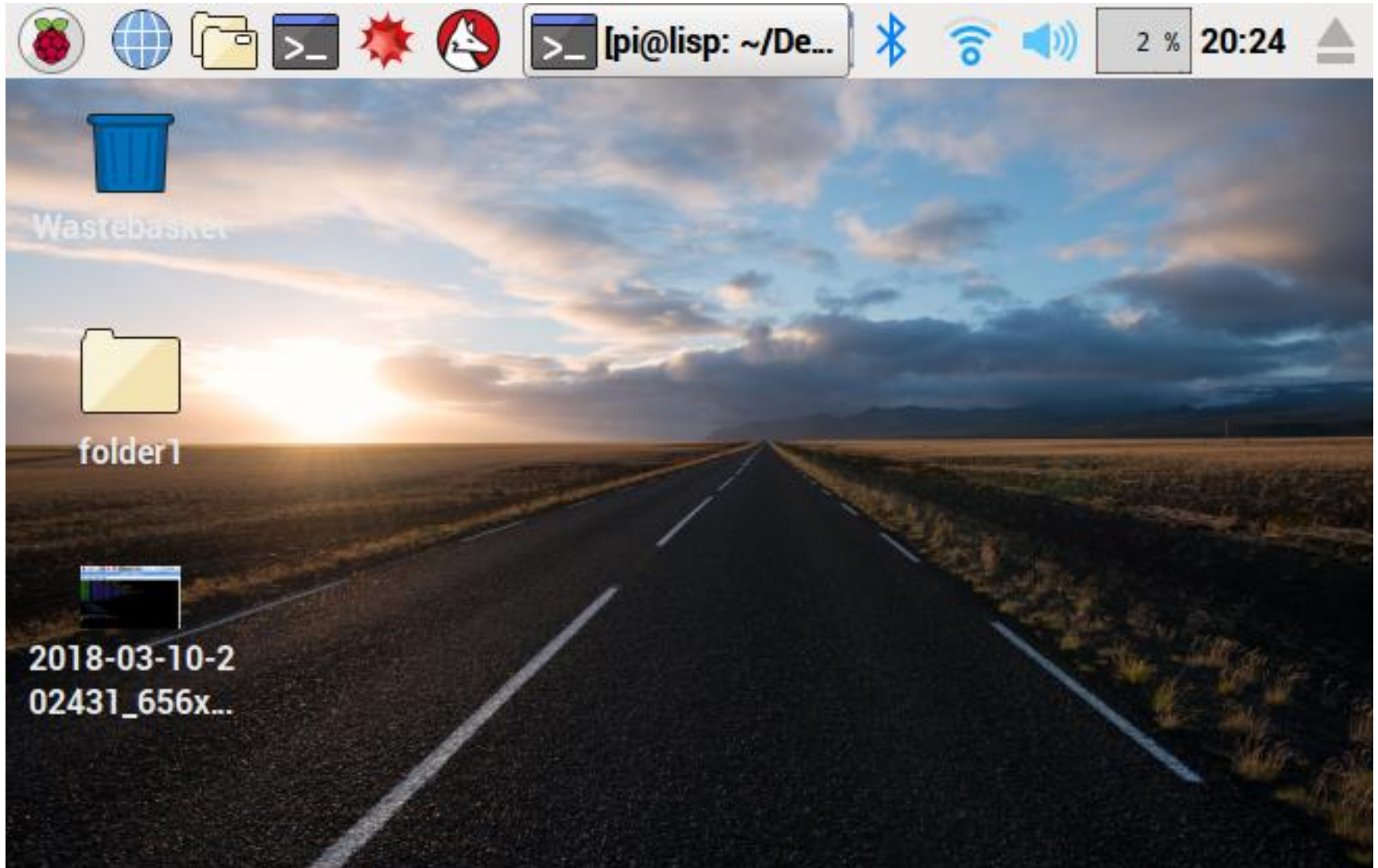
Below the tree output, it says '2 directories, 2 files' and then 'pi@lisp:~/Desktop !'.

„mkdir“ is
„make directory“
Guess what it does?!

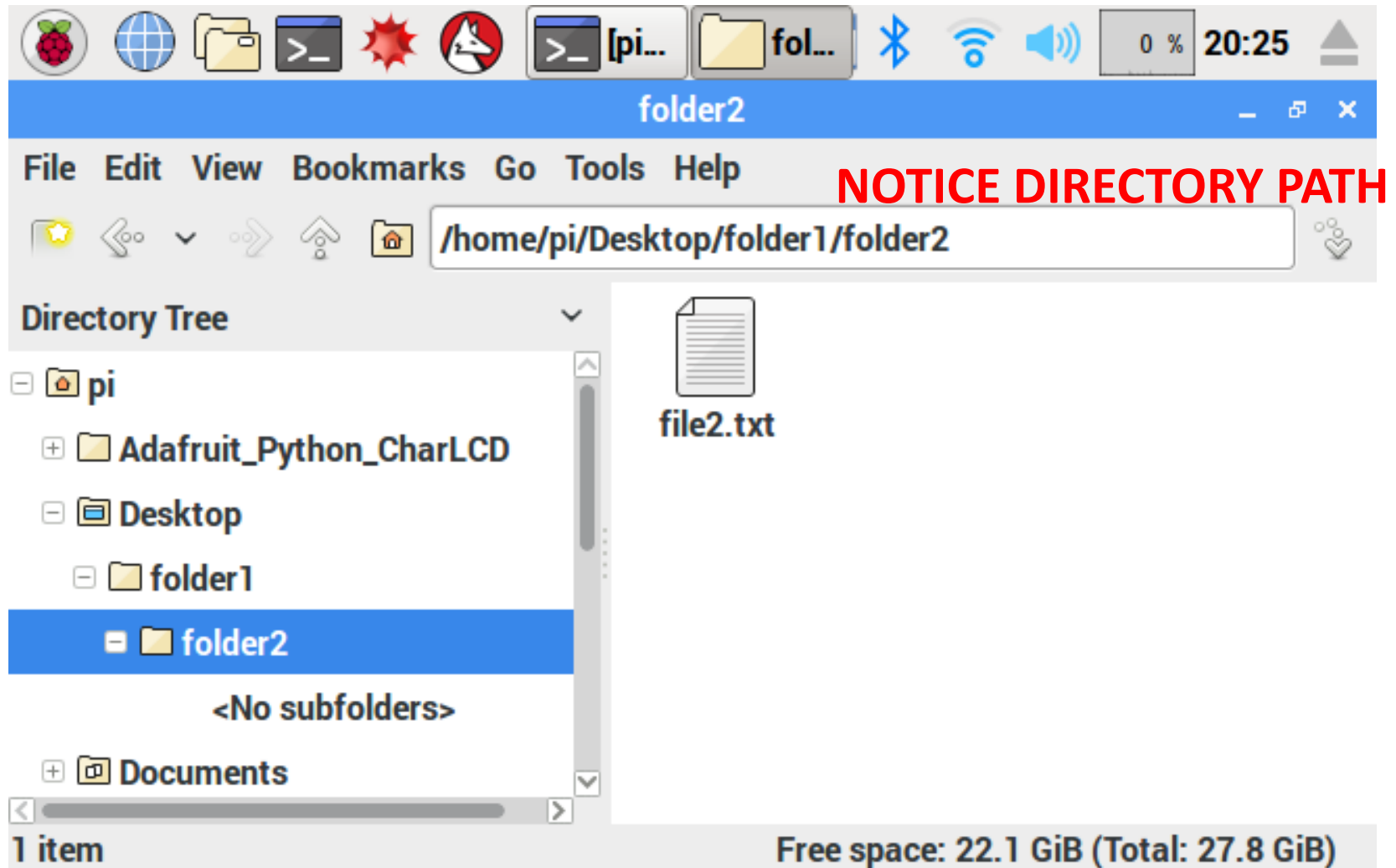
„cd“ changes the
current directory

„touch“ creates
an empty file

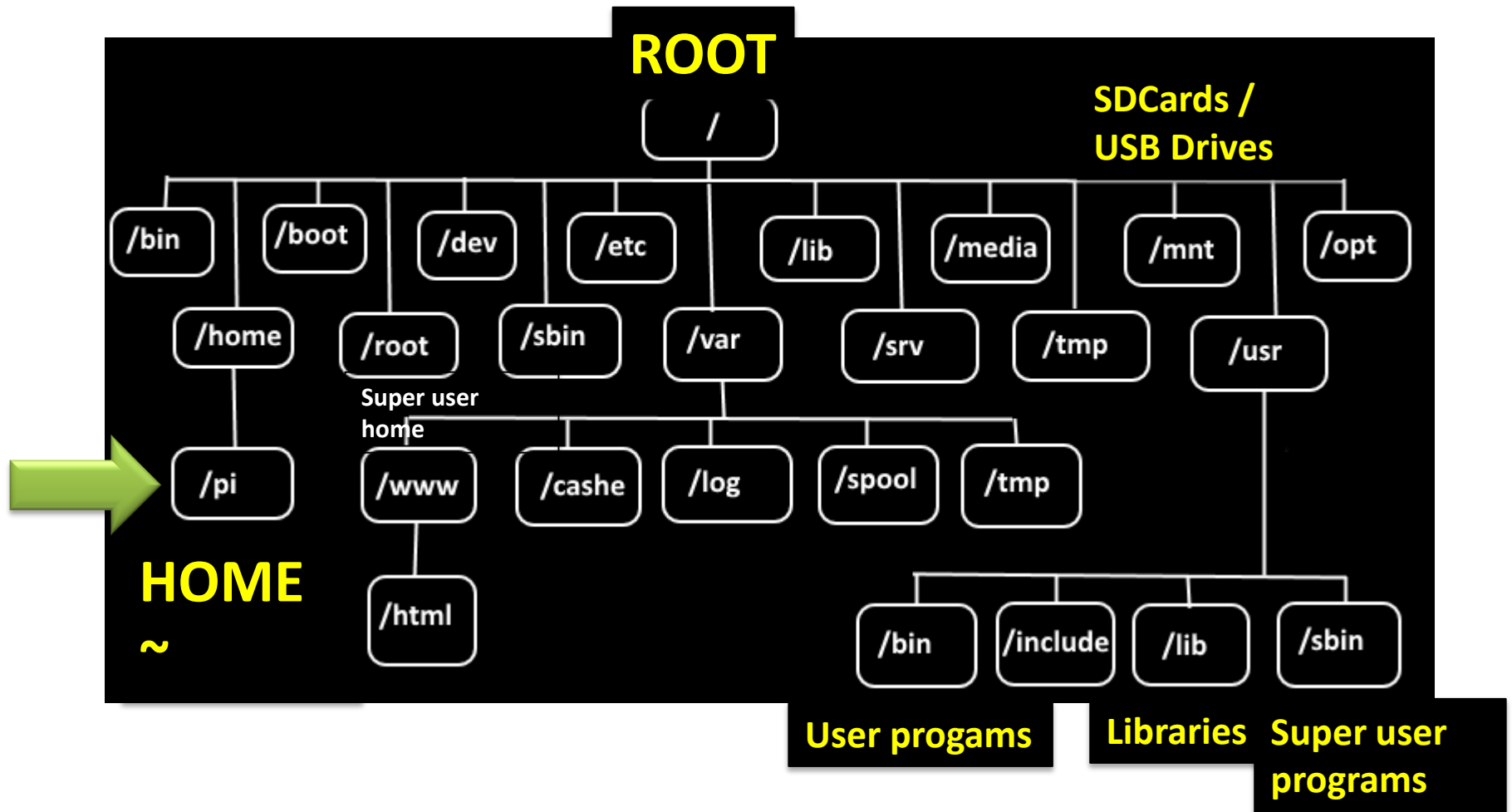
We created a folder on the desktop!



