BTS | Barcelona Technology School

Data Science Foundations

Master in Big Data Solutions 2019-2020

Víctor Pajuelo

victor.pajuelo@bts.tech

Today’s class

Contents

2. Loading and processing images and text
   - Image loading, pre-processing and filtering
   - Image pre-processing for object detection and segmentation
   - Text pre-processing, normalization, stemming, stopword removal
   - Converting text to vectors and computing text similarity

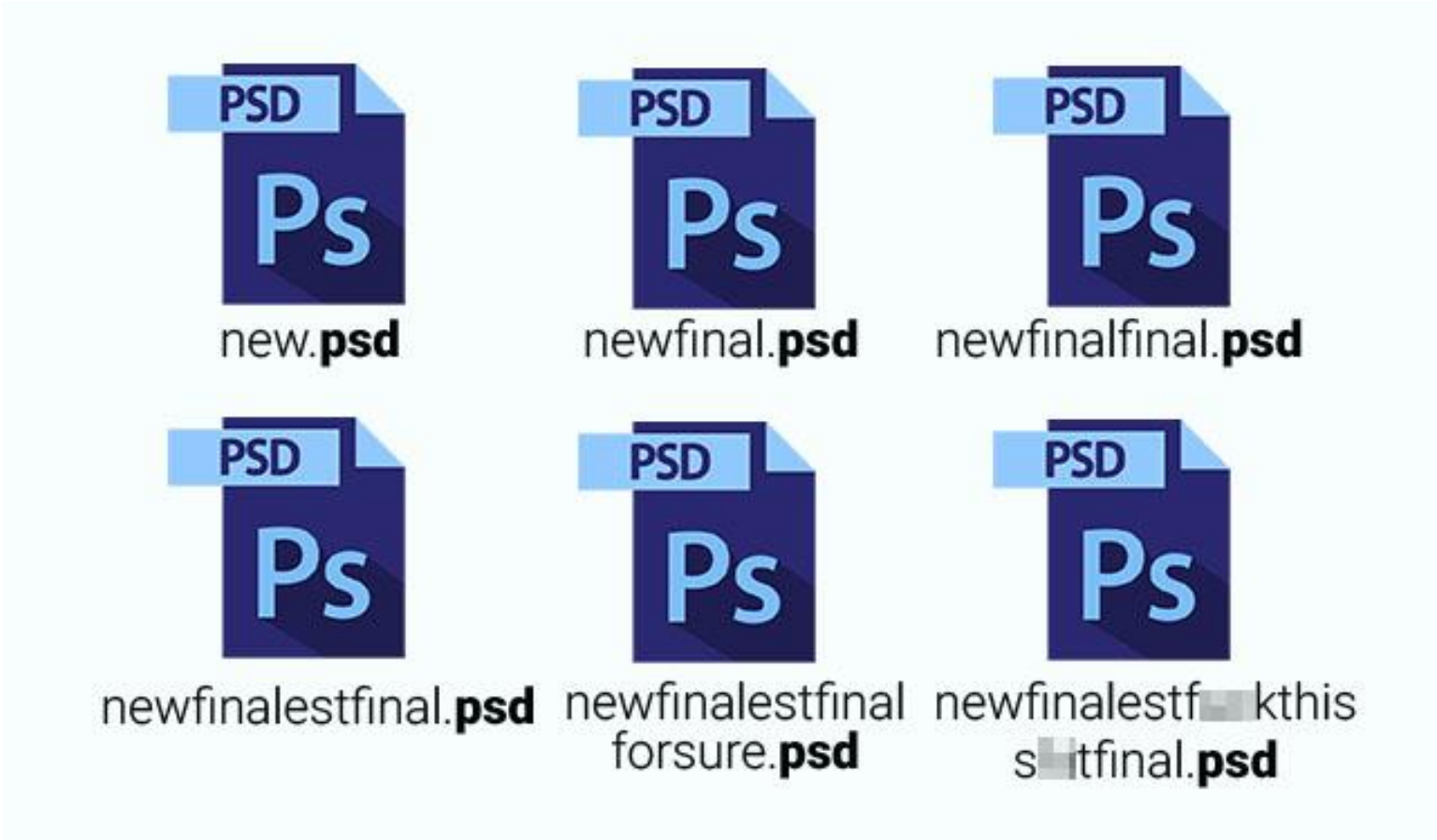
Today's Objective

- Get familiar with Git
- Get acquainted with the SciPy ecosystem
- Get to know your friend NumPy
- Start processing and filtering images using Scikit Image

