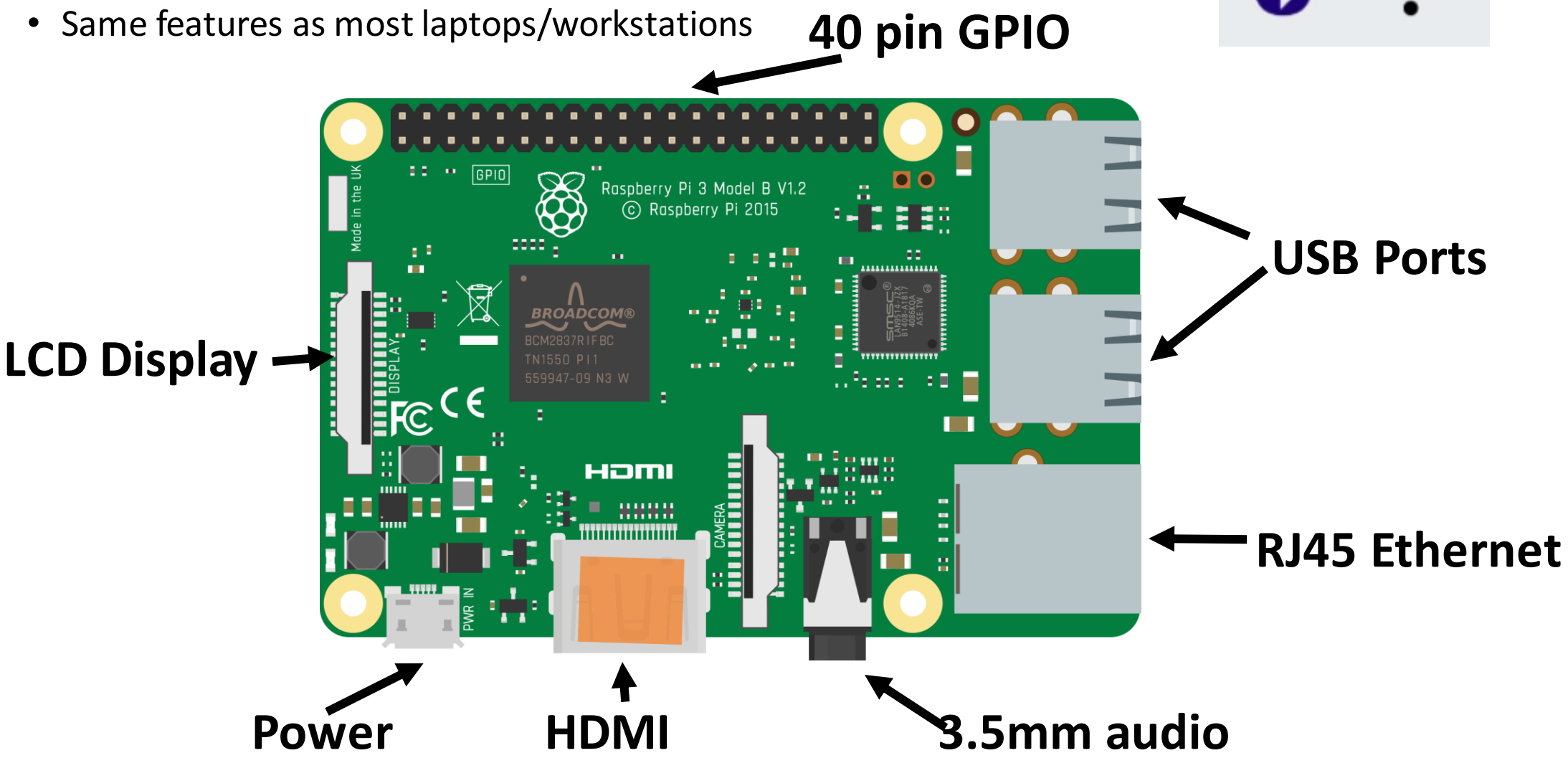
Raspberry Pi/ SenseHAT Introduction

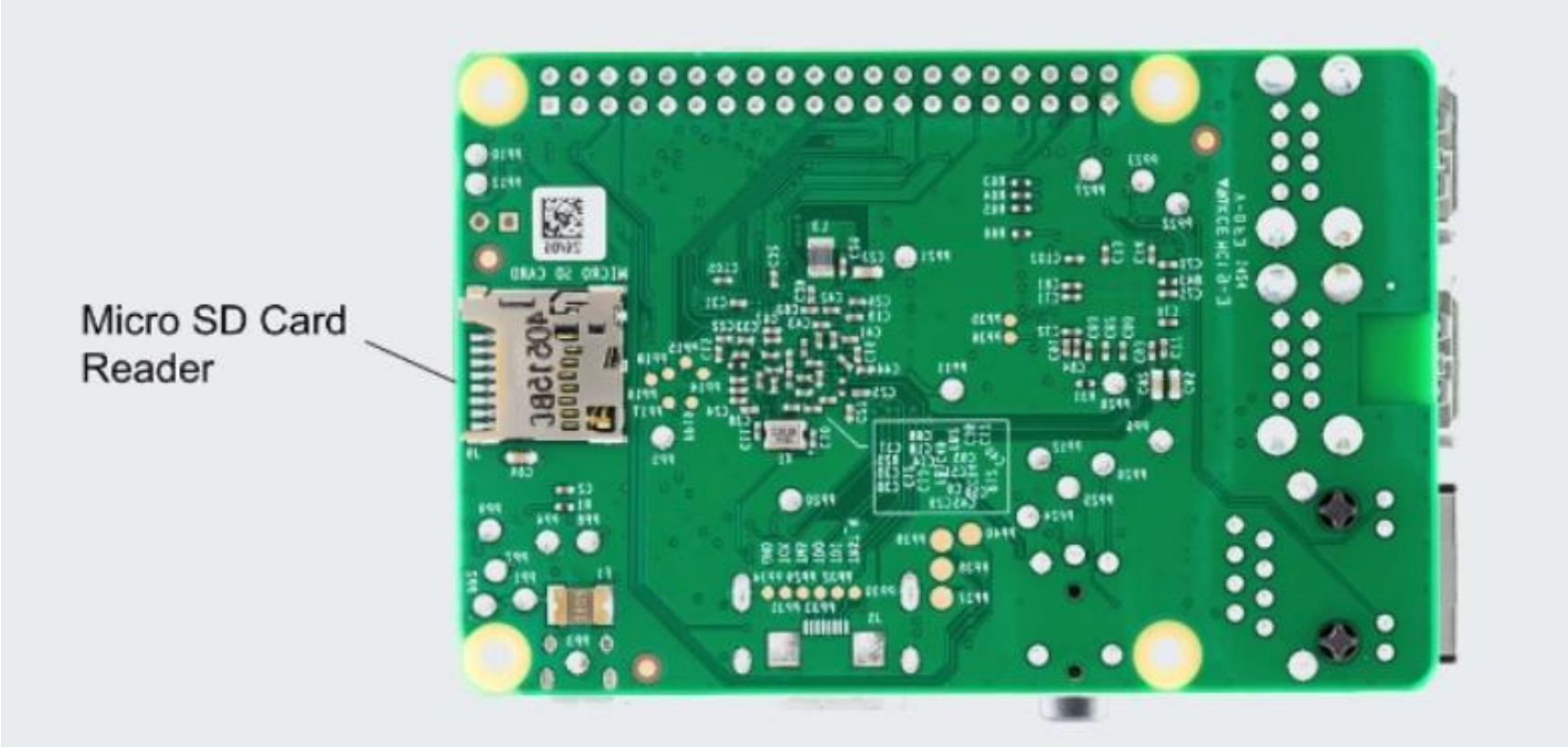
Frank Walsh

What is a Raspberry Pi?

- Low cost, single board computer
- Same features as most laptops/workstations