Let's look inside with the File Manager

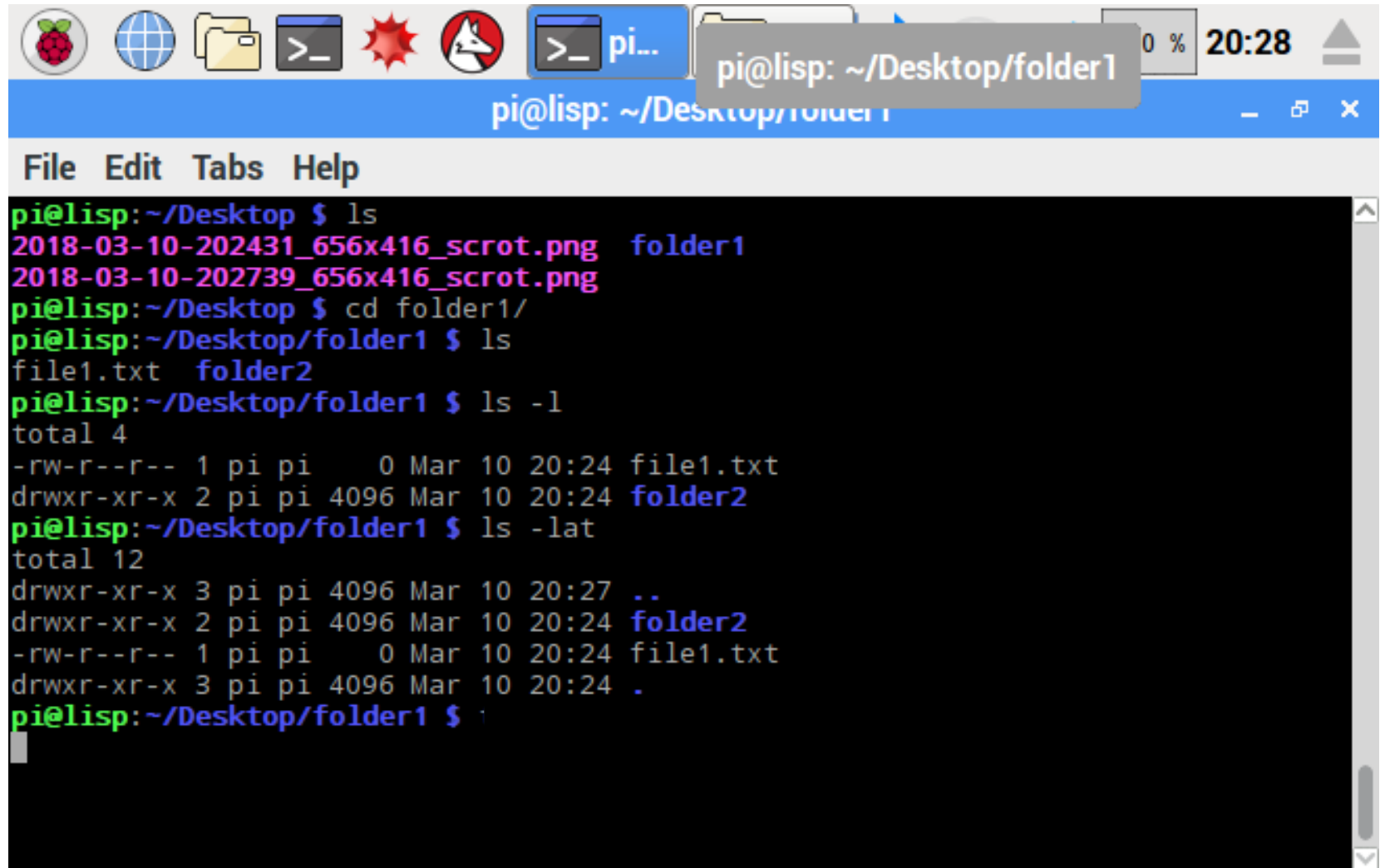


Raspbian Filesystem Tree



What's in a directory?

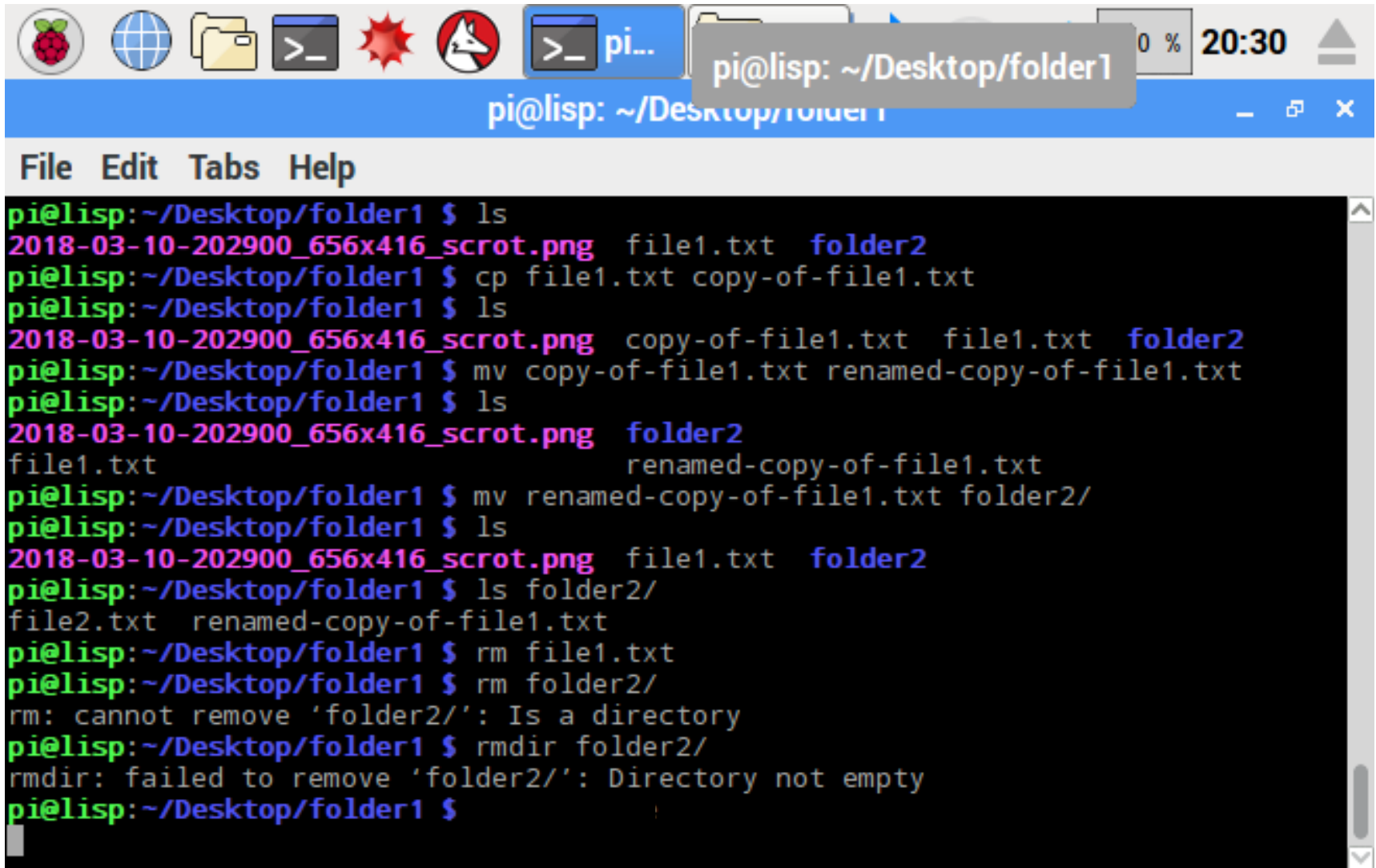
„ls“ to „list files“ – Files have Attributes



The screenshot shows a terminal window on a Raspberry Pi. The title bar indicates the user is 'pi' at 'lisp' in the directory '~/Desktop/folder1'. The terminal output shows the following commands and results:

```
pi@lisp:~/Desktop $ ls
2018-03-10-202431_656x416_scrot.png  folder1
2018-03-10-202739_656x416_scrot.png
pi@lisp:~/Desktop $ cd folder1/
pi@lisp:~/Desktop/folder1 $ ls
file1.txt  folder2
pi@lisp:~/Desktop/folder1 $ ls -l
total 4
-rw-r--r-- 1 pi pi    0 Mar 10 20:24 file1.txt
drwxr-xr-x 2 pi pi 4096 Mar 10 20:24 folder2
pi@lisp:~/Desktop/folder1 $ ls -lat
total 12
drwxr-xr-x 3 pi pi 4096 Mar 10 20:27 ..
drwxr-xr-x 2 pi pi 4096 Mar 10 20:24 folder2
-rw-r--r-- 1 pi pi    0 Mar 10 20:24 file1.txt
drwxr-xr-x 3 pi pi 4096 Mar 10 20:24 .
pi@lisp:~/Desktop/folder1 $
```

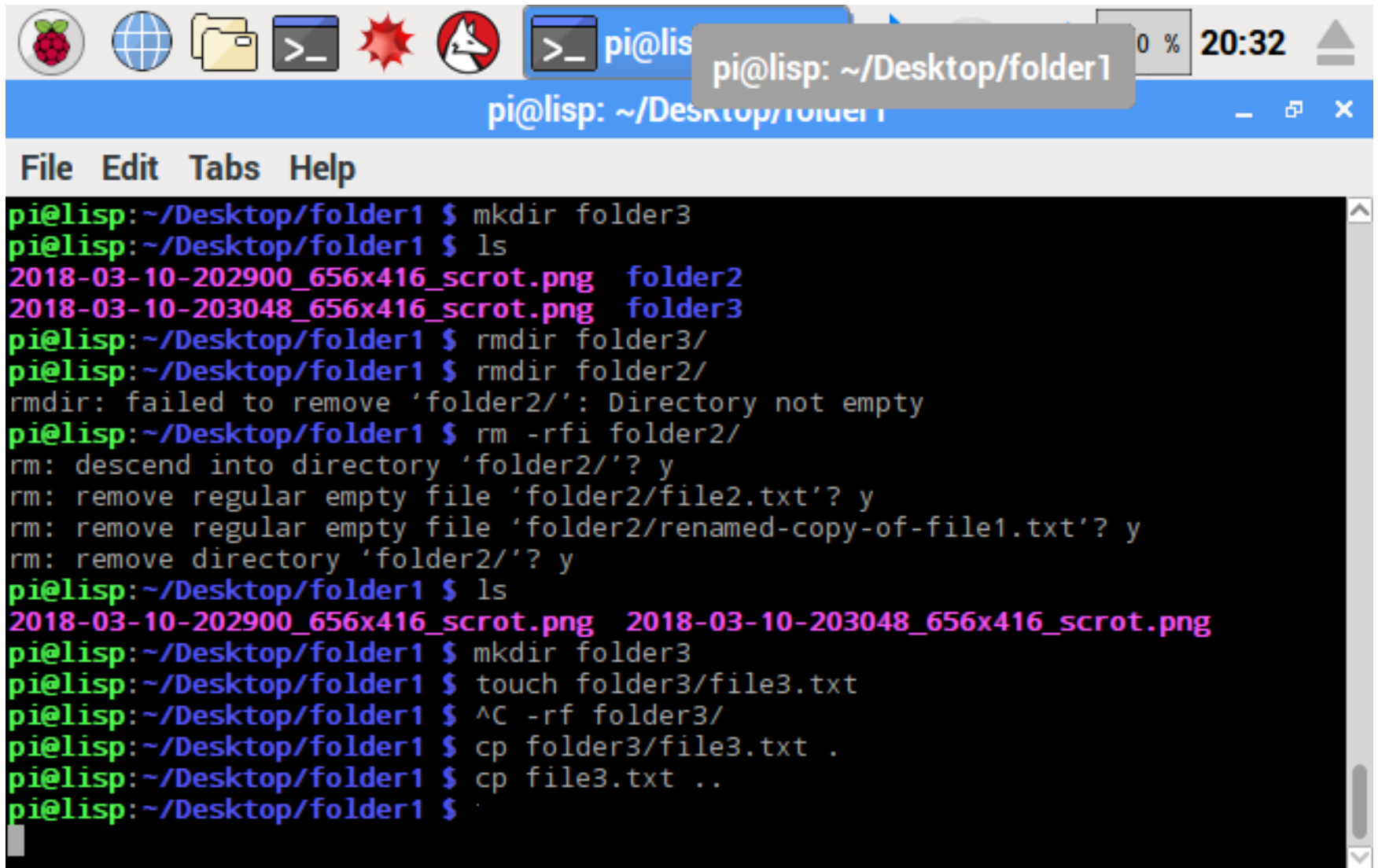
copy (cp), move (mv), remove (rm),
remove directory (rmdir)



