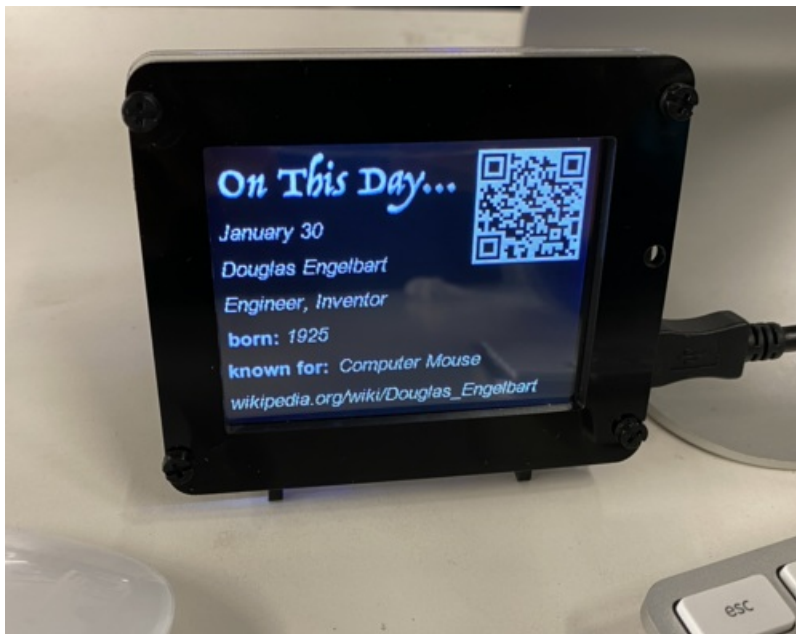


## Electronic History of the Day with PyPortal

Created by Isaac Wellish



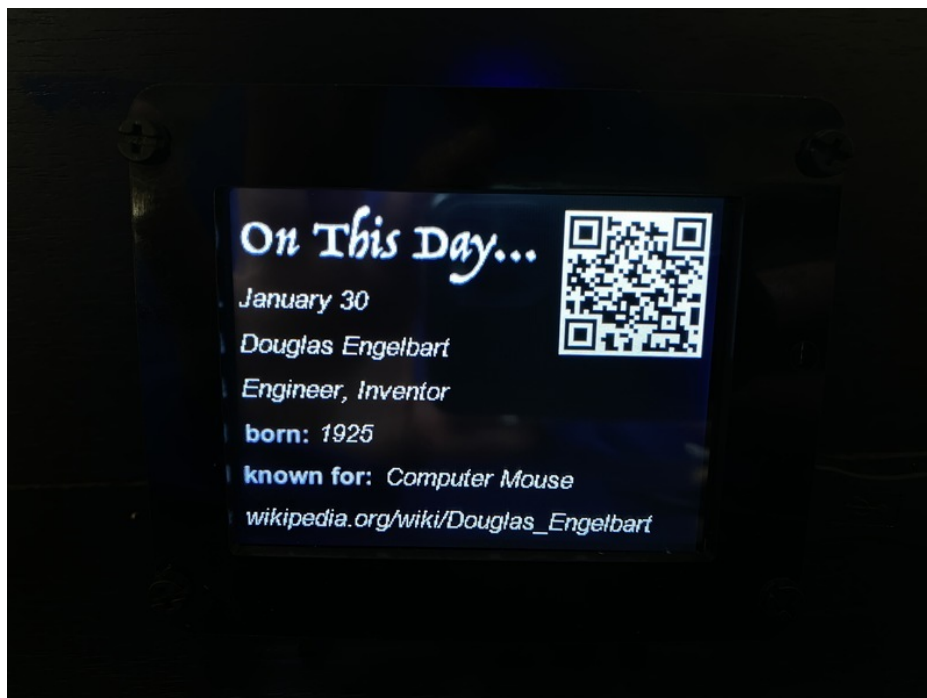
Last updated on 2020-10-28 03:51:03 PM EDT

# Overview



This PyPortal project shows one notable electronic or science fact every day. Based on what day it is, CircuitPython code grabs JSON data from a database and then prints information about the person associated with that day. Want more info on the person? Use the QR code!