Raspberry Pi - Storage



History

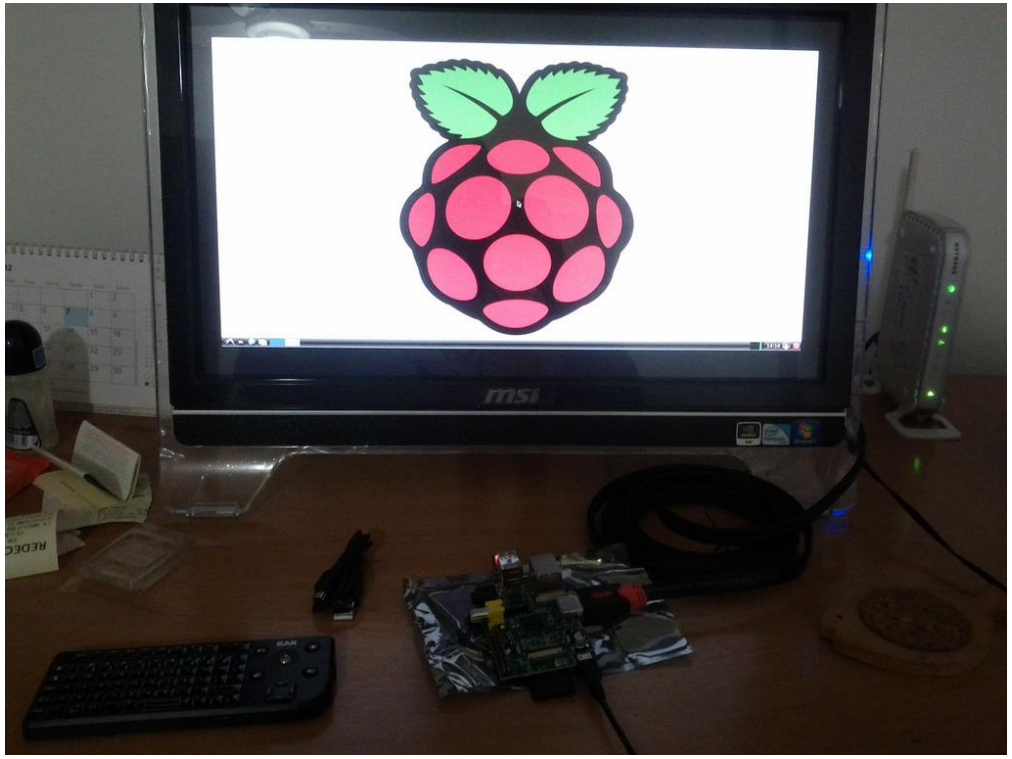
- Launched in 2012
- Inspired by Acorn's BBC Micro from 1981
- Original concept to inspire/educate kids
- Used a lot in 3rd level/home