The screenshot shows a terminal window titled "pi@lisp: ~/Desktop/folder1". The window has a blue title bar and a menu bar with "File", "Edit", "Tabs", and "Help". The terminal output shows the following commands and results:

```
pi@lisp:~/Desktop/folder1 $ ls
2018-03-10-202900_656x416_scrot.png file1.txt folder2
pi@lisp:~/Desktop/folder1 $ cp file1.txt copy-of-file1.txt
pi@lisp:~/Desktop/folder1 $ ls
2018-03-10-202900_656x416_scrot.png copy-of-file1.txt file1.txt folder2
pi@lisp:~/Desktop/folder1 $ mv copy-of-file1.txt renamed-copy-of-file1.txt
pi@lisp:~/Desktop/folder1 $ ls
2018-03-10-202900_656x416_scrot.png folder2
file1.txt renamed-copy-of-file1.txt
pi@lisp:~/Desktop/folder1 $ mv renamed-copy-of-file1.txt folder2/
pi@lisp:~/Desktop/folder1 $ ls
2018-03-10-202900_656x416_scrot.png file1.txt folder2
pi@lisp:~/Desktop/folder1 $ ls folder2/
file2.txt renamed-copy-of-file1.txt
pi@lisp:~/Desktop/folder1 $ rm file1.txt
pi@lisp:~/Desktop/folder1 $ rm folder2/
rm: cannot remove 'folder2/': Is a directory
pi@lisp:~/Desktop/folder1 $ rmdir folder2/
rmdir: failed to remove 'folder2/': Directory not empty
pi@lisp:~/Desktop/folder1 $
```

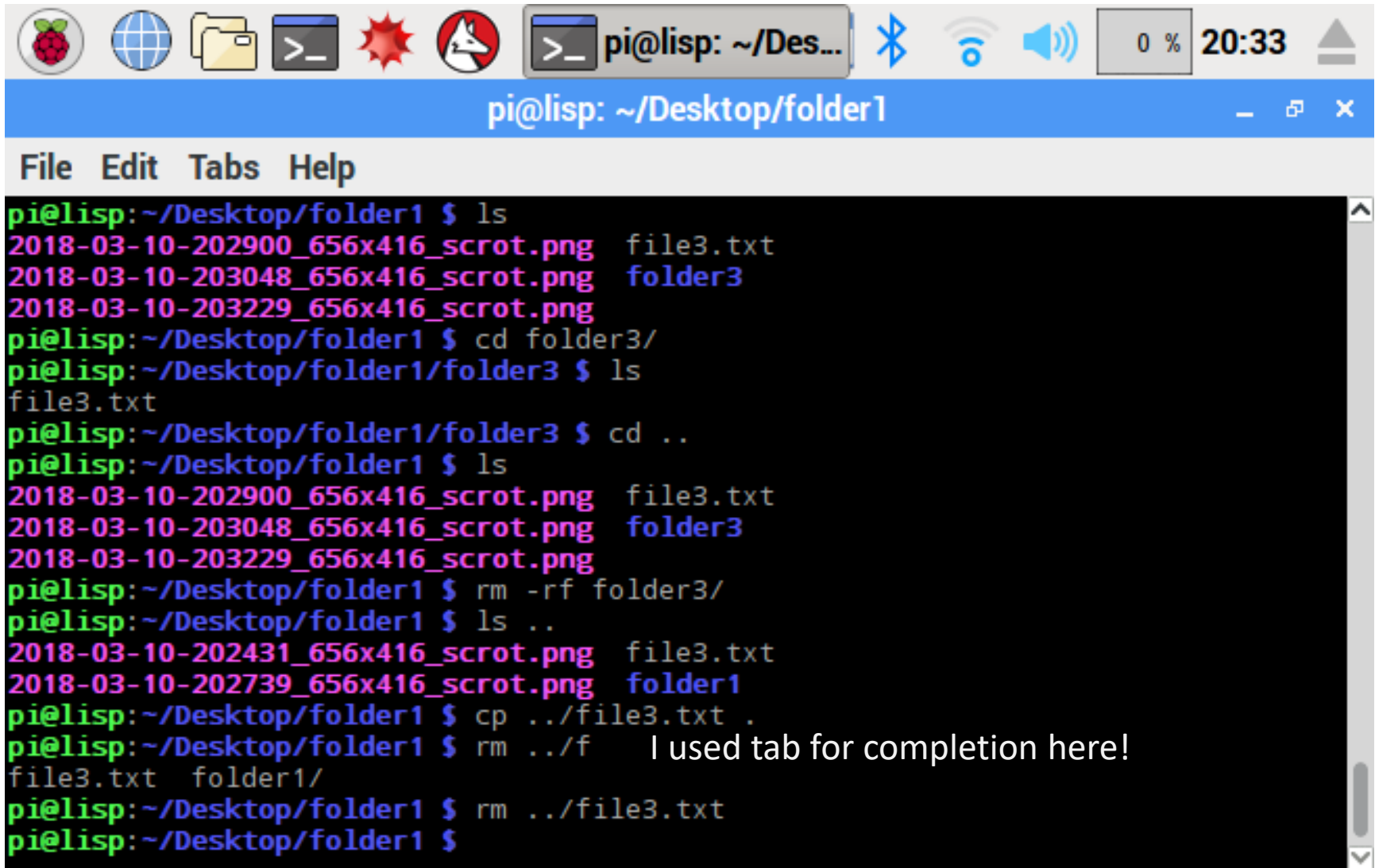
Deleting directories – „rmmdir“ and „rm -rfi“



The screenshot shows a terminal window titled "pi@lisp: ~/Desktop/folder1". The window has a menu bar with "File", "Edit", "Tabs", and "Help". The terminal output shows the following commands and their results:

```
pi@lisp:~/Desktop/folder1 $ mkdir folder3
pi@lisp:~/Desktop/folder1 $ ls
2018-03-10-202900_656x416_scrot.png  folder2
2018-03-10-203048_656x416_scrot.png  folder3
pi@lisp:~/Desktop/folder1 $ rmdir folder3/
pi@lisp:~/Desktop/folder1 $ rmdir folder2/
rmdir: failed to remove 'folder2/': Directory not empty
pi@lisp:~/Desktop/folder1 $ rm -rfi folder2/
rm: descend into directory 'folder2/'? y
rm: remove regular empty file 'folder2/file2.txt'? y
rm: remove regular empty file 'folder2/renamed-copy-of-file1.txt'? y
rm: remove directory 'folder2/'? y
pi@lisp:~/Desktop/folder1 $ ls
2018-03-10-202900_656x416_scrot.png  2018-03-10-203048_656x416_scrot.png
pi@lisp:~/Desktop/folder1 $ mkdir folder3
pi@lisp:~/Desktop/folder1 $ touch folder3/file3.txt
pi@lisp:~/Desktop/folder1 $ ^C -rf folder3/
pi@lisp:~/Desktop/folder1 $ cp folder3/file3.txt .
pi@lisp:~/Desktop/folder1 $ cp file3.txt ..
pi@lisp:~/Desktop/folder1 $
```

Using „.“ and „..“ in commands

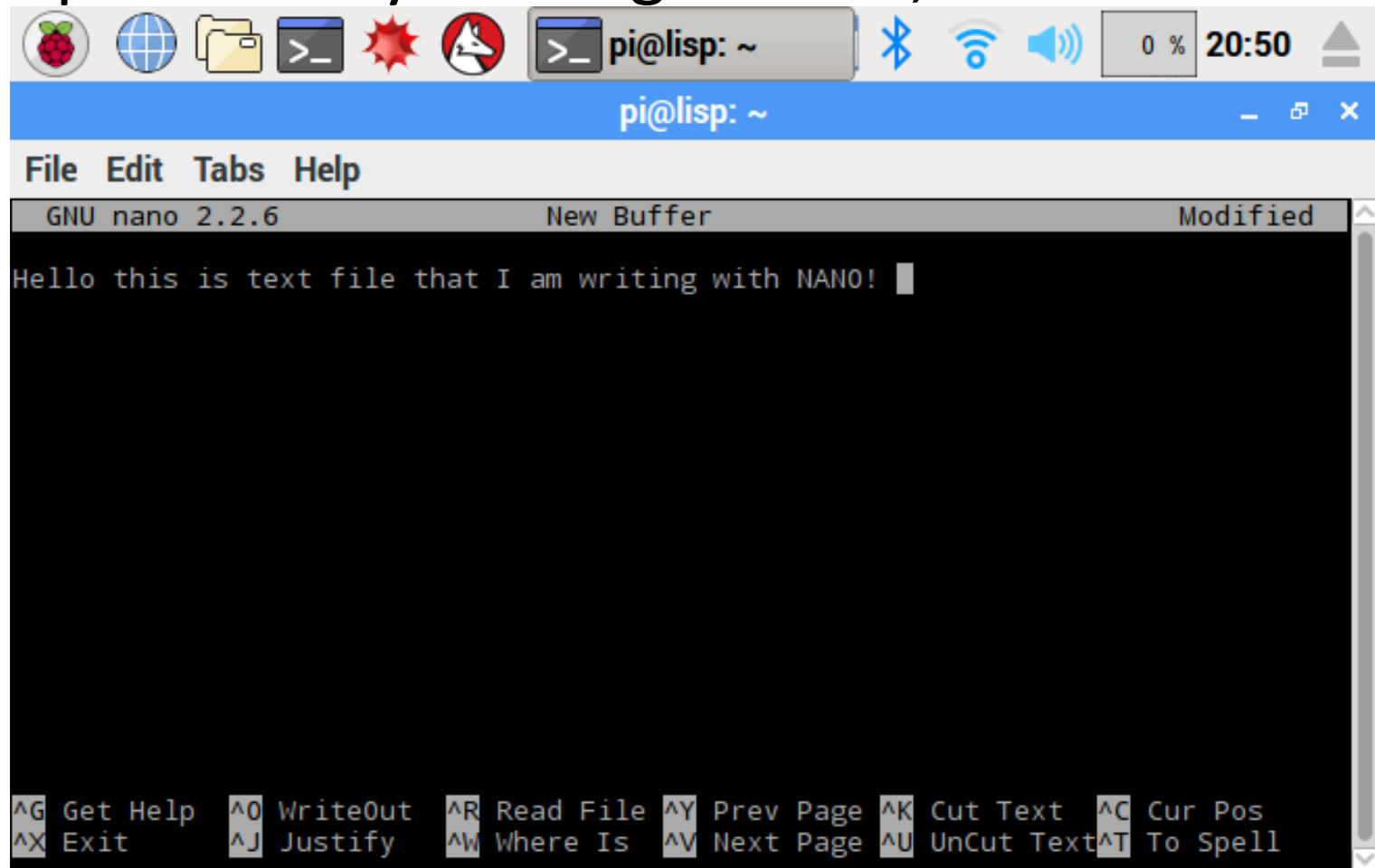


The screenshot shows a terminal window on a Raspberry Pi. The title bar indicates the user is 'pi@lisp' in the directory '~/Desktop/folder1'. The terminal content shows a series of commands and their outputs:

```
pi@lisp:~/Desktop/folder1 $ ls
2018-03-10-202900_656x416_scrot.png  file3.txt
2018-03-10-203048_656x416_scrot.png  folder3
2018-03-10-203229_656x416_scrot.png
pi@lisp:~/Desktop/folder1 $ cd folder3/
pi@lisp:~/Desktop/folder1/folder3 $ ls
file3.txt
pi@lisp:~/Desktop/folder1/folder3 $ cd ..
pi@lisp:~/Desktop/folder1 $ ls
2018-03-10-202900_656x416_scrot.png  file3.txt
2018-03-10-203048_656x416_scrot.png  folder3
2018-03-10-203229_656x416_scrot.png
pi@lisp:~/Desktop/folder1 $ rm -rf folder3/
pi@lisp:~/Desktop/folder1 $ ls ..
2018-03-10-202431_656x416_scrot.png  file3.txt
2018-03-10-202739_656x416_scrot.png  folder1
pi@lisp:~/Desktop/folder1 $ cp ../file3.txt .
pi@lisp:~/Desktop/folder1 $ rm ../f    I used tab for completion here!
file3.txt  folder1/
pi@lisp:~/Desktop/folder1 $ rm ../file3.txt
pi@lisp:~/Desktop/folder1 $
```