We created our very own database with data on the most memorable electronics and science peoples (<https://adafru.it/IE2>). The CircuitPython code grabs from this database to bring you all these great factoids!



## Parts

You can pick up an Adafruit PyPortal and a USB cable (if needed). If you like, you can mount the PyPortal in the Adafruit laser-cut acrylic stand.



Adafruit PyPortal - CircuitPython Powered Internet Display

\$54.95  
IN STOCK

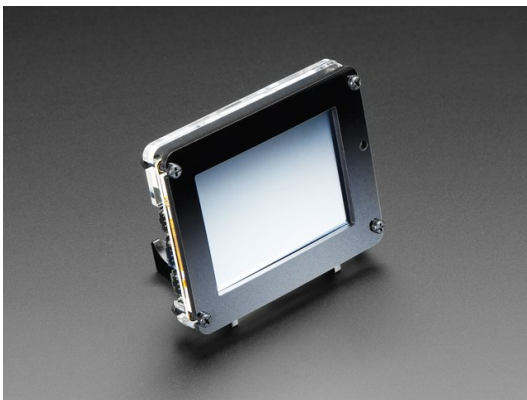
Add To Cart



USB cable - USB A to Micro-B

\$2.95  
IN STOCK

Add To Cart



Adafruit PyPortal Desktop Stand Enclosure Kit

\$9.95  
IN STOCK

Add To Cart

You can use any case you like. If you get the stand enclosure kit, the instructions for assembly are in [this guide page \(https://adafru.it/IKC\)](https://adafru.it/IKC). You can see some other cases (3D printed, etc. via [this Adafruit Learning System link \(https://adafru.it/IKD\)](https://adafru.it/IKD)).

# Install CircuitPython

CircuitPython (<https://adafru.it/tB7>) is a derivative of MicroPython (<https://adafru.it/BeZ>) designed to simplify experimentation and education on low-cost microcontrollers. It makes it easier than ever to get prototyping by requiring no upfront desktop software downloads. Simply copy and edit files on the **CIRCUITPY** "flash" drive to iterate.

The following instructions will show you how to install CircuitPython. If you've already installed CircuitPython but are looking to update it or reinstall it, the same steps work for that as well!

## Set up CircuitPython Quick Start!

Follow this quick step-by-step for super-fast Python power :)

<https://adafru.it/Egk>

<https://adafru.it/Egk>

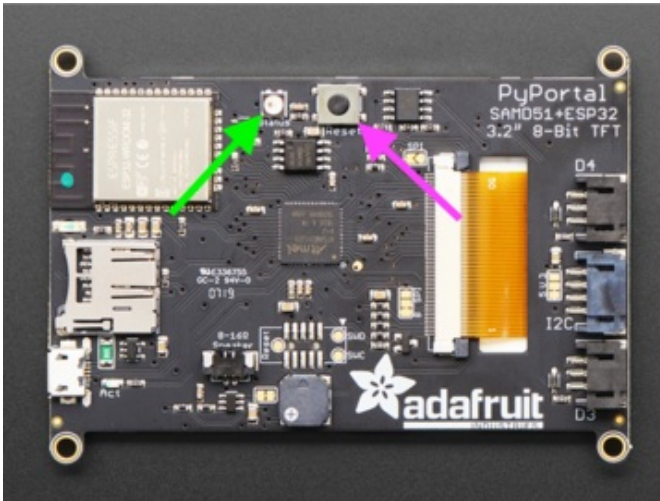
<https://adafru.it/HFd>

<https://adafru.it/HFd>



**Click the link above to download the latest version of CircuitPython for the PyPortal.**

Download and save it to your desktop (or wherever is handy).