1981

2016

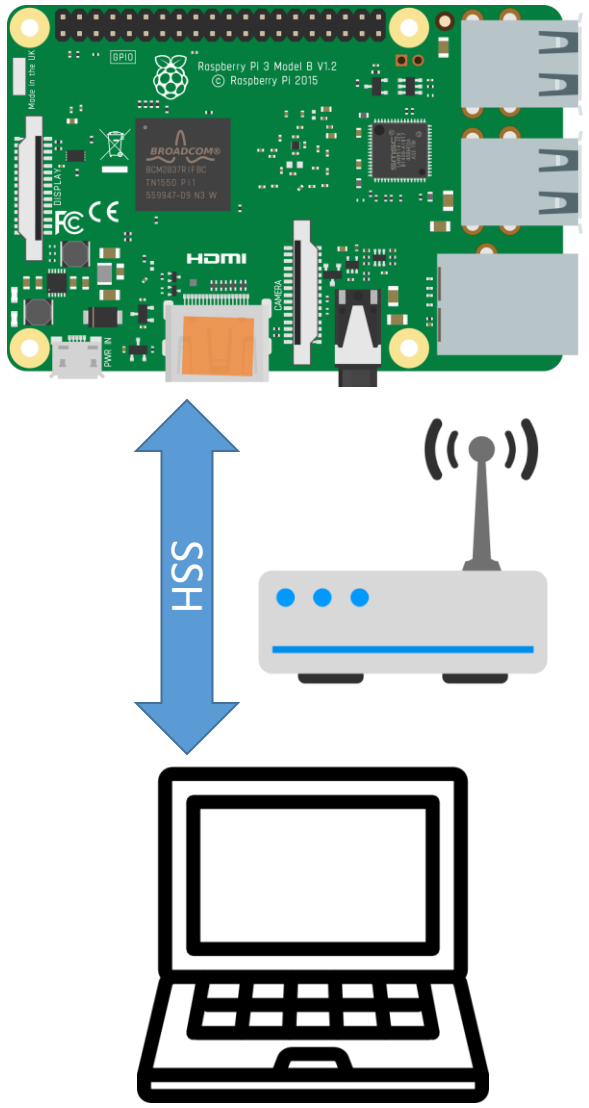
Getting Started -Conventional

- Raspberry Pi 3 B or higher
- Keyboard and mouse
- HDMI display monitor
- SD Card (8 GB+ recommended)
- SD Card Reader (usually a laptop)

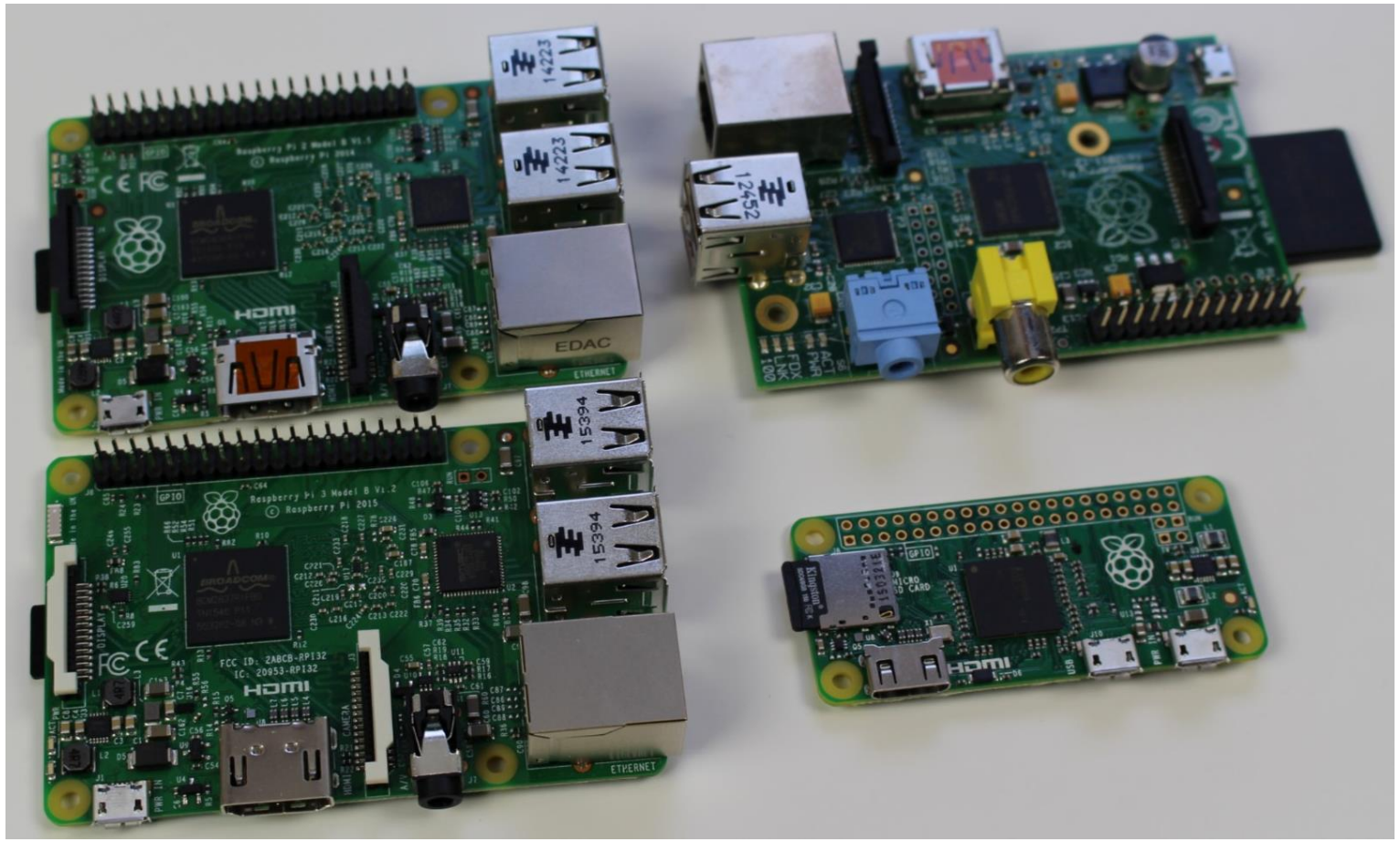


Getting Started –Headless

- Raspberry Pi 3 B or higher
- ~~Keyboard and mouse~~
- ~~HDMI display monitor~~
- SD Card (8 GB+ recommended)
- SD Card Reader (usually a laptop)
- Accessible WiFi network



