

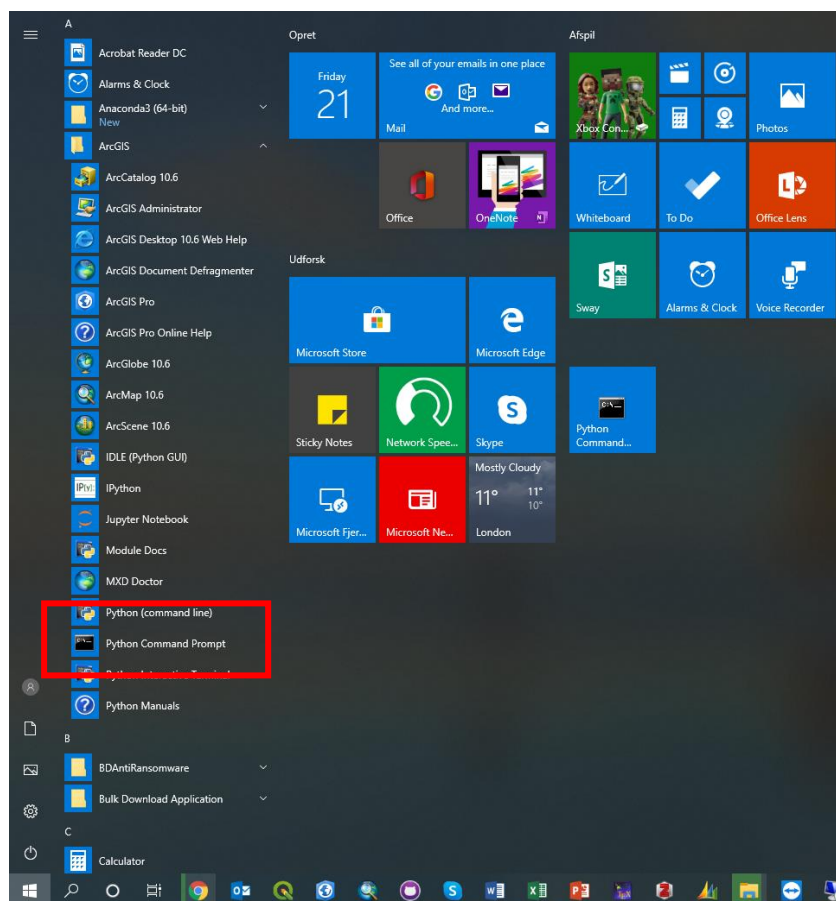
# Setting up Python for batch downloading Sentinel-2 scenes

By Penny How – [how@asiaq.gl](mailto:how@asiaq.gl)

This set of instructions is for downloading Sentinel-2 scenes from the SciHub and Google Cloud repositories.

## 1. Setting up Python to facilitate downloading

We first need to create Conda environment to hold the packages we need to batch download Landsat scenes. To do this, find Python Command Prompt which will be on the Start menu either under Anaconda3:



This will open a black box console where you can type text. The text printed before the cursor signifies the Conda environment that is active (in the brackets) followed by the active file directory. We firstly want to create a new Conda environment, by copying the base environment holding all the key Python packages. To do this, enter the following command:

```
conda create --clone base --name download
```

By entering this command we are stating to create a new Conda environment, which is a copy of the base environment, and we are going to call it 'download'. Once entered, you will see many statements showing the environment being constructed. Once finished, you will see your cursor prompt return. To activate this environment, type:

```
activate download
```

You will now see the activated Conda environment change from the base environment to your newly made environment.

Now we have to install the package needed for our batch downloading to work. This package is called SentinelSat, which was developed to search, download and retrieve data from the Copernicus Open Access Hub. SentinelSat is available to download through pip, an easy installation method which requires just one command:

```
pip install sentinelsat
```