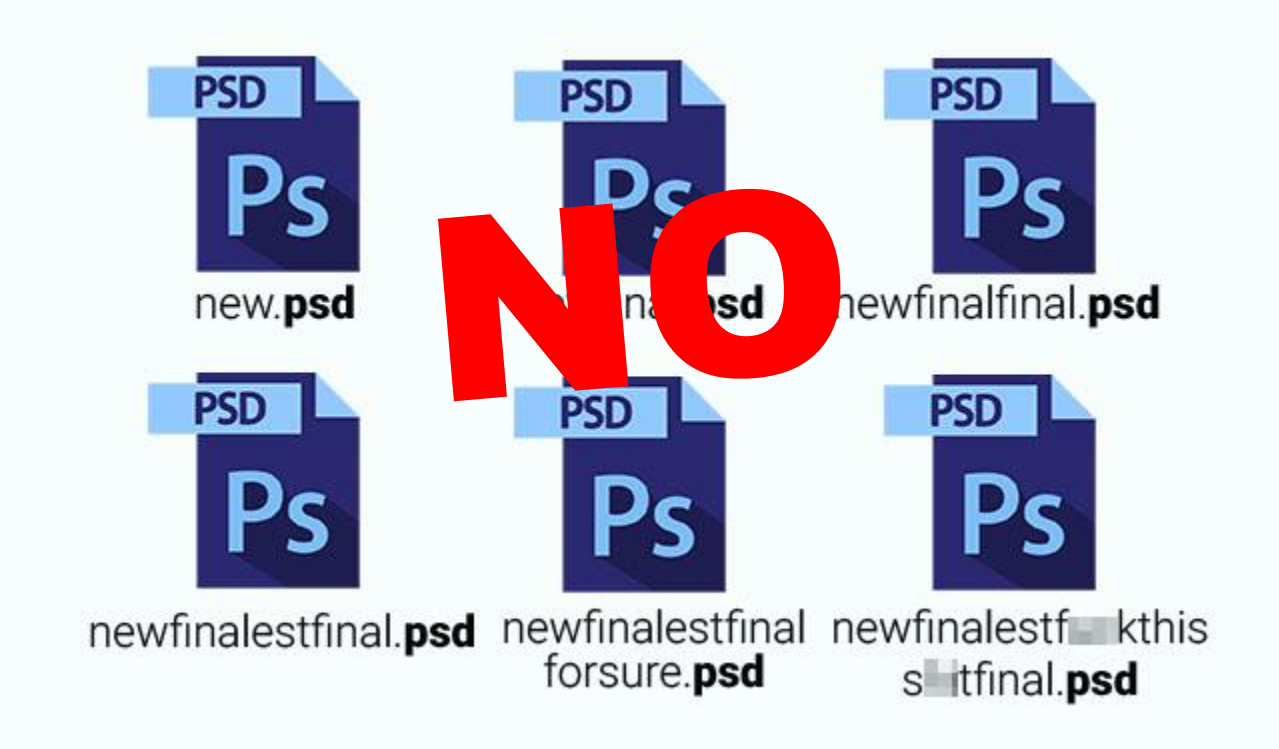
- Why is this useful for a digital project?
  - Git version control it is crucial for maintaining the development of a digital project
  - Array, matrix and image manipulation is a vital skill for many type of digital projects, not only dealing with images, but also audio or text.

Let's git things done!

Why version control?

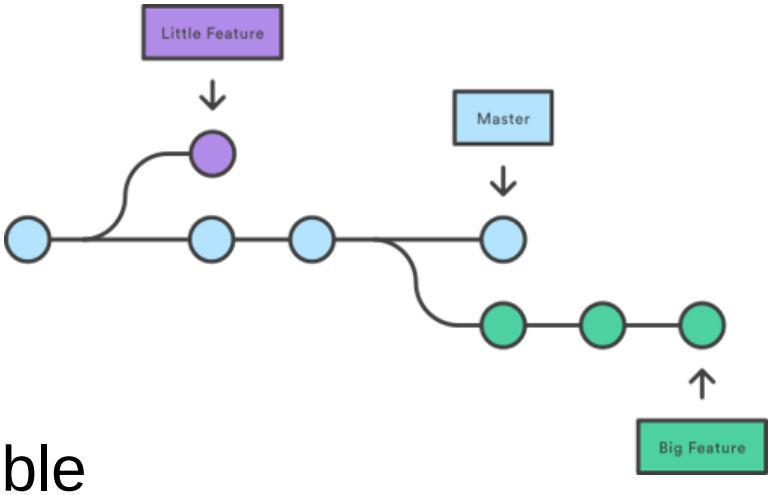


Why version control?