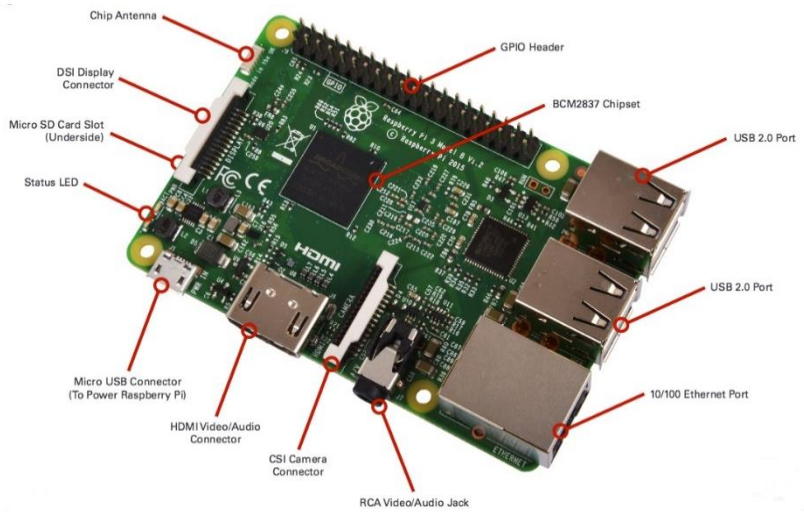
Raspberry Pi Model Comparison



Raspberry Pi Model Comparison

	Raspberry Pi 3 Model B	Raspberry Pi Zero	Raspberry Pi 2 Model B	Raspberry Pi Model B+
Introduction Date	2/29/2016	11/25/2015	2/2/2015	7/14/2014
SoC	BCM2837	BCM2835	BCM2836	BCM2835
CPU	Quad Cortex A53 @ 1.2GHz	ARM11 @ 1GHz	Quad Cortex A7 @ 900MHz	ARM11 @ 700MHz
Instruction set	ARMv8-A	ARMv6	ARMv7-A	ARMv6
GPU	400MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV
RAM	1GB SDRAM	512 MB SDRAM	1GB SDRAM	512MB SDRAM
Storage	micro-SD	micro-SD	micro-SD	micro-SD
Ethernet	10/100	none	10/100	10/100
Wireless	802.11n / Bluetooth 4.0	none	none	none
Video Output	HDMI / Composite	HDMI / Composite	HDMI / Composite	HDMI / Composite
Audio Output	HDMI / Headphone	HDMI	HDMI / Headphone	HDMI / Headphone
GPIO	40	40	40	40
Price	$35	$5	$35	$35

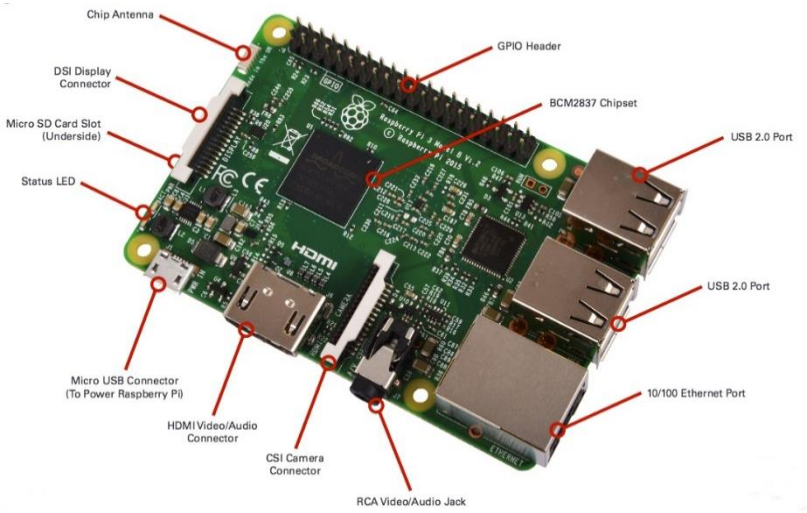
Raspberry Pi - Hardware



- Processor and Memory (for pi 3 B+)
  - Broadcom 2836 System On Chip
    - 64 bit ARM RISC CPU core (not x86 compatible)
    - Videocore IV GPU
  - Default clock is 1200Mhz. Overclocking not permitted in Rpi-3
  - GPU is basically as powerful as what was on the original Xbox
- Mass Storage: micro SD card
  - Any SD card
  - Kernel boots from SD card
  - Easy to access root FS on other device

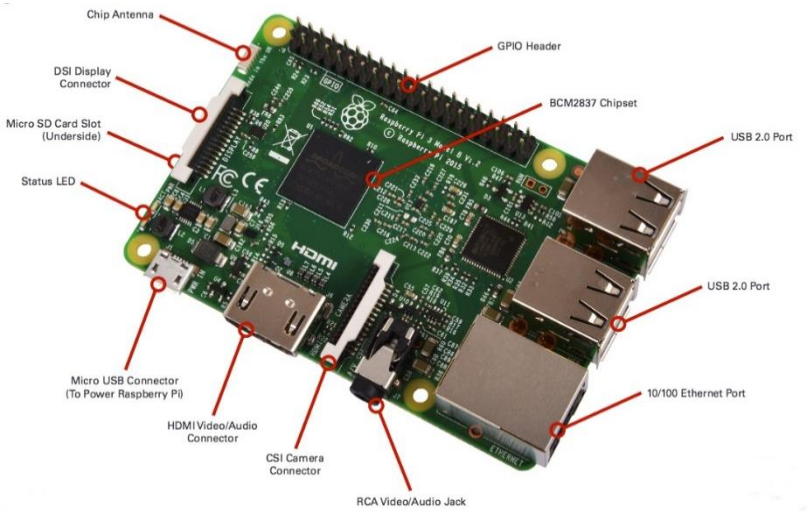
Raspberry Pi - Hardware

- Video
  - HDMI or (digital) DVI via cheap adaptor/cable
  - Wide range of resolutions
    - 640×350 to 1920×1200
- Audio
  - Via HDMI
  - Via 3.5mm stereo jack
  - Stereo analog
  - Output only