NANO Text Editor

- Start with „nano“ – from the Shell
- Important key bindings: Ctrl-X, etc. See below:



```
pi@lisp: ~  
File Edit Tabs Help  
GNU nano 2.2.6 New Buffer Modified  
Hello this is text file that I am writing with NANO!  
  
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

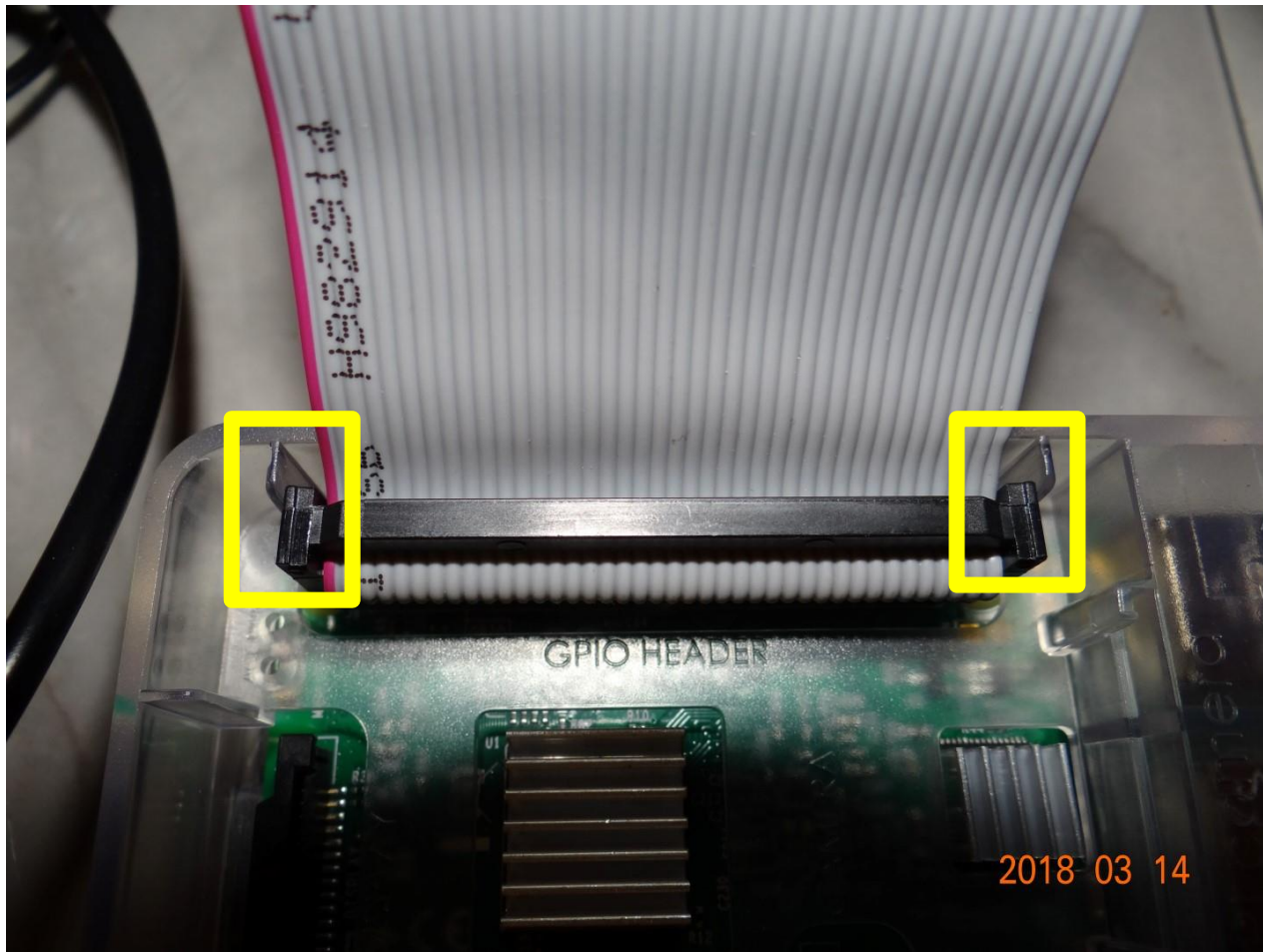
Other Useful Commands

- Space free on disk:
df -h
- Look at a text file:
cat file1.txt
Better: use **more** program if more than one screen! Use „q“ to quit **more**
- Run a program as „super user = root“:
sudo <command here>
E.g., to reboot or halt RPi:
sudo restart, sudo halt
- **sudo** also needed for Raspberry Pi updates!

Plug in Breadboard Cobbler



Notice !



Plug in Breadboard Cobbler

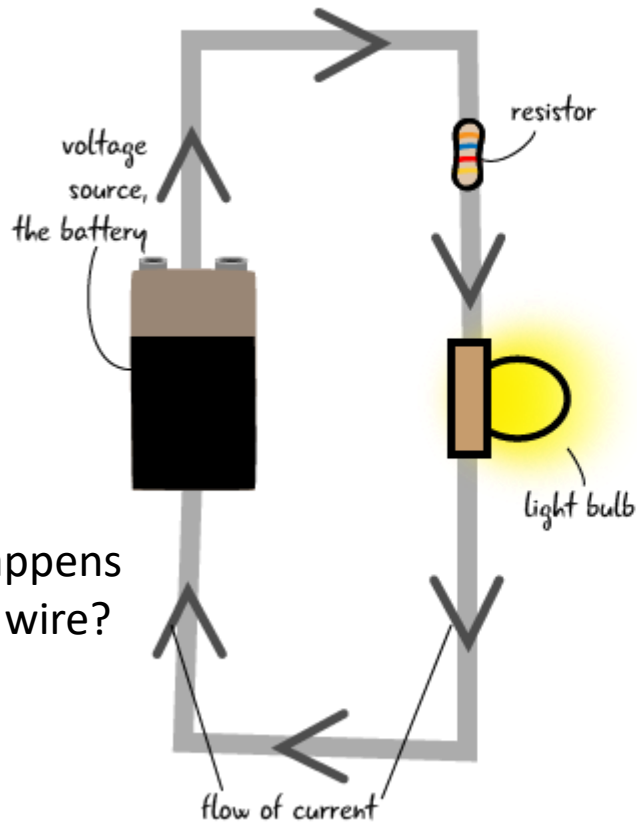


Lid on Top Fits - Gap for Cable



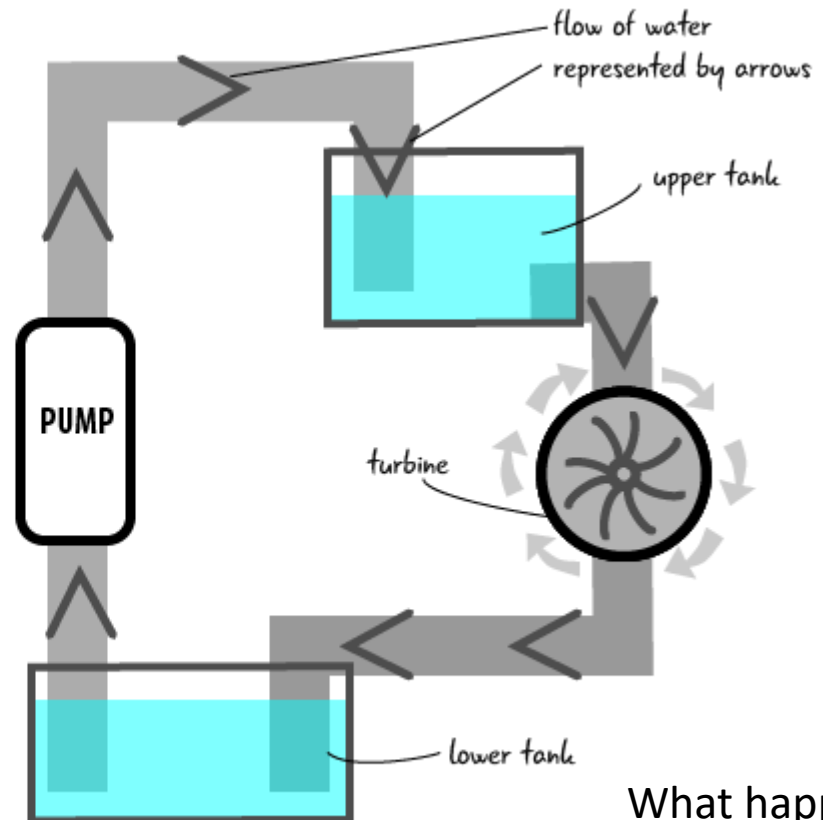
Electric Circuit & Water Analogy

water analogy for electricity



What happens if I cut a wire?

flow of current, represented by arrows
flow of charges, flow of electrons
caused by battery = voltage source

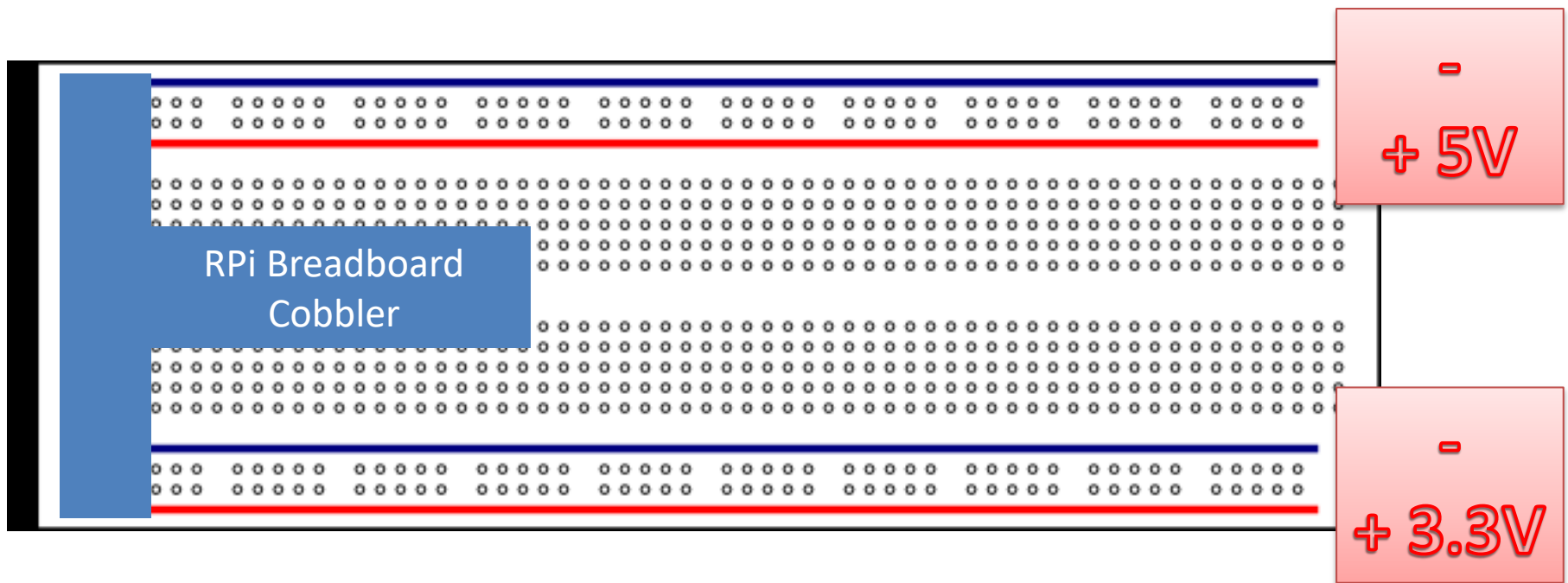


What happens if I block the pipe?

flow of water caused by pump

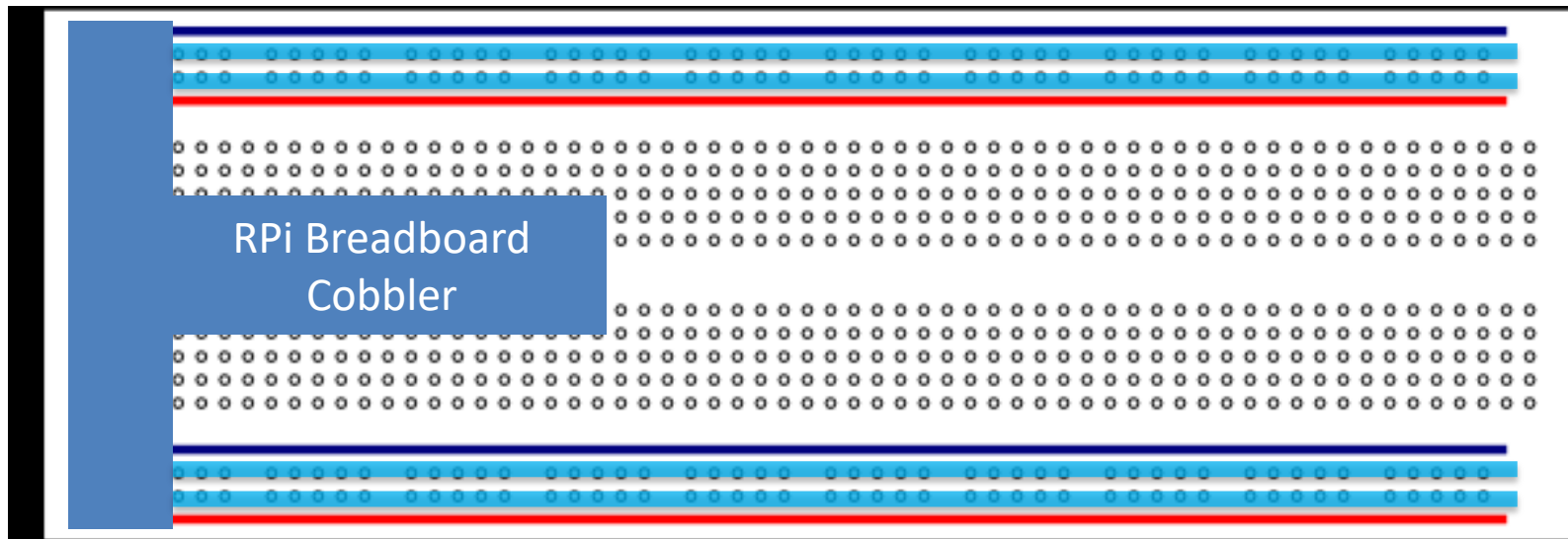
Raspberry Pi

- BLUE = „MINUS“ or „GROUND“ (GND), 0V
- RED = „PLUS“, 3.3 V or 5.0 V



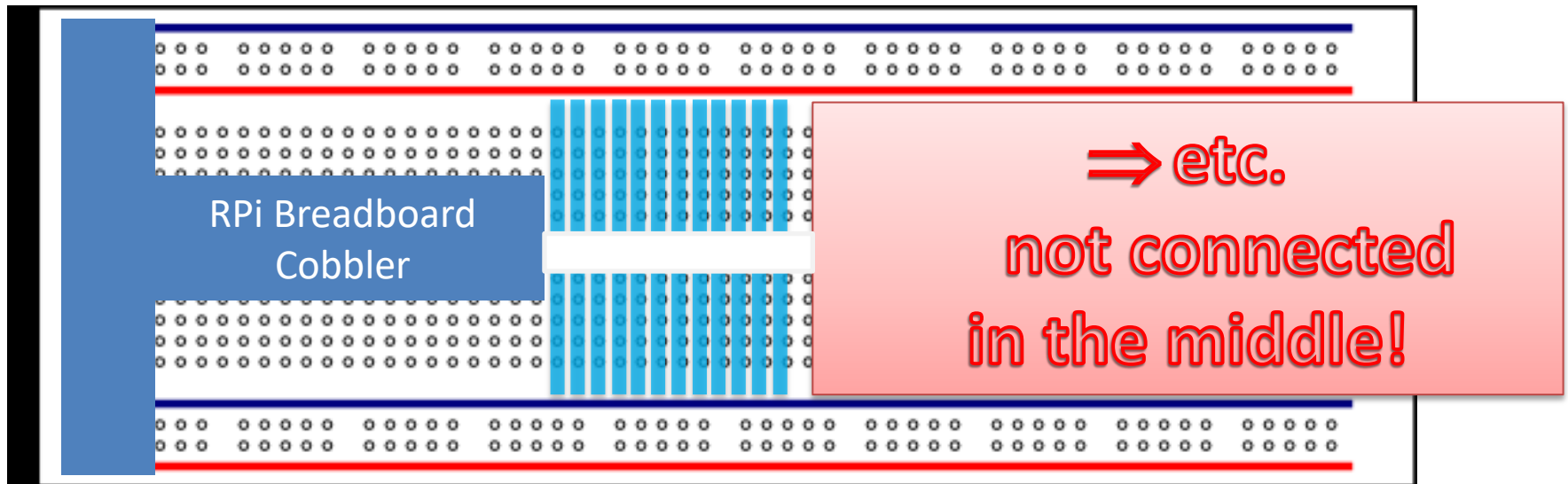
Row Connections

- The holes on the power / voltage lines are connected horizontally as shown