Introduction to git

- Fast, distributed version control system
- Can work locally as well as against an online project
- Creating branches is easy, many different workflows possible



- ...Lots of jargon, documentation difficult to read, many moving parts

GitHub

- GitHub.com is a code hosting platform *that uses git*
- Currently the biggest in terms of users and project activity
- Issue tracker, wiki, "pull requests", previsualization of common file formats, integration with external services...
- Bought by Microsoft in summer 2018

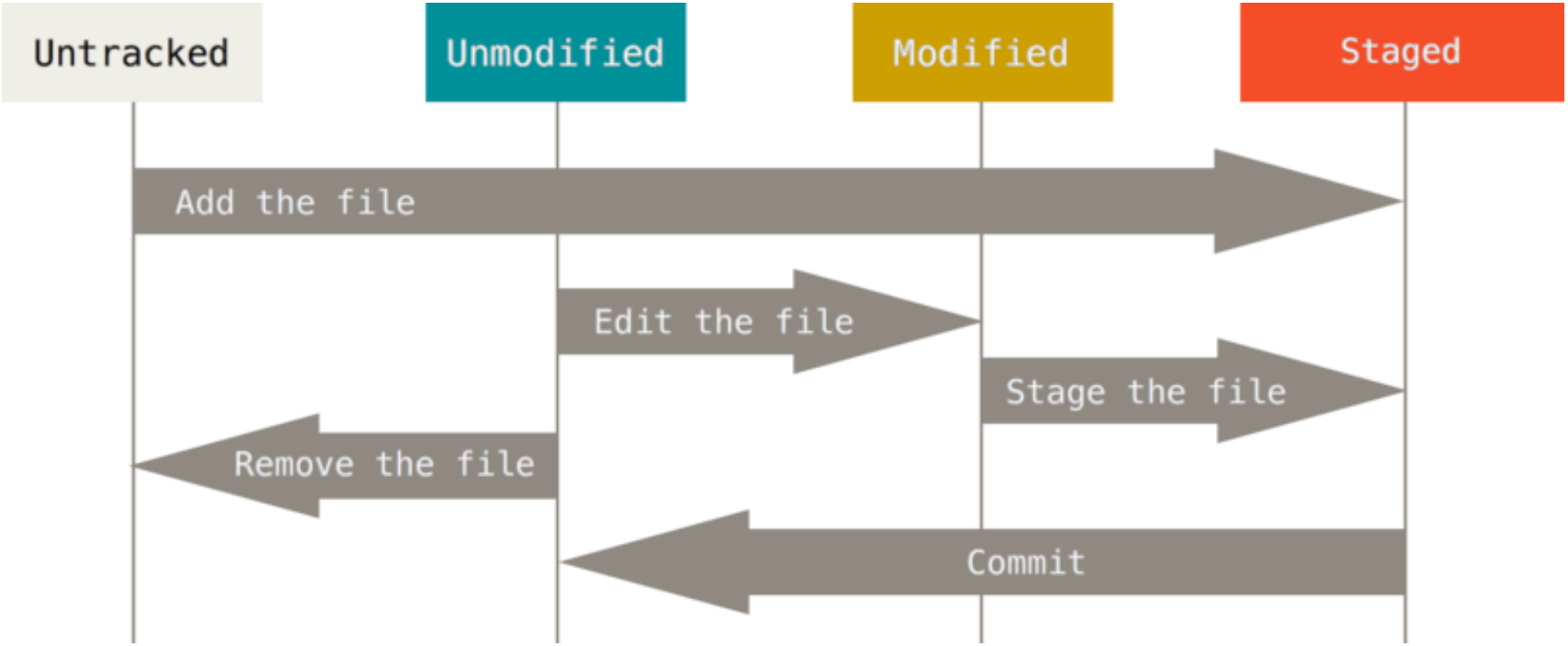
Configuring git

- Configuration https://help.github.com/articles/setting-your-commit-email-address-in-git/

```
$ git config --global user.name “Victor"
$ git config --global user.email victor.pajuelo@bts.tech
$ git config --global core.editor nano
```

First steps and lifecycle

- First, start by creating a local repository using `git init`
- All files will go through this lifecycle:



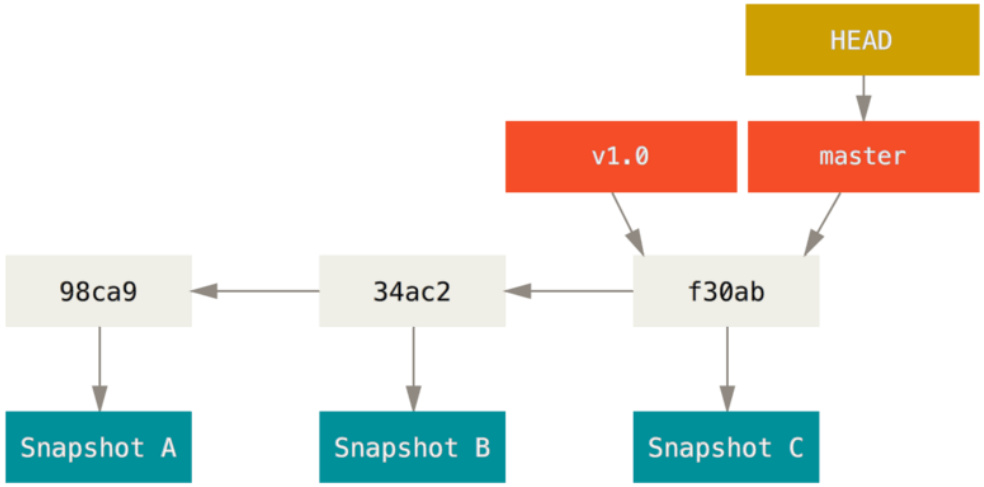
Simplest workflow

1. Create file with some contents or edit an existing one
2. Add it to the "staging area" using `git add`
3. Create a commit (a snapshot) using `git commit`
4. Go to 1

- Some useful commands:
  - git status – Check current commit, modified files...
  - git log – Check latest commits
  - git show – Show last commit

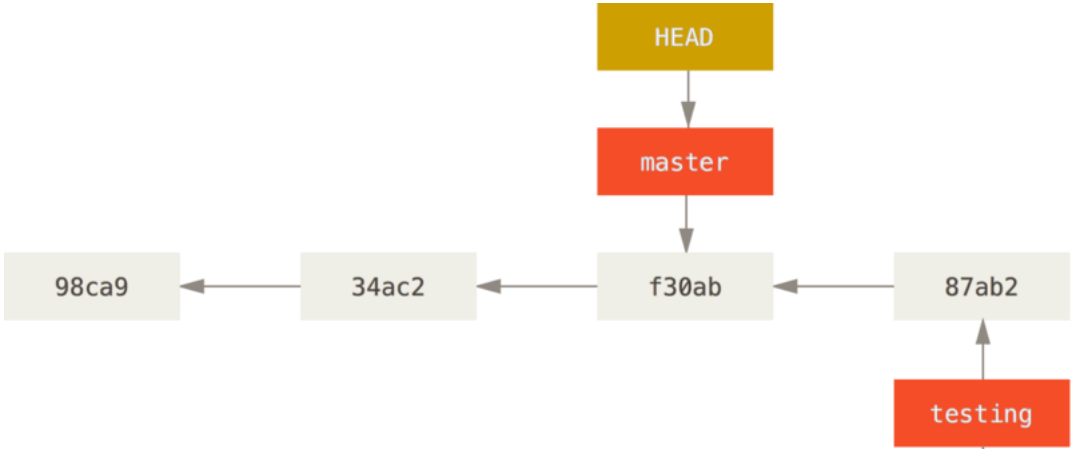
Branches and merging

- Branches divert from the main project history
- They let us explore, fix bugs... without affecting the main branch, called `master`



Creating and navigating branches

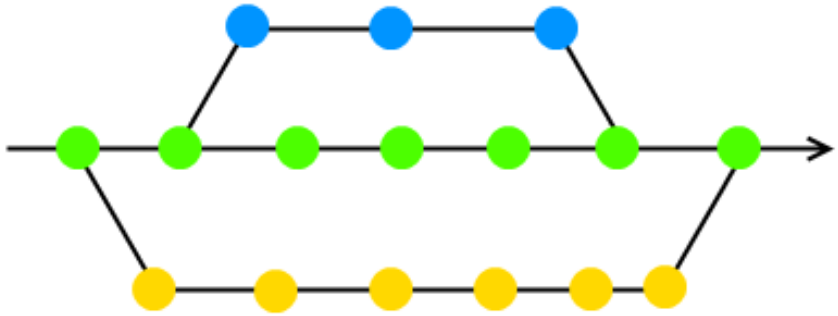
- Two basic commands:
  - git branch – Creates new branch
  - git checkout – Changes the current active branch
    - One can also checkout a specific commit in history – this is called "detached HEAD" status and must be handled with care
- Once in a different active branch, commits will go under that name and the old one will be left behind