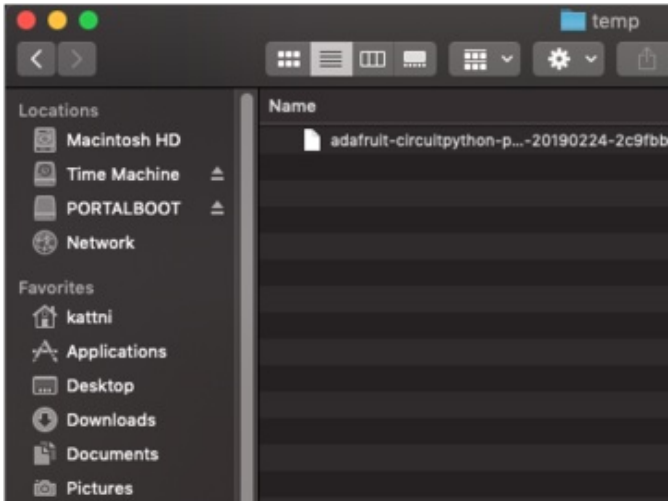


Plug your PyPortal into your computer using a known-good USB cable.

**A lot of people end up using charge-only USB cables and it is very frustrating! So make sure you have a USB cable you know is good for data sync.**

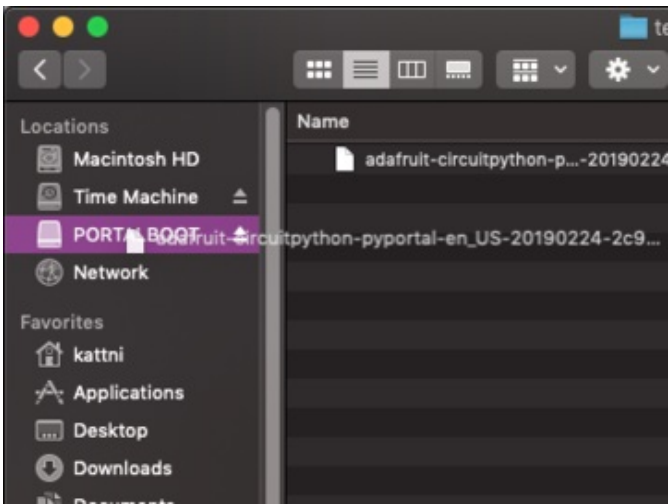
Double-click the **Reset** button on the top in the middle (magenta arrow) on your board, and you will see the NeoPixel RGB LED (green arrow) turn green. If it turns red, check the USB cable, try another USB port, etc. **Note:** The little red LED next to the USB connector will pulse red. That's ok!

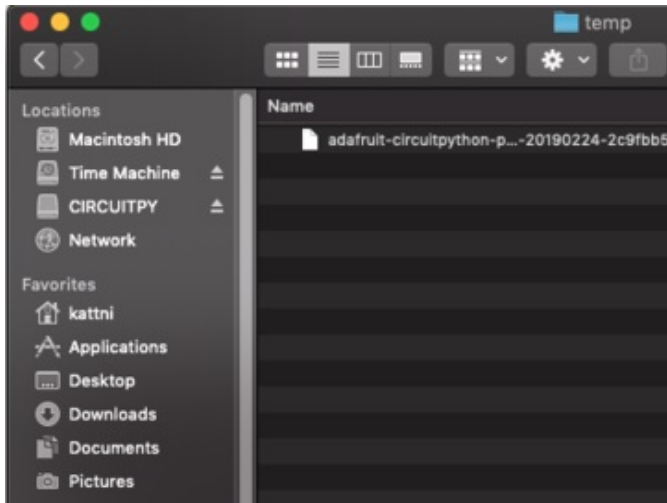
If double-clicking doesn't work the first time, try again. Sometimes it can take a few tries to get the rhythm right!



You will see a new disk drive appear called **PORTALBOOT**.

Drag the **adafruit-circuitpython-pyportal-<whatever>.uf2** file to **PORTALBOOT**.





The LED will flash. Then, the **PORTALBOOT** drive will disappear and a new disk drive called **CIRCUITPY** will appear.

If you haven't added any code to your board, the only file that will be present is **boot\_out.txt**. This is absolutely normal! It's time for you to add your **code.py** and get started!

That's it, you're done! :)

## PyPortal Default Files

Click below to download a zip of the files that shipped on the PyPortal or PyPortal Pynt.