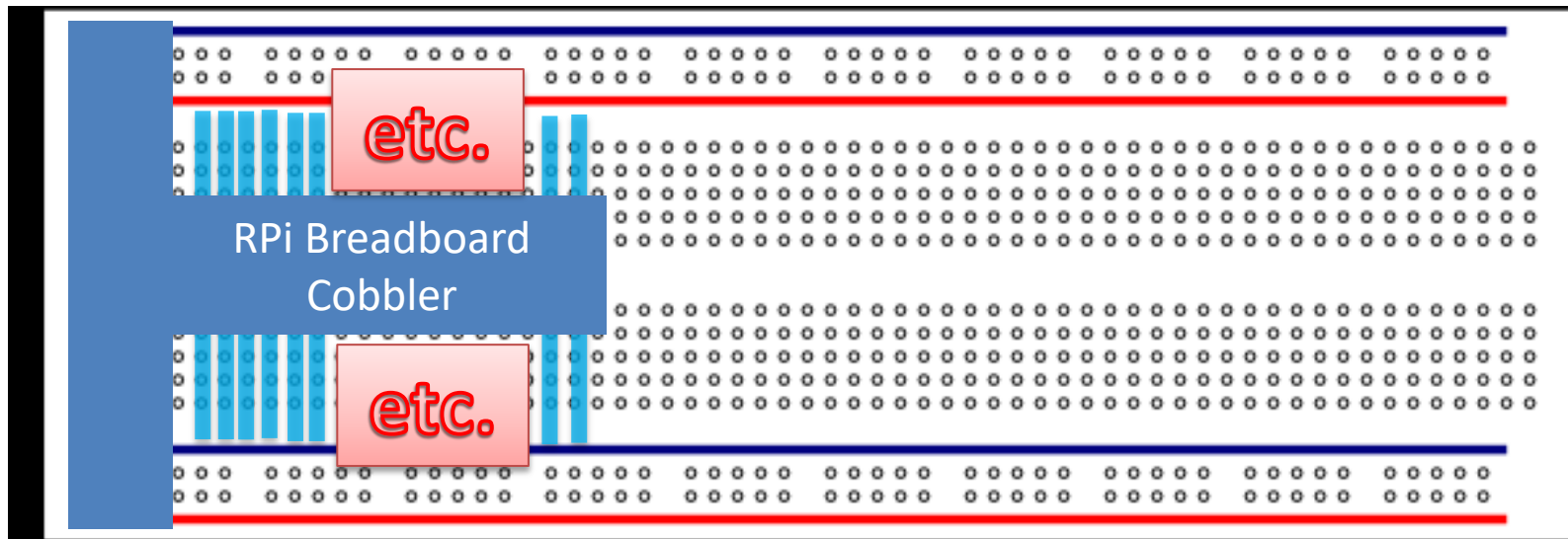
Column Connections

- The other holes are connected vertically as shown, throughout the board



Connections to Rpi GPIOs – Use Later

- This is how we connect to the GPIOs of the Rpi
- For now, we only use the RPi as our „battery“

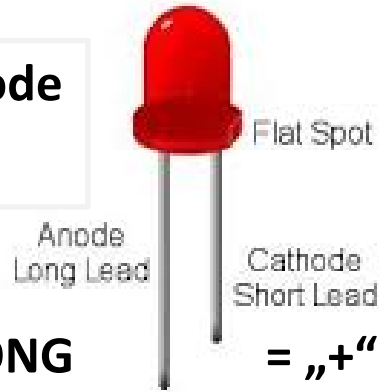


Careful with GPIO Ports !

- **Never**
 - **Connect 5 V or 3.3 V directly to MINUS (GND)**
Connect PLUS directly to MINUS (GND)
-> SHORT CIRCUIT
 - Do not connect output directly to 5 V or 3.3 V
 - Do not „draw“ more than 50 mA current
(eg, do not try to connect a motor to a GPIO pin!)
 - A GPIO pin can ONLY power a LED or relay or the like,
not much more
- And careful with static electricity / discharge!

Let's Make a Circuit – LED & Button Components Needed

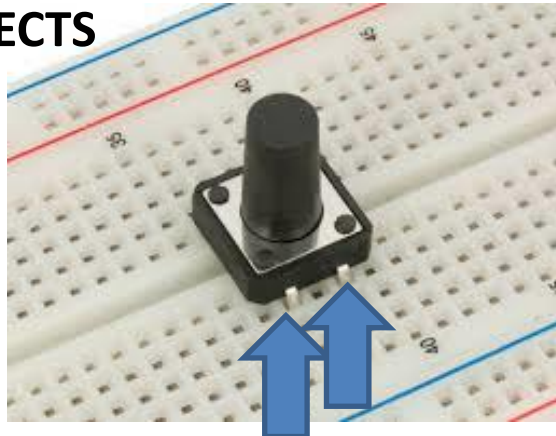
LED = Light Emitting Diode



ANODE = „+“ PIN = LONG
CATHODE = „-“ PIN = SHORT

= „+“ = MORE
= „-“ = LESS

„MOMENTARY PUSH BUTTON“
WHEN PUSHED, CONNECTS
THE TWO PINS



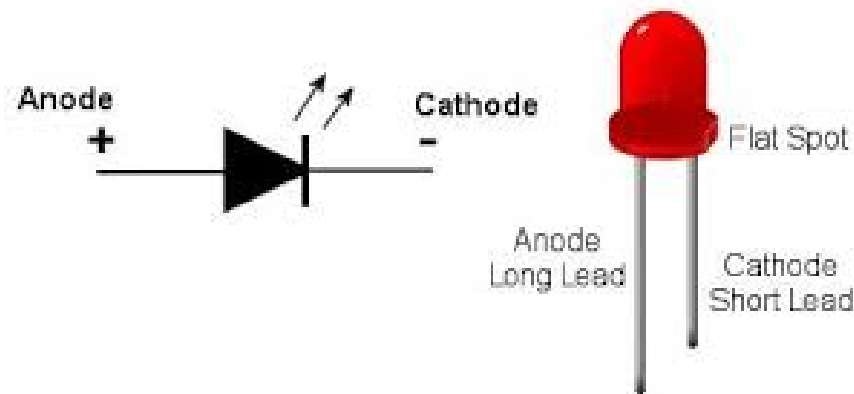
RESISTOR = LIMITS CURRENT

PROTECTS OUR LED !
LED NEED TO HAVE ~ 1.3 V
OTHERWISE IT BURNS OUT
IMMEDIATELY

We need one from the
10 Resistor set! (220 Ohm)

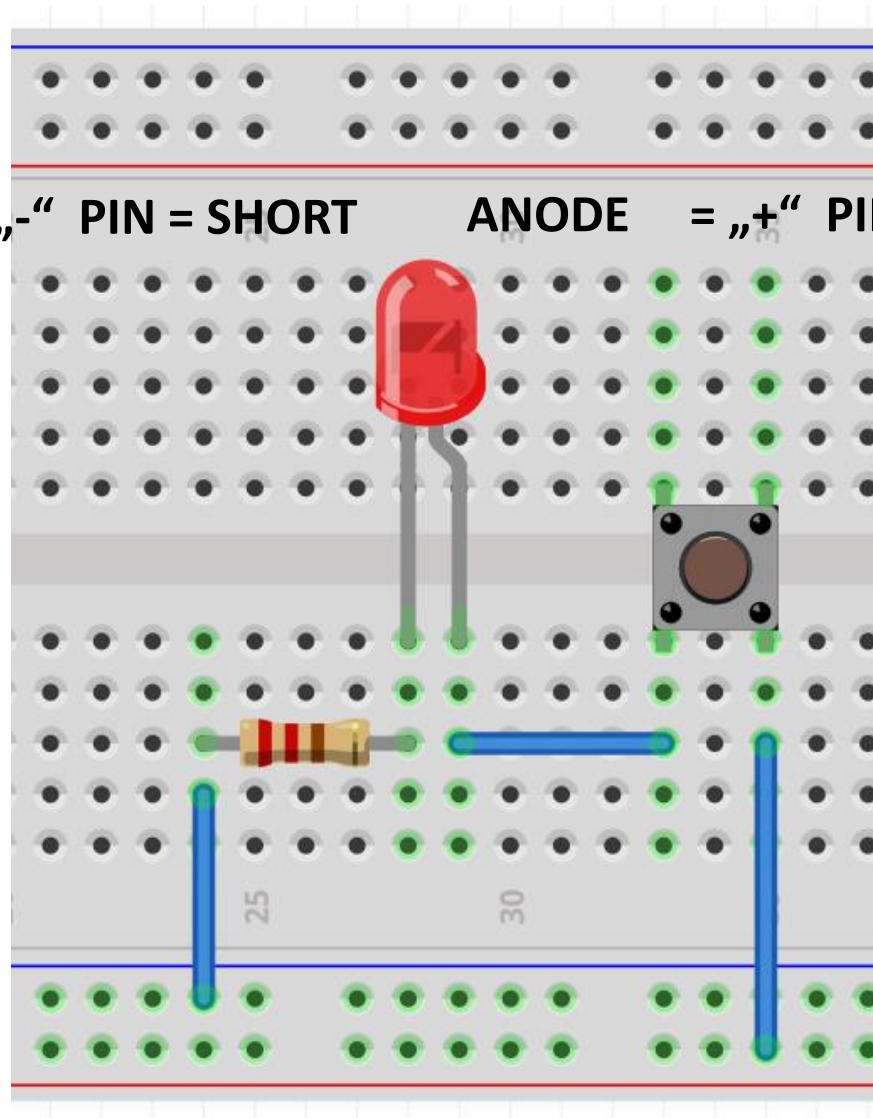
The LED is a „One Way Road“...

- ... for electricity / electric current / charges / electrons !
 - Current flows from ANODE to CATHODE
 - In direction of the ARROW