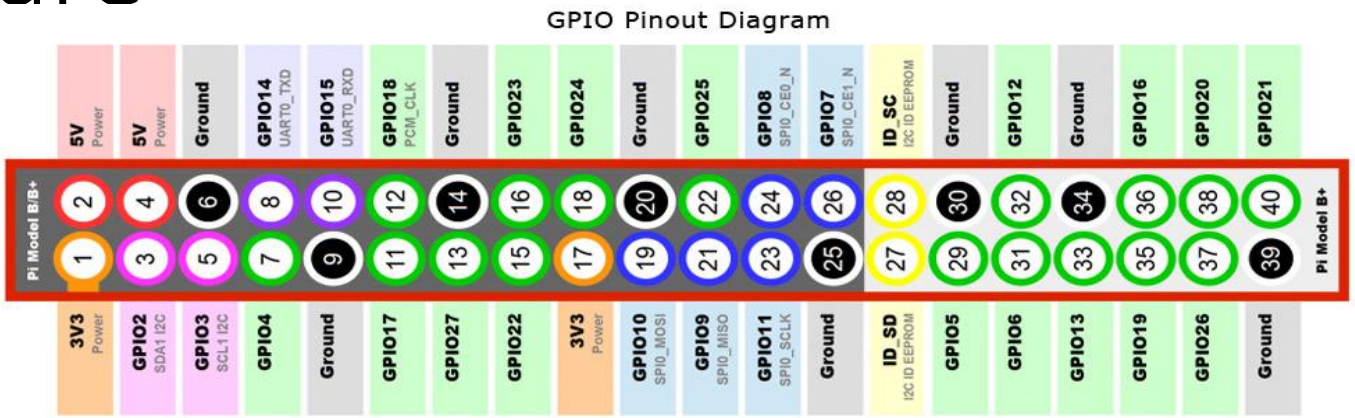
Raspberry Pi - Hardware

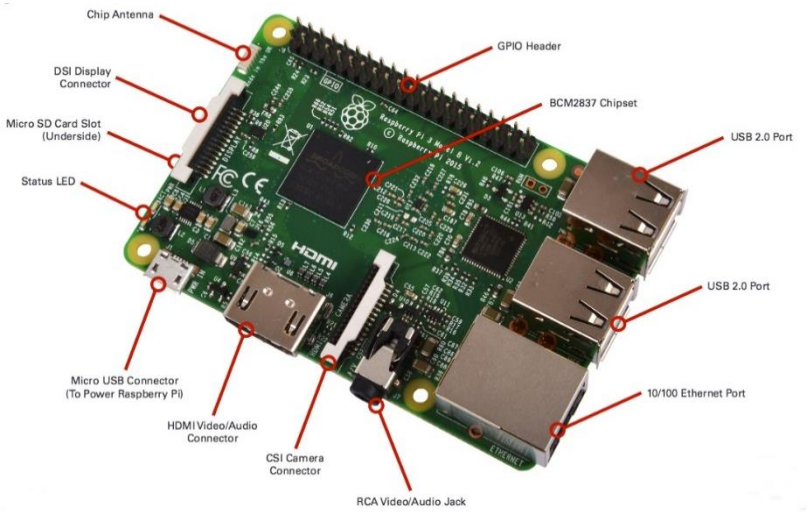
- Networking
  - 10/100Mbps via RJ45 on model B
  - Wireless – Wifi
  - Bluetooth LE 4.2
- USB
  - Four USB sockets on RPi model B, single on model A
  - Expandable via regular or powered hubs
- Power
  - Primary power via microUSB plug
    - A 2.5A cell charger works well, but a 1A cell charger might be
  - Pi-1 requires only 0.8A current
  - PC USB port may not work – Why?



Raspberry Pi - Hardware

- General Purpose I/O (GPIO)
  - 3.3 volt logic via 40 pin header
    - NOT 5 volt or short tolerant
  - Parallel I/O pins
  - UART (Linux console support)
  - I2C, SPI for peripherals
  - No Analog to Digital/D, so no analog input
  - Can do software-based PWM
  - Libraries exist for interacting with the GPIO through several programming languages
- More I/O
  - Display Serial Interface (DSI) LCD panel support
  - Camera Serial Interface (CSI) camera support





Warning - Don't fry your RPi

- RPis are fairly robust but...
- If using with keyboard/screen
  - Connect the USB keyboard and USB first
  - Connect the HMDI connector
  - Turn on the monitor
- Plug in the power cable
  - Do not plug in the SenseHat when the RPi is plugged in and booted into Raspian.
  - ALWAYS plug in the HAT before plugging in the power cable.

Warning - Don't fry your RPi

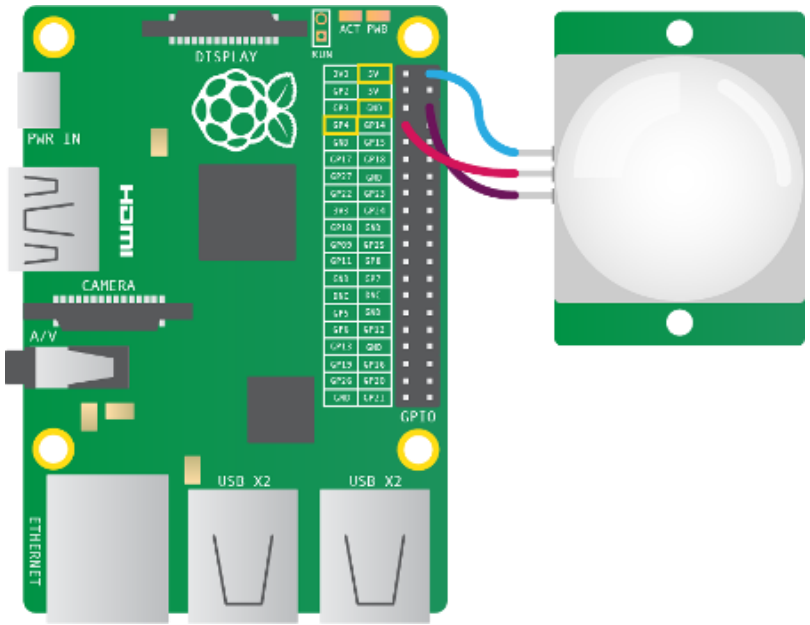
- If you're doing some "physical computing"
  - Input no more that 3.3V to a GPIO
  - Use GPIO.cleanup() to set the GPIO pins back to a safe state; input only
- Current load should be less than 16mA on any one GPIO pin
- Current load should be less than 100mA for all GPIO pins
- Do not connect a motor directly to the Pi's GPIO, the back EMF(Electromotive Force) will fry the Pi;
  - Use a transistor

SenseHAT

- HAT stands for Hardware Attached on Top.
- Several HATs available for RPi.
  - These are external boards you can purchase that plug right into the GPIOs.
- We're using SenseHAT
  - 8×8 RGB LED matrix
  - five-button joystick and includes the following sensors:
  - Gyroscope
  - Accelerometer
  - Magnetometer
  - Temperature
  - Barometric pressure
  - Humidity
- Also a Python library provides an easy programmatic interface to everything on the board.