<https://adafru.it/Env>

<https://adafru.it/Env>

<https://adafru.it/HFf>

<https://adafru.it/HFf>

# PyPortal CircuitPython Setup

To use all the amazing features of your PyPortal with CircuitPython, you must first install a number of libraries. This page covers that process.

## Adafruit CircuitPython Bundle

---

Download the Adafruit CircuitPython Library Bundle. You can find the latest release here:

<https://adafru.it/ENC>

<https://adafru.it/ENC>

Download the **adafruit-circuitpython-bundle-\*.x-mpy-\*.zip** bundle zip file where **\*.x MATCHES THE VERSION OF CIRCUITPYTHON YOU INSTALLED**, and unzip a folder of the same name. Inside you'll find a **lib** folder. You have two options:

- You can add the **lib** folder to your **CIRCUITPY** drive. This will ensure you have *all the drivers*. But it will take a bunch of space on the 8 MB disk
- Add each library as you need it, this will reduce the space usage but you'll need to put in a little more effort.

At a minimum we recommend the following libraries, in fact we more than recommend. They're basically required. So grab them and install them into **CIRCUITPY/lib** now!

- **adafruit\_esp32spi** - This is the library that gives you internet access via the ESP32 using (you guessed it!) SPI transport. You need this for anything Internet
- **adafruit\_requests** - This library allows us to perform HTTP requests and get responses back from servers. GET/POST/PUT/PATCH - they're all in here!
- **adafruit\_pyportal** - This is our friendly wrapper library that does a lot of our projects, displays graphics and text, fetches data from the internet. Nearly all of our projects depend on it!
- **adafruit\_touchscreen** - a library for reading touches from the resistive touchscreen. Handles all the analog noodling, rotation and calibration for you.
- **adafruit\_io** - this library helps connect the PyPortal to our free datalogging and viewing service
- **adafruit\_imageload** - an image display helper, required for any graphics!
- **adafruit\_display\_text** - not surprisingly, it displays text on the screen
- **adafruit\_bitmap\_font** - we have fancy font support, and its easy to make new fonts. This library reads and parses font files.
- **adafruit\_slideshow** - for making image slideshows - handy for quick display of graphics and sound
- **neopixel** - for controlling the onboard neopixel
- **adafruit\_adt7410** - library to read the temperature from the on-board Analog Devices ADT7410 precision temperature sensor
- **adafruit\_sdcard** - support for reading/writing data from the onboard SD card slot.
- **adafruit\_bus\_device** - low level support for I2C/SPI



# Download Project Files from GitHub

All of the files required for this project (the code, graphics, and font) are available for download on GitHub. There are **two code files** for this project, our regular **code.py** along with **secrets.py**

You'll need to edit **secrets.py** with your applicable information. You can find more info on all of the code files in the next two pages of this guide.

## CircuitPython Code

In the embedded code element below, click on the **Download: Project Zip** link, and save the .zip archive file to your computer.

Then, uncompress the .zip file, it will unpack to a folder named **PyPortal\_on\_this\_day**.

Copy the contents of the **PyPortal\_on\_this\_day** directory to your PyPortal's **CIRCUITPY** drive.



```

"""
'of this day' demo
Display notable info about famous electronics-related peoples
Data sources: https://github.com/adafruit/OTD/tree/master/electronics
"""

import time
import board
from adafruit_pyportal import PyPortal

cwd = ("%"+__file__).rsplit('/', 1)[0] # the current working directory (where this file is)

DAY = ["Day of the year"]
PERSON = ["Person"]
NOTABLE = ["Notable for"]
YEAR = ["Year"]
ACCOMPLISH = ["Accomplishment"]
WEB = ["Web Reference"]

BASE_DATA = "https://raw.githubusercontent.com/adafruit/OTD/master/electronics/"

# a function that returns whatever is passed in
def identity(x):
    return x