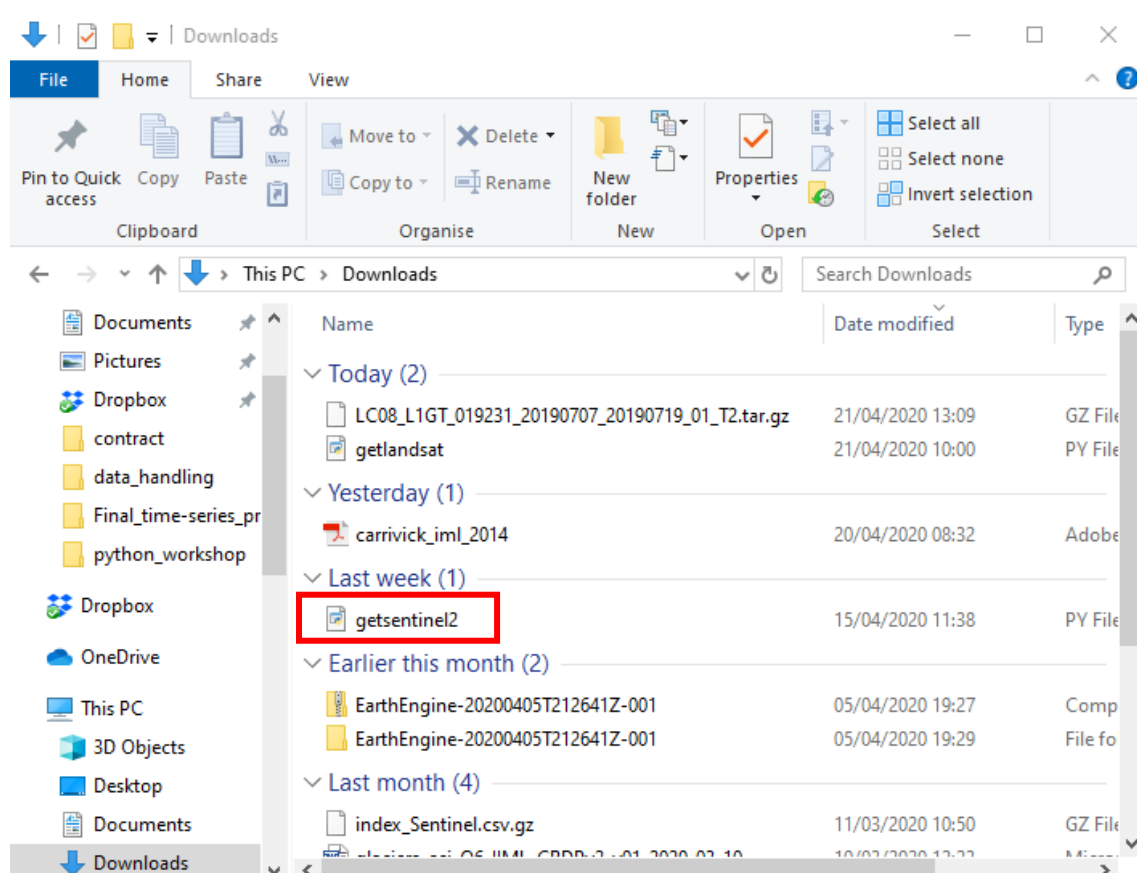
We now have a Conda environment called 'download' and has the SentinelSat package installed in it.

## 2. Getting the Python script and setting it up for downloading

Now we have our activated Conda environment, we need to get our Python script for batch downloading and choose a folder where our initial downloads will be inputted. First, make a copy of the Python script from the P: drive directory:

*P:\B35\_Remote\_Sensing\B35-01\_Projektstyring-og-Markedsføring\B35-01-004\_Strategi-og-Metode\Python tutorials\Batch downloading with Python\sentinel2\getsentinel2.py*

And place this file in a folder where you want your Sentinel imagery to be downloaded to. A suitable location would be the Downloads folder, for example.



To download Sentinel images from the SciHub online repository, we need to have login details (a username and password) to access SciHub. If you do not have an account with SciHub, you will need to make one at the web address <https://scihub.copernicus.eu/dhus/#/self-registration>.

Your SciHub login details need to be inputted into the getsentinel.py file, so that Sentinel-2 imagery can be retrieved from the repository through your SciHub account. Open the getsentinel.py file by double-clicking on it. This should open a Python console where you can scroll through the code and edit it. Scroll to the lines shown below; these should be near to the top of the script.



```
getsentinel2.py - C:\Users\how\Downloads\getsentinel2.py (3.6.2)
File Edit Format Run Options Window Help

import requests
import zipfile
import shutil
import glob
import sys
import os

sys.path.append('.')

#-----
#Activate parser and parser arguments
parser = argparse.ArgumentParser(description='A script to download Sentinel-2 '
+ 'scenes in batch based on an inputted AOI,' +
' and unzip them to a structured folder ' +
'system')

parser.add_argument('--aoi', required=True, type=str, help='Filepath to ' +
'geojson polygon denoting the area of interest, or list ' +
'of tile identifiers')

parser.add_argument('--date1', required=True, type=str,
help='Start date for Sentinel scenes, yyyyymmdd')

parser.add_argument('--date2', required=True, type=str,
help='End date for Sentinel scenes, yyyyymmdd')

parser.add_argument('--loc1', default=Path(__file__).parent.absolute(),
type=str,
help='Directory (folder location) for downloaded files')

parser.add_argument('--loc2', default="D:/data", type=str,
help='Directory (folder location) for unzipped files')

parser.add_argument('--cloud', default=None, type=tuple,
help='Cloud cover percentage range. E.g. (0,5) denotes ' +
'a percentage range of 0 to 5%')

parser.add_argument('--user', default="guest", type=str,
help='SciHub username')

parser.add_argument('--pwd', default="guest", type=str,
help='SciHub password')

parser.add_argument('--prod', default="S2MSI1C", type=str,
help='Product type')

parser.add_argument('--over', default=False, type=bool, help='Flag denoting ' +
'if unzipped files will overwrite pre-existing files')

parser.add_argument('--offline', default=False, type=bool,
help='Flag denoting if offline products will be requested' +
' from the Sentinel LTA archive')

#Retrieve arguments
args = parser.parse_args()

#Retrieve AOI file or tile identifier list

Ln: 23 Col: 13
```