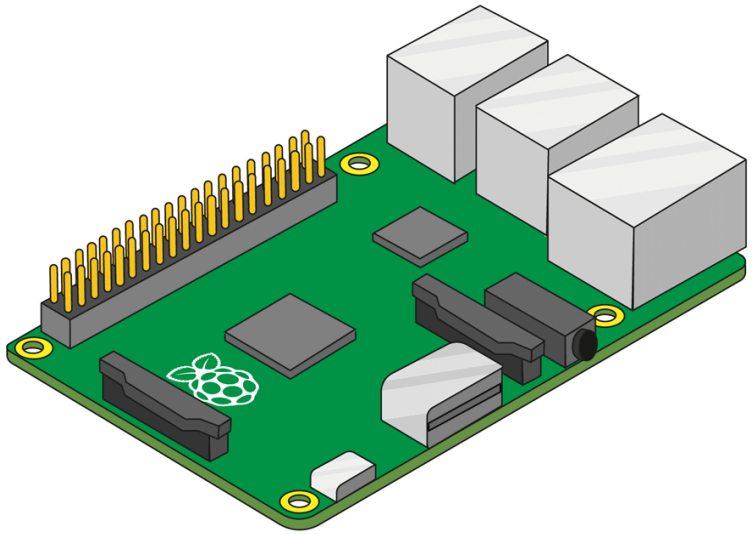
Other things you can do with RPi – Physical Computing

- Can interface with many different devices/components
- Use the GPIO pins on your Raspberry Pi to interface with electronic components, such as LEDs and PIRs.
- Write programs that interact/control attached components



Setting Up Your Pi - Attaching the SenseHAT

- Follow the instructions here: https://projects.raspberrypi.org/en/projects/getting-started-with-the-sense-hat/3



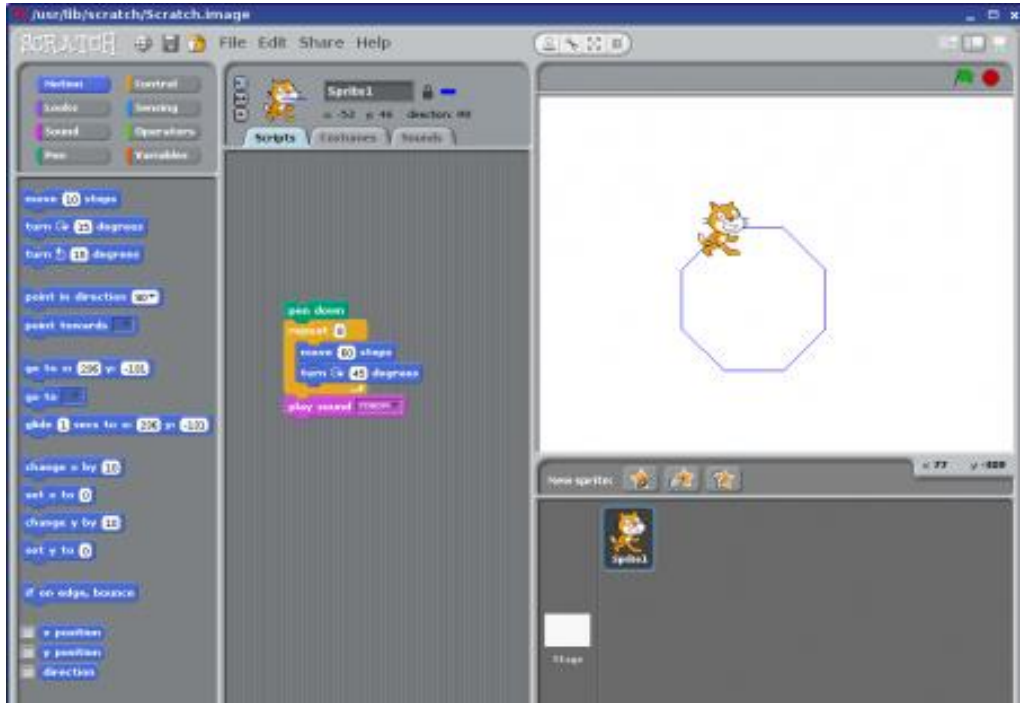
Setting up your RPi – Installing an OS

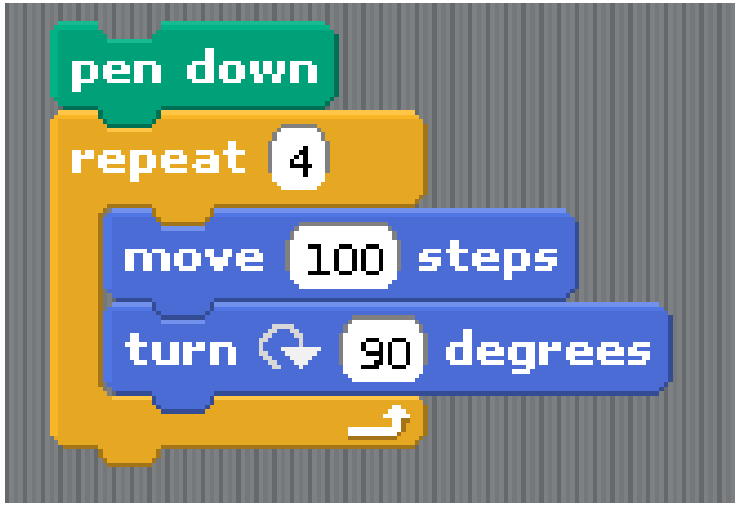
- If you bought your RPi with a pre-installed SD card you're good to go.
- If you have a blank SD card, do this weeks lab!
- Supported OS
  - Raspbian (Debian optimized for Rpi)
  - Debian (non-optimized)
  - Arch Linux
  - RISC OS
  - NetBSD
  - Openelec – an XBMC media center
  - Pidora – Fedora remix