```

```

# create pyportal object w no data source (we'll feed it text later)
pyportal = PyPortal(url = BASE_DATA, debug=True,
                    json_path = (DAY, PERSON, NOTABLE, YEAR, ACCOMPLISH, WEB),
                    status_neopixel = board.NEOPIXEL,
                    default_bg = cwd + "/on_this_day_bg.bmp",
                    text_font = cwd+"fonts/Arial-ItalicMT-17.bdf",
                    text_transform = [identity]*6, # we do this so the date doesnt get commas
                    text_position=((10, 70), (10, 100), (10, 130),(60, 160), (105, 190), (10, 220)),
                    text_color=(0xFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFF),
                    text_maxlen=(50, 50, 50, 50, 50, 50), # cut off characters
                    )

while True:
    try:
        print("Getting time from internet!")
        pyportal.get_local_time()
        refresh_time = time.monotonic()
    except RuntimeError as e:
        print("Some error occured, retrying! -", e)
        continue

    now = time.localtime()
    print("Current time:", now)
    url = BASE_DATA+"%02d_%02d.json" % (now.tm_mon, now.tm_mday)
    print(url)
    response = None
    try:
        response = pyportal.fetch(url)
        print("Response is", response)
    except RuntimeError as e:
        print("Some error occured, retrying! -", e)

    # Make a QR code from web reference
    pyportal.show_QR(bytearray(response[5]), qr_size=3,
                    x=220, y=10)

    # wait 10 minutes before running again
    time.sleep(10*60)

```

## Downloading the libraries

Per the CircuitPython Libraries page previously, download the library bundle appropriate for your version of CircuitPython. Copy the below library files/folders to a directory named **lib** on your **CIRCUITPY** drive.



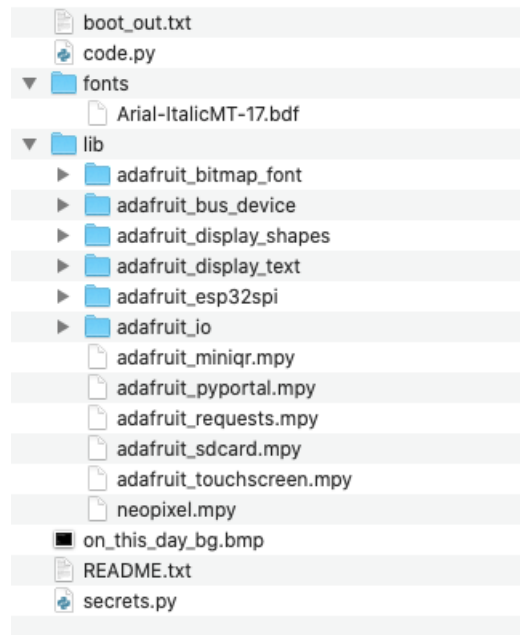
Make sure to add the necessary libraries to the lib folder, info on how to do this can be found in the "PyPortal CircuitPython Setup" section

This project uses the following CircuitPython libraries:

- **adafruit\_bitmap\_font** (directory)
- **adafruit\_bus\_device** (directory)
- **adafruit\_display\_shapes** (directory)
- **adafruit\_display\_text** (directory)
- **adafruit\_esp32spi** (directory)

- `adafruit_io` (directory)
- `adafruit_miniqr.mpy` (file)
- `adafruit_pyportal.mpy` (file)
- `adafruit_requests` (file)
- `adafruit_sdcard.mpy` (file)
- `adafruit_touchscreen.mpy` (file)
- `neopixel.mpy` (file)

This is what the final contents of the **CIRCUITPY** drive will look like:



## Double check

See the directory listing above and double check that you have all the files listed to make this project function. If any are missing or in an incorrect directory, move them so they're in the right places. The project will not work correctly unless the files are in the right locations.

# Code Walkthrough - secrets.py

secrets.py

If you've ever done a PyPortal project or other IoT project with CircuitPython, then the **secrets.py** file will be familiar to you. It allows you to store your WiFi network's SSID and password, along with other sensitive information, separately from your main program, so that you don't accidentally share your info when sharing your code.

In the case of this project, you will need your network information, as well as an Adafruit IO account. Your Adafruit AIO information and your location will also be placed in the file, all formatted to be able to access the correct day and time. By keeping your location in **secrets.py**, you can easily edit it without diving too deep into the main program or risk breaking anything.