Currently the SciHub username and password are set to 'guest'. This is a guest account, which is valid for logging in but does not allow Sentinel-2 image downloading. We need to edit the two commands highlighted with our own SciHub username and password in order to gain permission to download. In each command, replace the word 'guest' with your username (for the first command) and password (for the second command). Remember that there should be quotation marks around your username and password, and the words should therefore be highlighted in green (as above). After you are modified the script, save it. For each time we run this script, your user details will always be used to log into SciHub now.

Close the Python script and look at your command line again. We now have to navigate to the directory our Python script is in using command line, which is where we will run this Python script. Use the command 'cd' to navigate to a file directory in command line, inputting the folder name after to navigate into a given folder.

```
(download) C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3-clone>cd/  
(download) C:\>cd Users/how/Downloads  
(download) C:\Users\how\Downloads>
```

In this example, we first type the command `'cd/'` to navigate back up the C: directory, then `'cd Users/how/Downloads'` to navigate to the Downloads folder in my user space (user 'how'). If you are unsure of the contents of the directory you are in, type `'dir'` to print the contents of the directory.

We are now ready to run the Python script and start downloading our images.

### 3. Running the Python script

To run the script, we type the command `'python getsentinel2.py'` into the command line, signifying to open a Python window and run the getsentinel.py script through it.

```
python getsentinel2.py
```

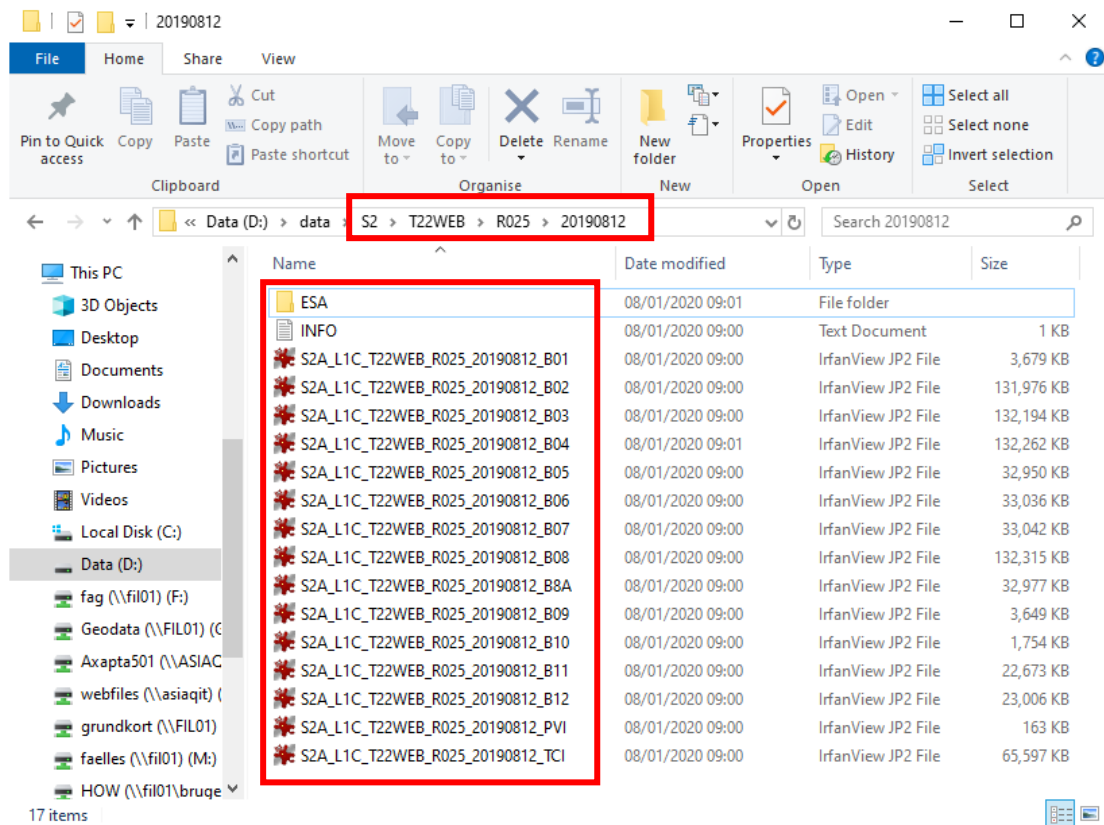
There are a couple of variables we need to define in order to download and retrieve our desired satellite imagery. There are three mandatory variables that are needed in order to make the script run – the area of interest (AOI) you wish to download scenes from (as either a list of tile ids, or a geojson file containing a polygon of your AOI), the start date for the download, and the end date for the download – which will look something like this:

```
python getsentinel2.py --aoi 22WEB --date1 20190801 --date2 20190830
```