Programming the RPi

- By default, supporting Python as the educational language
- Any language which will compile for ARMv6 can be used with the Raspberry Pi
- For primary age SCRATCH game maker is bundled
- Raspbian also contains Java SE platform

Programming – SCRATCH





scratch is free from MIT

Programming – Python

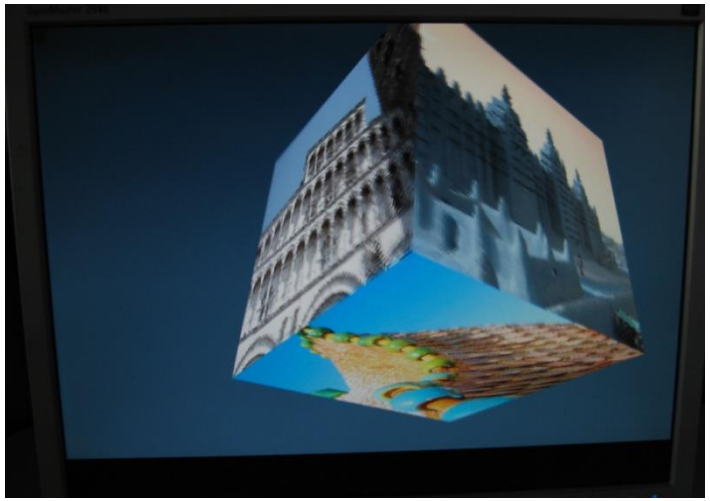
polygon.py

```
1  # polygon.py
2  # draws polygons
3
4  import turtle
5  def polygon(length, sides):
6      for i in range (sides):
7          turtle.fd(length)
8          turtle.left(360/sides)
9  # main
10 print("Let's draw a polygon.")
11 how_many = int(input("How many sides would you like?"))
12 how_big = int(input("How long do you want the sides?"))
13 polygon(how_big, how_many)
14 input("Press a key to quit.")
```

Programming - OpenGL

- Raspberry Pi incorporates a powerful graphics accelerator – OpenGL
  - Examples, including Quake 3 at
    - https://github.com/raspberrypi/quake3

```
   I A  triangle.c (Modified)(c)  static void redraw_sc  Row 359  Col 48   6:10  Ctrl-K H for help
static void redraw_scene(CUBE_STATE_T *state)
{
   // Start with a clear screen
   glClear( GL_COLOR_BUFFER_BIT );
   glMatrixMode(GL_MODELVIEW);

   glEnable(GL_TEXTURE_2D);
   glTexEnvx(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE);

   glBindTexture(GL_TEXTURE_2D, state->tex[0]); // bind texture
   glRotatef(270.f, 0.f, 0.f, 1.f ); // front face normal along z axis
   glDrawArrays( GL_TRIANGLE_STRIP, 0, 4);

   // same pattern for other 5 faces - rotation chosen to make image orientation 'nice'
   glBindTexture(GL_TEXTURE_2D, state->tex[1]);
   glRotatef(90.f, 0.f, 0.f, 1.f ); // back face normal along z axis
   glDrawArrays( GL_TRIANGLE_STRIP, 4, 4);

   glBindTexture(GL_TEXTURE_2D, state->tex[2]);
   glRotatef(90.f, 1.f, 0.f, 0.f ); // left face normal along x axis
   glDrawArrays( GL_TRIANGLE_STRIP, 8, 4);

   glBindTexture(GL_TEXTURE_2D, state->tex[3]);
   glRotatef(90.f, 1.f, 0.f, 0.f ); // right face normal along x axis
   glDrawArrays( GL_TRIANGLE_STRIP, 12, 4);

   glBindTexture(GL_TEXTURE_2D, state->tex[4]);
   glRotatef(270.f, 0.f, 1.f, 0.f ); // top face normal along y axis
   glDrawArrays( GL_TRIANGLE_STRIP, 16, 4);

   glTexEnvx(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);

   glBindTexture(GL_TEXTURE_2D, state->tex[5]);
   glRotatef(90.f, 0.f, 1.f, 0.f ); // bottom face normal along y axis
   glDrawArrays( GL_TRIANGLE_STRIP, 20, 4);

   glDisable(GL_TEXTURE_2D);

   eglSwapBuffers(state->display, state->surface);
}
_
```

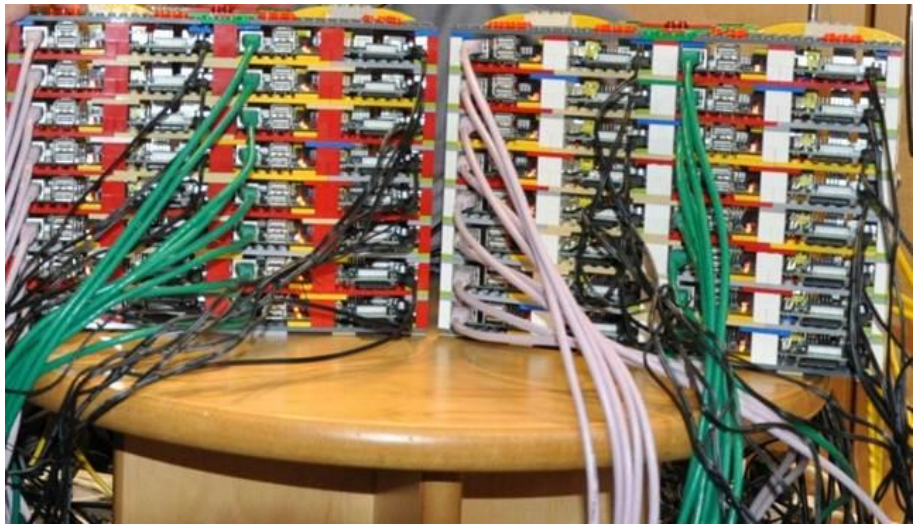


Some Other Things

- Can run Kodi software
  - Turn your TV into a Smart TV!
- Emulators for tons of old platforms (C64, Atari, NES, etc.)
  - I have a mame emulator.
- Brew your own beer!
  - I've been doing this with the help of an RPi since 2014
- Has its own app store
  - http://store.raspberrypi.com/