## Adafruit IO set up

Using [Mu \(https://adafru.it/HA8\)](https://adafru.it/HA8) or any text editor, you should add your Adafruit IO Username and Adafruit IO Key to the **secrets.py** file. This project connects to Adafruit IO to get the time and date. Adafruit IO is free to use, but you'll need to log in with your Adafruit account to use it. If you don't already have an Adafruit login, create [one here \(https://adafru.it/dAQ\)](https://adafru.it/dAQ).



Once you have logged into your account, there are two pieces of information you'll need to place in your **secrets.py** file: Adafruit IO username, and Adafruit IO key. Head to [io.adafruit.com \(https://adafruit.com\)](https://adafruit.com) and simply click the **View AIO Key** link on the left hand side of the Adafruit IO page to get this information.

Then, update the **aoi\_username** and **aoi\_key** values in the **secrets.py** file. Your **secrets.py** file should look similar to this:

```
# This file is where you keep secret settings, passwords, and tokens!
# If you put them in the code you risk committing that info or sharing it

secrets = {
    'ssid' : 'your-ssid-here',
    'password' : 'your-password-here',
    'aoi_username' : "your-aio-username-here",
    'aoi_key' : 'your-aio-key-here',
    'location' : 'New York, US'
}
```

# Code Walkthrough - code.py

`code.py` is the file where the main code of the program resides!

## Background

First, the program displays a bitmap graphic as the screen's background.

This background has the title of the program as well as a couple of sub-titles to make the data more readable.

This is a 320 x 240 pixel RGB 16-bit raster graphic in `.bmp` format.

If you would like to create your own background, awesome! You'll want to save the file with these specifications:

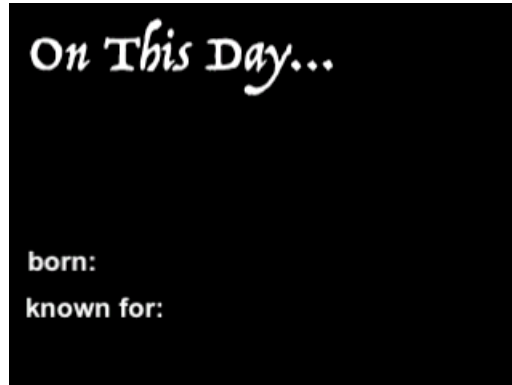
- 320 x 240 pixels
- 16-bit RGB color
- Save file as `.bmp` format

You can then copy the `.bmp` file to the root level of the **CIRCUITPY** drive. Make sure you refer to this new filename in the pyportal constructor line:

```
default_bg=cwd+ "/on_this_day_bg.bmp"
```

Change that line to use the new filename name, such as:

```
default_bg=cwd+"/my_new_background.bmp"
```



## Font

The fonts used here are bitmap fonts made from the Arial Italic typeface. You can learn more about [converting type in this guide \(https://adafru.it/E7E\)](https://adafru.it/E7E).

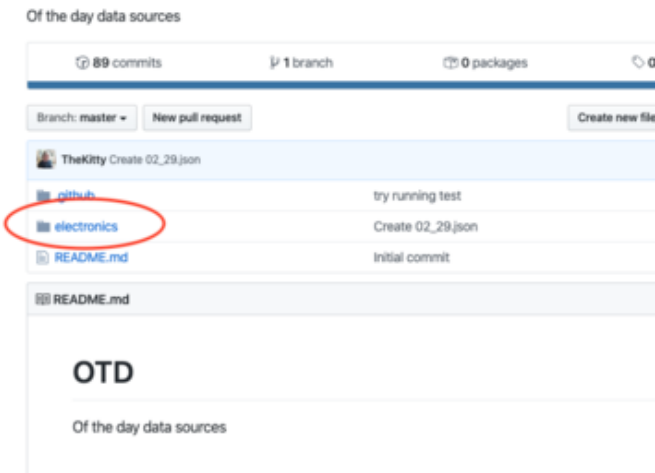
## JSON

The neat part is that the text is not coming from a file on the device, but rather it is taken from a website!