 - From „PLUS“ (= LONG) to „MINUS“ (= SHORT)



Solution

CATHODE = „-“ PIN = SHORT ANODE = „+“ PIN = LONG



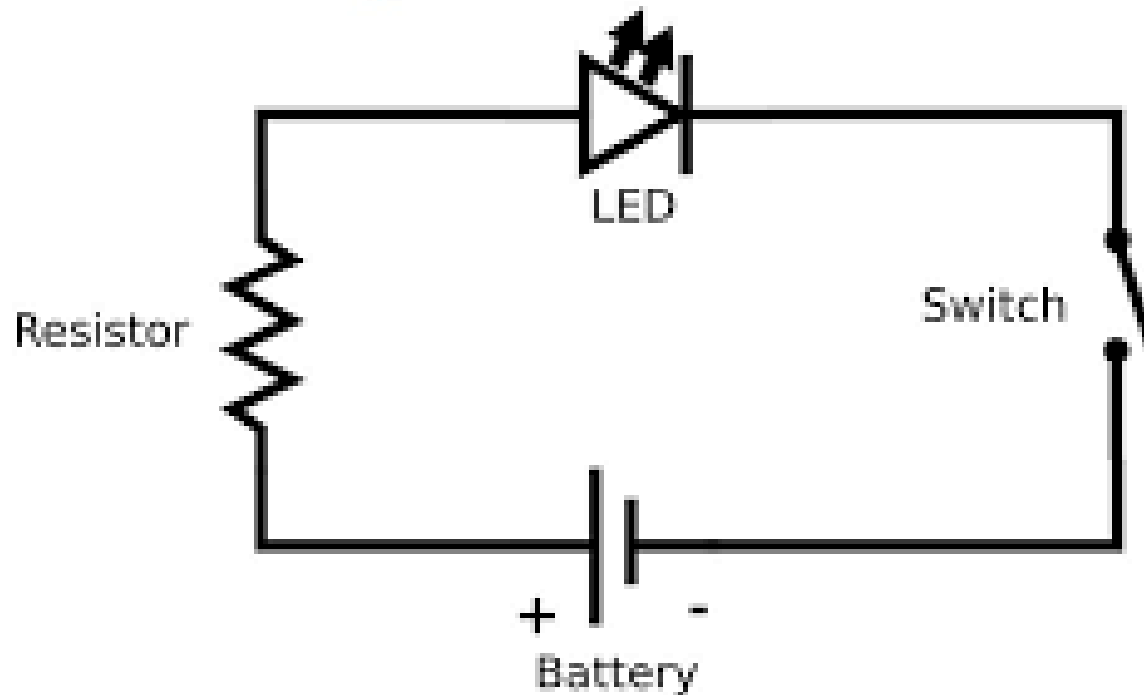
MINUS = BLUE = GND

PLUS = RED = 3.3 Volt

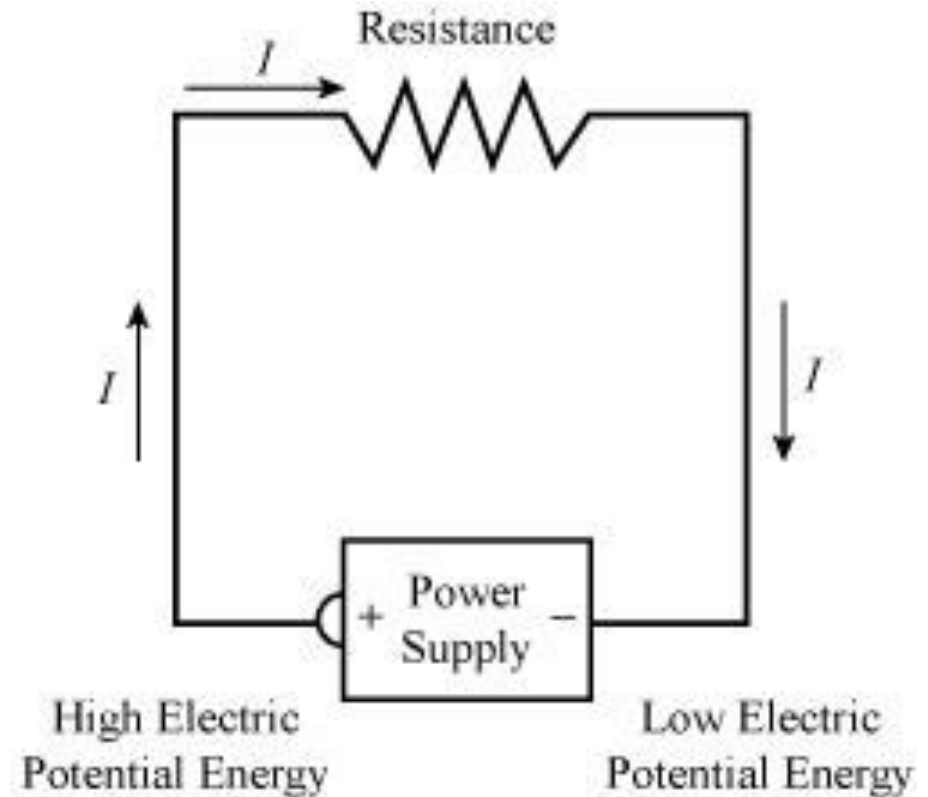
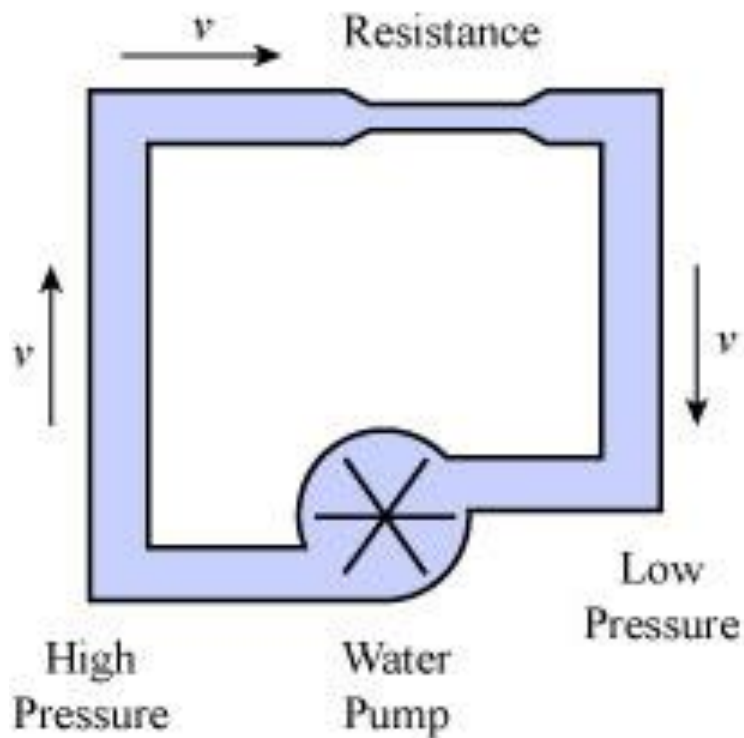


What's different in ours?
Does it matter?

Circuit Diagram

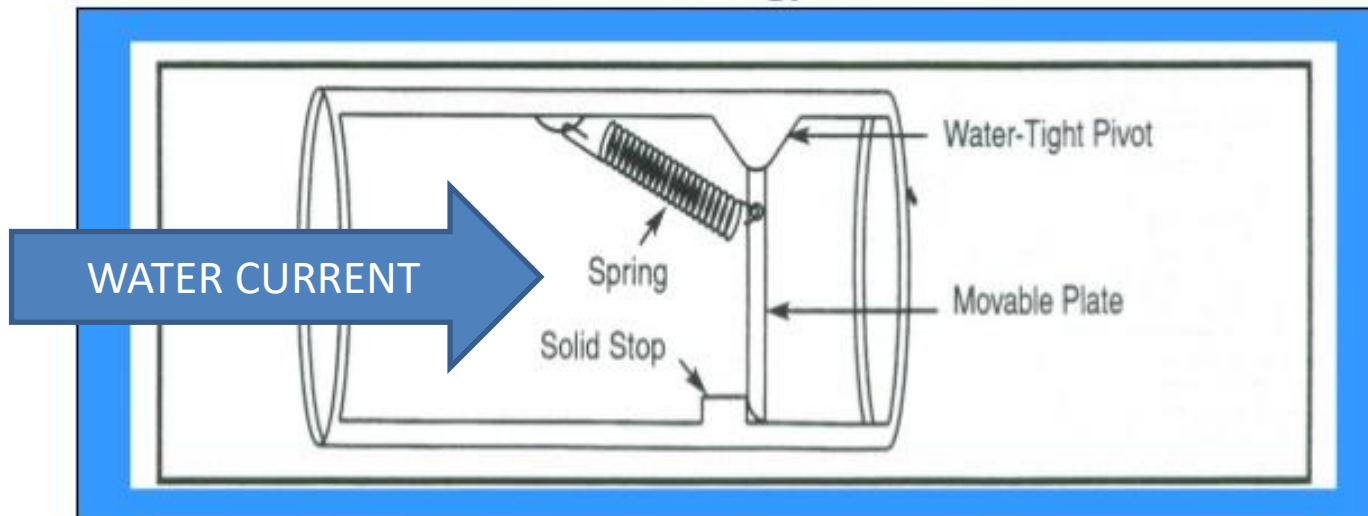


Water Analog for Resistor



Water Analogy for Diode

Water Analogy of Diodes



- When water pressure on left overcomes the restoring force of spring, the gate is opened and water is allowed to flow →.
- When water pressure is from right to left, the gate is pressed against the solid stop and no water is allowed to flow.
- Spring restoring force is analogous to 0.6V needed to forward bias a Si diode.

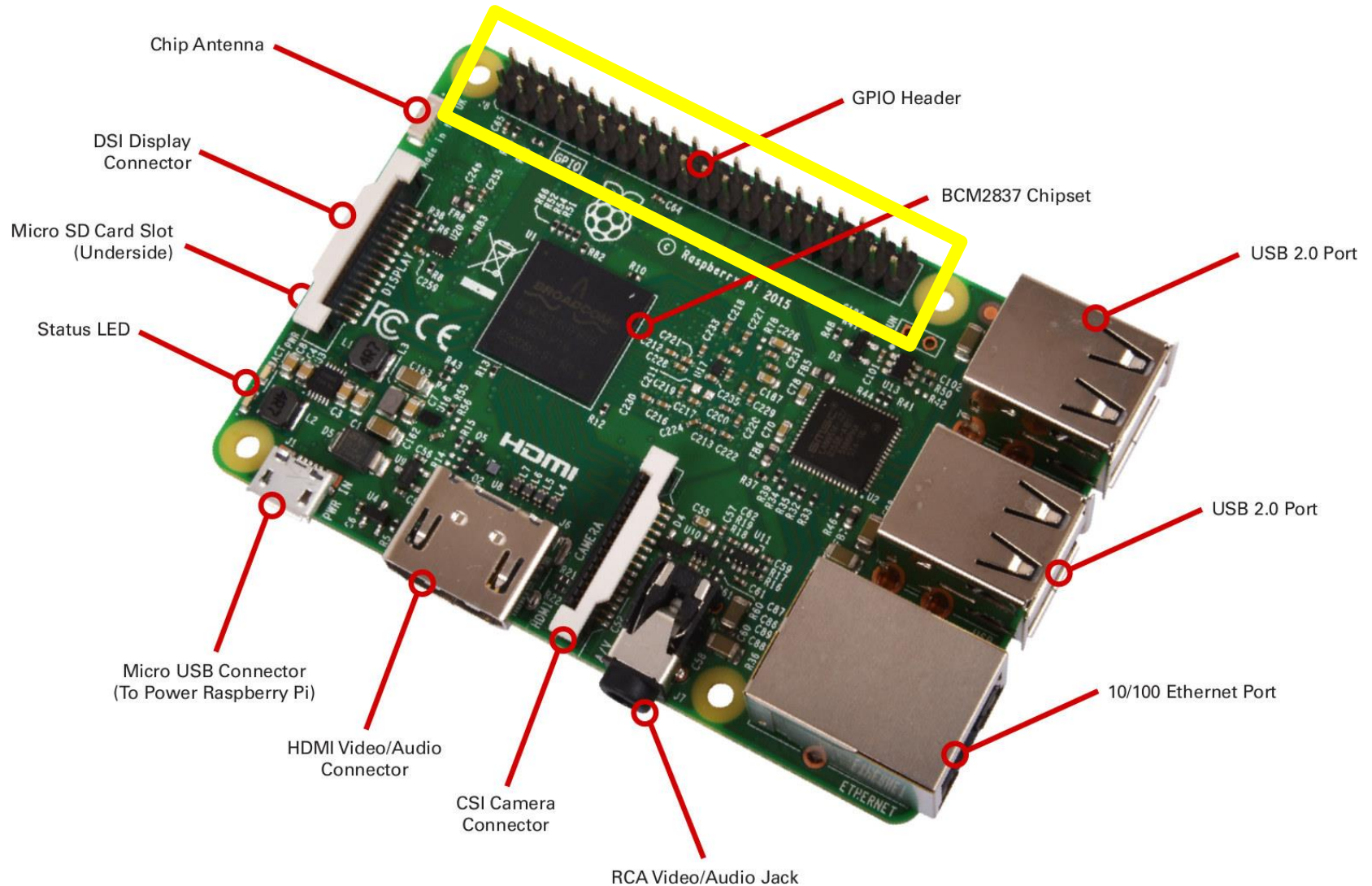
Experiment... What happens when we use 5 V instead of 3.3 V ?



Answer

- Nothing!
 - There is no change in brightness of the LED
 - Unlike a light bulb
 - LEDs cannot be „dimmed“ liked that
 - But there is a way...
We will learn later how to dimm LEDs...
- LEDs are very much „binary“ = on / off = 1 / 0 = high / low devices!
 - A traditional lightbulb is more „analog“
Here, Brightness can be changed by changing Voltage / Current
 - Not with LEDs (or only very slightly)

General Purpose Input Output Ports (GPIO)

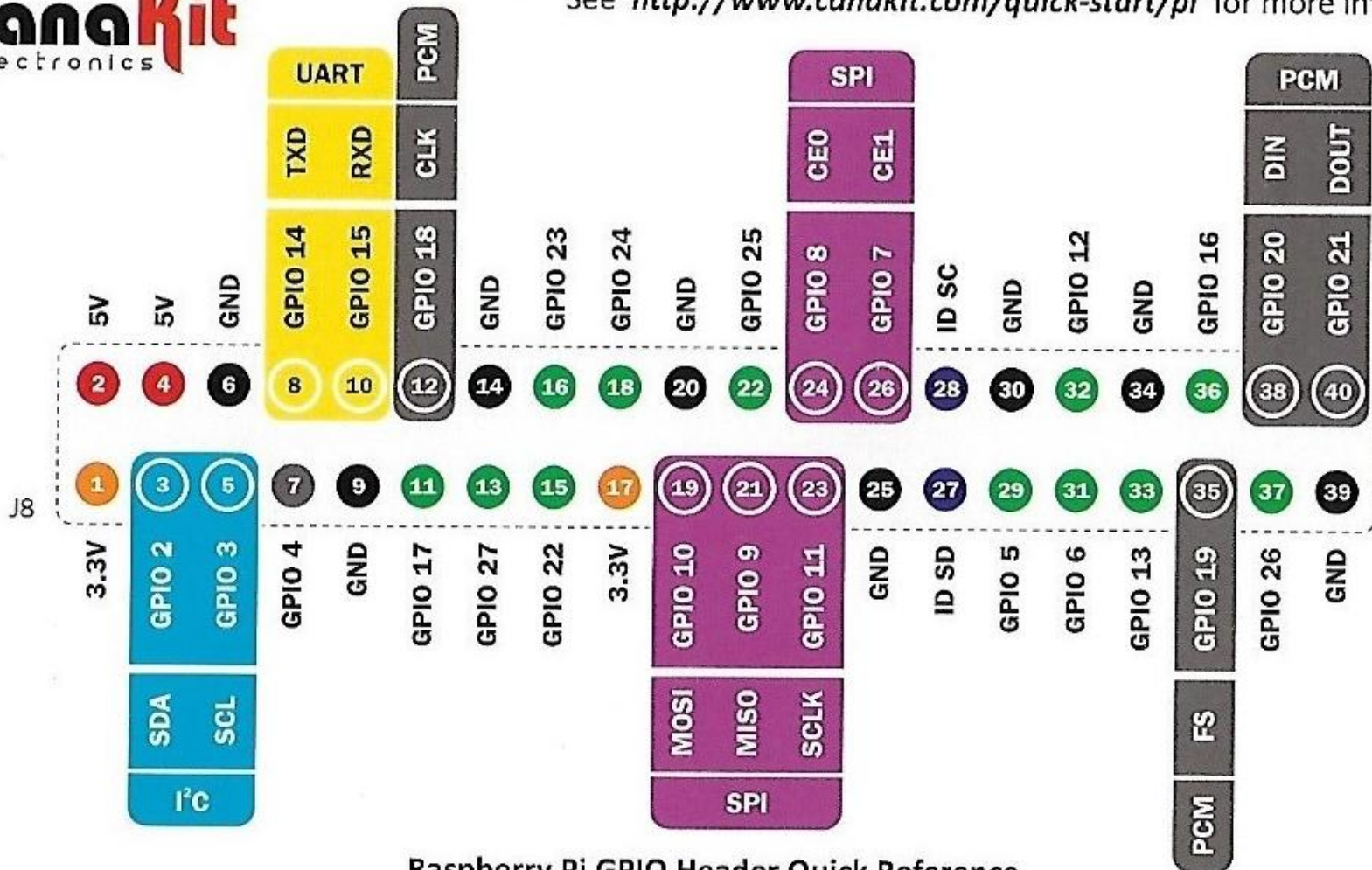


Functions of GPIO Pins

Also see <https://pinout.xyz/#>