Projects Using Raspberry Pi

- Raspberry Pi Cluster
  - https://arstechnica.com/information-technology/2012/09/university-builds-cheap-supercomputer-with-raspberry-pi-and-legos/
  - http://www.youtube.com/watch?v=Jq5nrHz9I94



Python Primer

With help from

https://lobster1234.github.io/2017/05/25/python-java-primer/

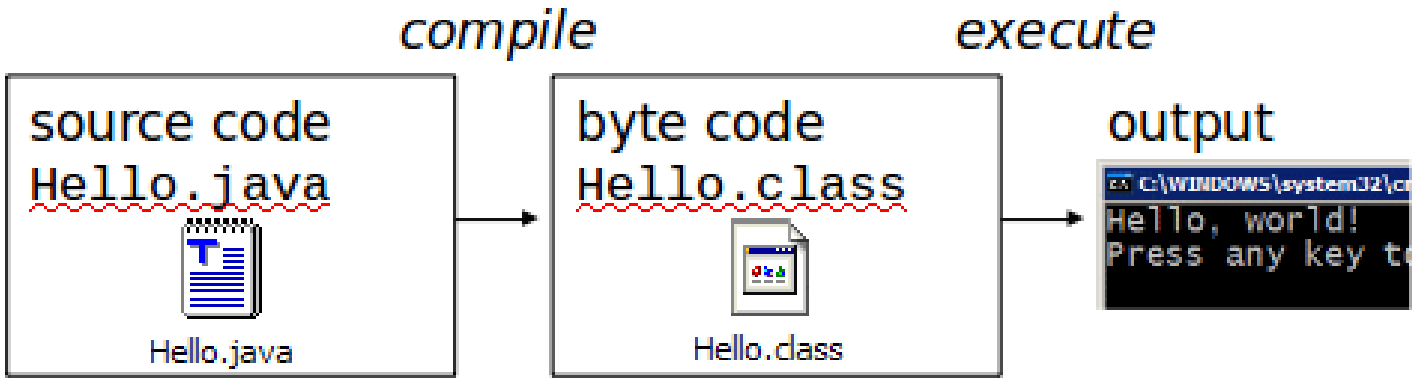
What's Python

- Created 1991
- Used for Web Dev, Software Dev, Maths, Embedded Devices (micro python), Scripting
- Why Python
  - Works on many platforms
  - Easy to learn(simple syntax)
- One thing to always remember about Python:

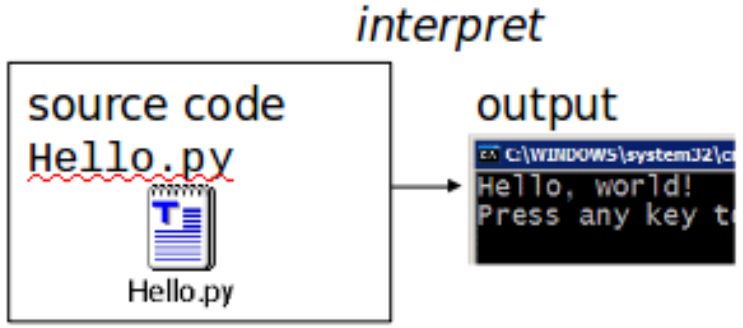
  Relies on indentation(using white space) to define scope for if statements, loops, functions. Java uses curly-brackets for this.

Python is interpreted

- Many languages require compilation of program to a form the machine understands



- Python is interpreted directlv to machine instructions



Variables

```
1  name = "Name"
2  age = 42
3  completed = False
4  c = 'A'
5  f = 0.0034
6
7  print(name + str(age))
```

Read Input

```
name = input("Enter name\n")
age = int(input("Enter age\n"))
print("Name is ", name ," and age is ", age)
```

Operators

```
i = 5;
j = 3;
print("Sum : ", i+j)
print("Diff : " , i-j)
print("Product : " , i*j)
print("Modulo : " , i%j)
print("Floor Division : " , i//j)
print("Float Division : " , i/j)
```

```
Sum :  8
Diff :  2
Product :  15
Modulo :  2
Floor Division :  1
Float Division :  1.6666666666666667
```

If-then-else

```
n=4

if n%2 == 1  :
    print("Weird")
elif n%2==0 and n >=2 and n <=5 :
    print("Not Weird")
elif n%2==0 and n >=6 and n <=20 :
    print("Weird")
elif n%2==0 and n > 20 :
    print("Not Weird")
```

For loop

```
for n in range(5) :
    print(n)

for c in "Computer-Systems" :
  print(c)
```

```
0
1
2
3
4
C
o
m
p
u
t
e
r
```

While loop

```
n = 0
while True :
    n = n + 1
    print(n)
    if n==20 :
        break
print("Loop broke with n = ",n)
```

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Loop broke with n =  20
```

Functions

```
def add(x, y):
    return x+y # If return is skipped, the method returns a None

print(add(1,2))

# Can be called with named arguments

print(add(x=5,y=7)) # Will print 12

# Can take default arguments too
def add(x, y=2):
  return x+y

print(add(5)) # Will return 7
```

```
3
12
7
```

Lists

```
list = [] # create a new list
list.append("A") # add to the end
list.append("B")
list.append("C")
list.insert(3,"D") # insert at an index
sorted_list = sorted(list) # returns a new, sorted list
list.sort() # in place sort
list.remove("C") # remove first occurrence of "C"
popped = list.pop(2) # return and remove the item at index 2
list.reverse() # reverse the list
a = list.index("A") # return index of "A"
list += ["E","F"] # add a list with E and F to the end of this list
size = len(list) # size of the list
print(size)
print(list)
```

```
4
['B', 'A', 'E', 'F']
```