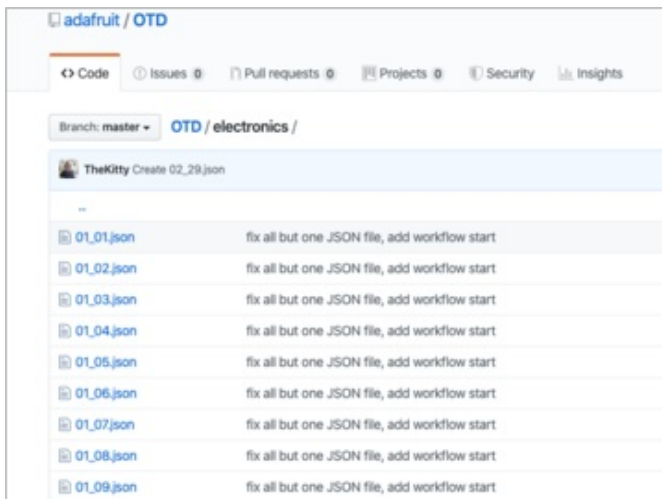
## The OTD Database

[The Adafruit "OTD" repository on Github \(https://adafru.it/IE4\)](https://adafru.it/IE4) is where we're storing all this enlightening data. We can

grab the JSON directly from Github site and display it on the PyPortal.



- Head to the site here (<https://adafru.it/IE4>) to see the lovely home of all this data.
- Click on the "electronics" folder (<https://adafru.it/IE2>).
- Now click any of the dates you want to see that data for, and voila there's the data!
- Now click on the "raw" button.
- This leads to the direct source of the JSON data, where the data is pulled from.



```
{
  "Day of the year": "January 25",
  "Person": "Robert Boyle",
  "Notable for": "Chemist, Physicist",
  "Year": "1627",
  "Accomplishment": "Chemistry, pV=Nrt",
  "Web Reference": "wikipedia.org/wiki/Robert_Boyle"
}
```

How to reference JSON data in code.py

The below code is the JSON data for Jan 25.

```
{
  "Day of the year": "January 25",
  "Person": "Robert Boyle",
  "Notable for": "Chemist, Physicist",
  "Year": "1627",
  "Accomplishment": "Chemistry, pV=Nrt",
  "Web Reference": "wikipedia.org/wiki/Robert_Boyle"
}
```

If we look through the JSON file, we'll see 6 keys called `Day of the year`, `Person`, `Notable for` etc.

Each key has an associated value paired with it.

Our CircuitPython code is able to grab and parse this data using these variables:

```
DAY = ["Day of the year"]
PERSON = ["Person"]
NOTABLE = ["Notable for"]
YEAR = ["Year"]
ACCOMPLISH = ["Accomplishment"]
WEB = ["Web Reference"]
```

Next, here's our data source from above!

```
BASE_DATA = "https://raw.githubusercontent.com/adafruit/OTD/master/electronics/"
```

## PyPortal Constructor

Then, in the `pyportal` query we ask for the `Day of the year`, `Person`, `Notable for` etc. from that URL, and then use the `text_` arguments to set the `font`, `transform`, `position`, `color`, and `maxlen` of the text when it is displayed.

```
pyportal = PyPortal(url = BASE_DATA, debug=True,
                    json_path = (DAY, PERSON, NOTABLE, YEAR, ACCOMPLISH, WEB),
                    status_neopixel = board.NEOPIXEL,
                    default_bg = cwd + "/on_this_day_bg.bmp",
                    text_font = cwd+"fonts/Arial-ItalicMT-17.bdf",
                    text_transform = [identity]*6, # we do this so the date doesnt get commas
                    text_position=((10, 70), (10, 100), (10, 130),(60, 160), (105, 190), (10, 220)),
                    text_color=(0xFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFF),
                    text_maxlen=(50, 50, 50, 50, 50, 50), # cut off characters
                    )
```

With all of this prepared, during the main loop of `while True:` the code will:

- Find out what day it is and insert that date at the end of the data URL.
- With that new URL, the program will query the Adafruit OTD Github page for the JSON data, and display it, along with the QR code for the associated Wikipedia link.
- Then the program will wait 10 minutes until repeating the process.

And that's the whole program!