See <http://www.canakit.com/quick-start/pi> for more information



Raspberry Pi GPIO Header Quick Reference

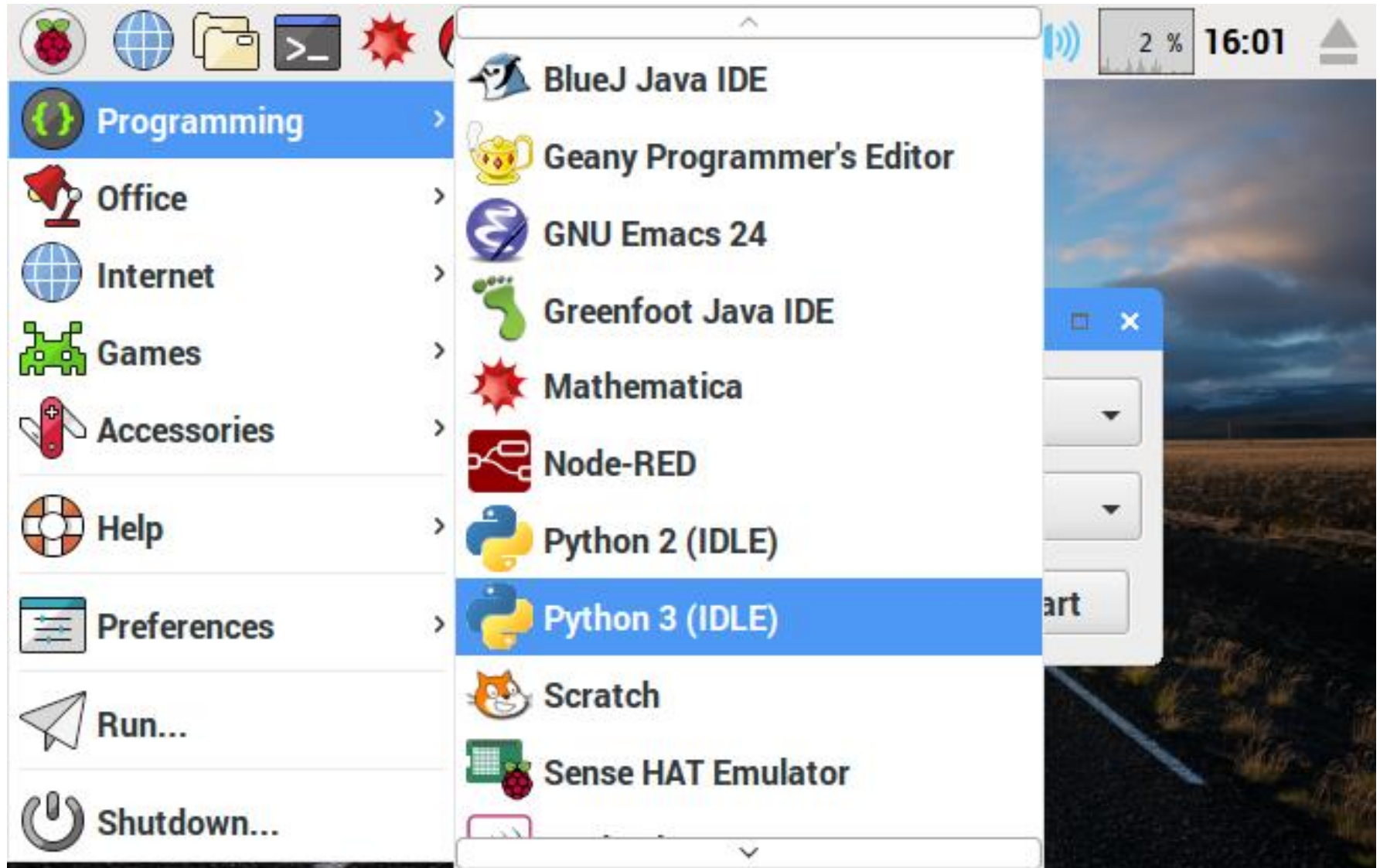
Key Features

- **GPIO Port = „General Purpose Input Output“ Port**
 - 40 pin „header“
 - Digital inputs and outputs (direction per pin configurable)
 - PWM = „Pulse Width Modulation“
 - SPI, I2C: Industry Standards Serial Protocols
 - Serial Peripheral Interface (SPI) – for Sensors, ...
 - 3 or 4 wires
 - Multiple devices with „select device“ signal
 - One master only
 - Full duplex
 - Longer ranges
 - I2C = I Squared C = I Two C = IIC = I²C (Inter-Integrated Circuit)
 - 2 wires only
 - Multiple devices on same bus, with chip addressing
 - Multiple masters possible
 - Cheaper
 - Half duplex
 - I2C more flexible, SPI faster
 - GPIOs need to be treated with care though; can be damaged by static electricity!
- **However, no ANALOG inputs / outputs (unlike Arduino)**

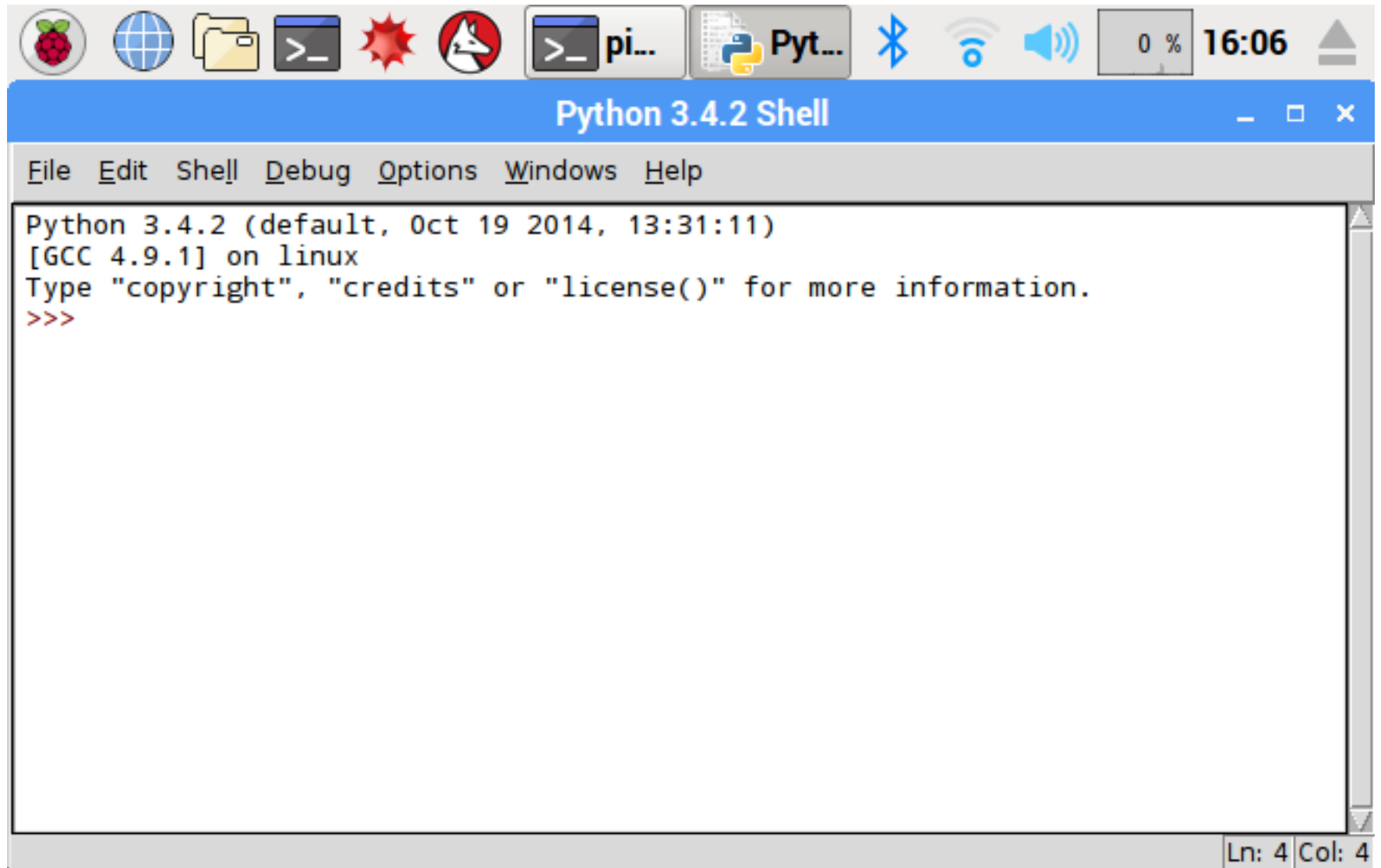
Careful with GPIO Ports !

- **Never:**
 - **Connect 5 V or 3.3 V directly to MINUS (GND)**
Connect PLUS directly to MINUS (GND)
-> SHORT CIRCUIT
 - **Connect GPIO PIN in OUTPUT MODE directly to MINUS (GND) – if the PIN = HIGH = 3.3 V, then this is a SHORT CIRCUIT, too**
 - It is OK if pin is set to INPUT mode, though
 - For OUTPUT mode, one always needs a „consumer“ – i.e., and LED and a resistor, otherwise it is a short circuit