In this example, we are looking for tiles with the identification of 22WEB (which is the Jakobshavn Isbræ catchment) between the dates 01/08/2019 and 30/08/2019. When you hit enter, a number of steps happen which are printed in the command line as they occur:

- A. Your log in details are used to sign into SciHub (which is where the Sentinel-2 scenes are held online). A print statement will signify if your login details are successful
- B. A list of scenes satisfying your inputted parameters will be retrieved
- C. Each product will be checked to see if it has already been downloaded. If a product already exists offline then it will move to the next product. If a product has not been downloaded previously then an enquiry is sent to the SciHub repository to see if the product is available for download
  - i. If a product is available to download from SciHub, then the download will start
  - ii. If a product is not available to download from SciHub, then the product will be searched for in the Google Cloud repository and downloaded from there if it exists
- D. Once downloaded, each product is unzipped and moved into a common folder directory structured as *platform >> tile identification >> relative orbit >> acquisition date*.

In each downloaded scene folder (i.e. after *'acquisition date'*), the band .jp2 files are in the top level and the remaining files (e.g. auxiliary files, cloud mask files) can be found in the *'ESA'* folder. Additionally, a readme file is generated with each downloaded scene called *'INFO.txt'*, which contains the metadata for the scene, the date the scene was downloaded, and who downloaded it (i.e. your Asiaq username).



Above is a downloaded and unzipped scene from our example download in this document – this is a scene from tile 22WEB on 12/08/2019. Note the folder structure and the contents of the directory – each band .jp2 file, the ESA folder containing all other scene data, and the INFO.txt file containing the metadata file.

#### 4. Adapting the input variables to refine your scene downloads

Above is an example of where we download Sentinel-2 scenes using the three mandatory variables – the AOI, start date and end date. There are other optional variables that can be inputted to refine the scene downloads and modify the output locations, which are summarised in the table below.

Variable	Name	Notes	Mandatory/optional variable?	Default input
--aoi	Area of interest	List of tile identifiers or filepath to geojson polygon file denoting the area of interest	Mandatory	N/A
--date1	Start date	Start date for scene download	Mandatory	N/A
--date2	End date	End date for scene download	Mandatory	N/A
--loc1	Download folder location	Folder location to download zip files to	Optional	Directory that the Python script is in*
--loc2	Unzip folder location	Folder location to unzip files to in the set folder structure	Optional	'G:/Satellitdata/S2'
--cloud	Cloud cover	Cloud cover percentage range; for example, '(0,5)' denotes a percentage range of 0% to 5%	Optional	None
--user	SciHub username	Username for logging in to SciHub	Optional	'guest'
--pswr	SciHub password	Password for logging in to SciHub	Optional	'guest'

<b>--prod</b>	Product type	Product type to download (e.g. S2MSI1C or S2MSI2A)	Optional	'S2MSI1C'
<b>--over</b>	Overwrite flag	Flag denoting whether files that exist should be overwritten with newly downloaded version	Optional	False
<b>--offline</b>	Offline products flag	Flag denoting whether offline products should be retrieved from the SciHub repository	Optional	False

\* `pathlib.Path(__file__).parent.absolute()`

For example, if we want to redefine the downloaded folder location and the unzipped folder location for the Sentinel-2 downloads, we can run the script with the '`--loc1`' and '`--loc2`' variables, in addition to our three mandatory variables:

```
python getsentinel2.py --aoi 22WEB --date1 20190801 --date2 20190830 --loc1 D:/data/downloaded --loc2 D:/data/unzipped_
```

In this instance, the zip files from SciHub will be downloaded to our directory defined by '`--loc1`', and unzipped to our directory defined by '`--loc2`'.