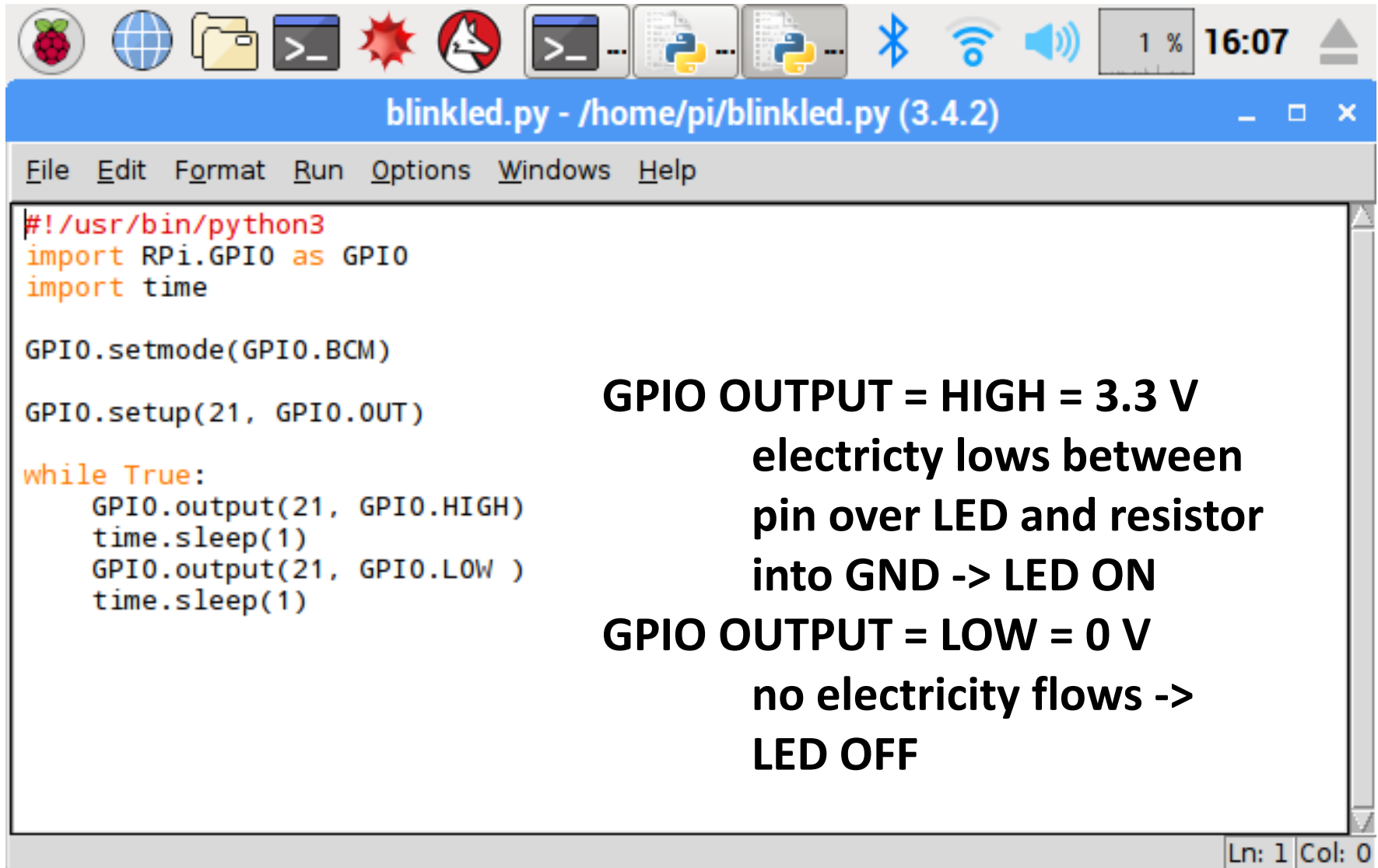
... on to Python GPIO Programming



Python Shell / Console / ...



Editor – Use Function-F5 / Ctrl-c



The screenshot shows a Raspberry Pi desktop environment with a terminal window titled "blinkled.py - /home/pi/blinkled.py (3.4.2)". The terminal displays a Python script that configures GPIO pin 21 as an output and blinks an LED by toggling the pin state between HIGH and LOW with 1-second delays. To the right of the terminal, text explains the logic: HIGH (3.3V) turns the LED on, and LOW (0V) turns it off. The terminal window has a menu bar with File, Edit, Format, Run, Options, Windows, and Help. The status bar at the bottom right shows "Ln: 1 Col: 0".

```
#!/usr/bin/python3
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

GPIO.setup(21, GPIO.OUT)

while True:
    GPIO.output(21, GPIO.HIGH)
    time.sleep(1)
    GPIO.output(21, GPIO.LOW )
    time.sleep(1)
```

GPIO OUTPUT = HIGH = 3.3 V
electricity flows between
pin over LED and resistor
into GND -> LED ON

GPIO OUTPUT = LOW = 0 V
no electricity flows ->
LED OFF

Ln: 1 Col: 0

Improvements

- Clean up source code
 - Write a couple of functions, „initialize“, „led_on“, „led_off“, „main“
 - Write a „main“ function that takes an argument
 - Time delay in milli seconds
 - Read in delay from keyboard using:
delay = int(input(„Delay?“))
- Also, use GPIO.cleanup() at the end

try:

<code>

finally:

GPIO.cleanup()