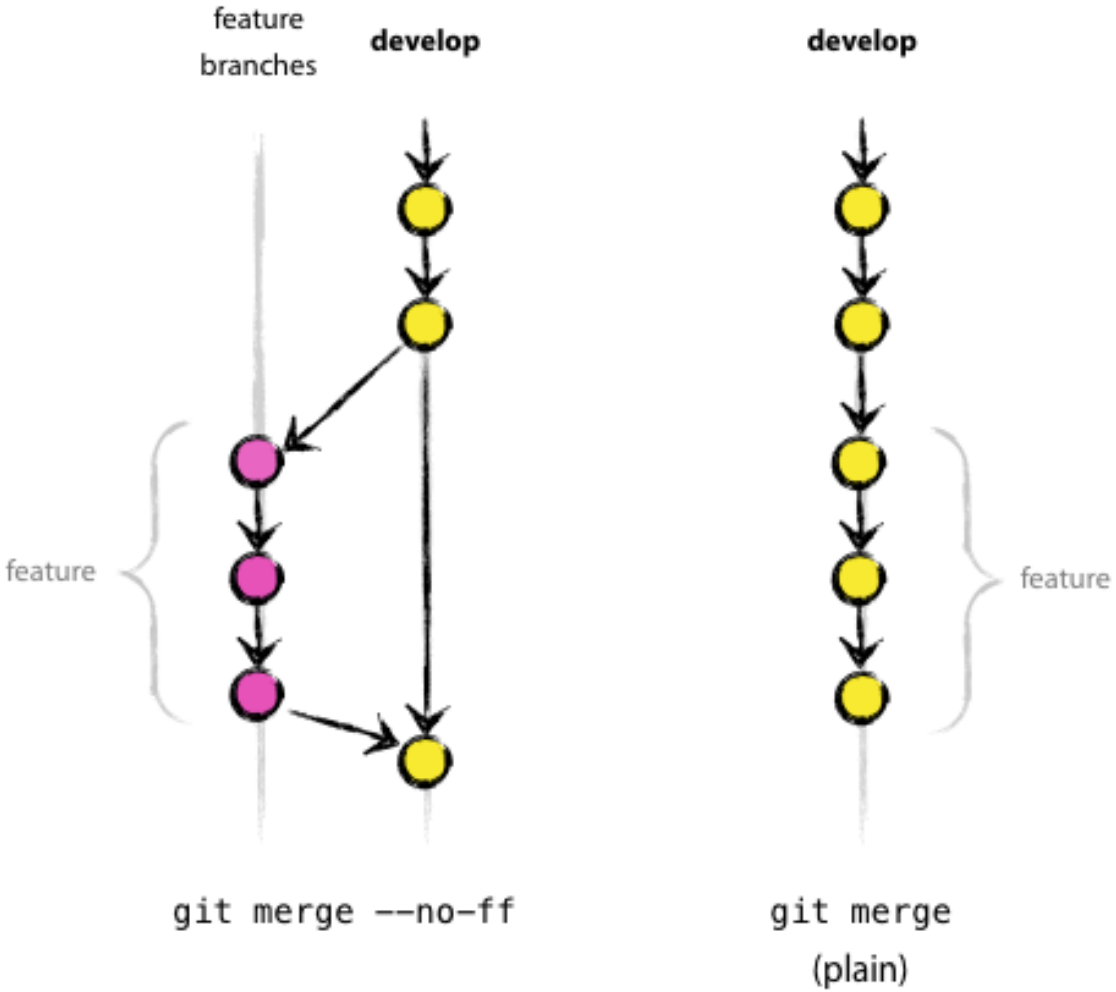
Merging

- Merging involves two branches, and melds their history
- Conflicts might appear, and it's important to solve them before committing, possibly using graphical tools

```
$ git checkout testing
# Do some work and commit...
$ git checkout master
$ git merge --no-ff testing # No fast forward
```



--no--ff


feature
branches
develop
feature
git merge --no-ff
develop
feature
git merge
(plain)

...and there's more!

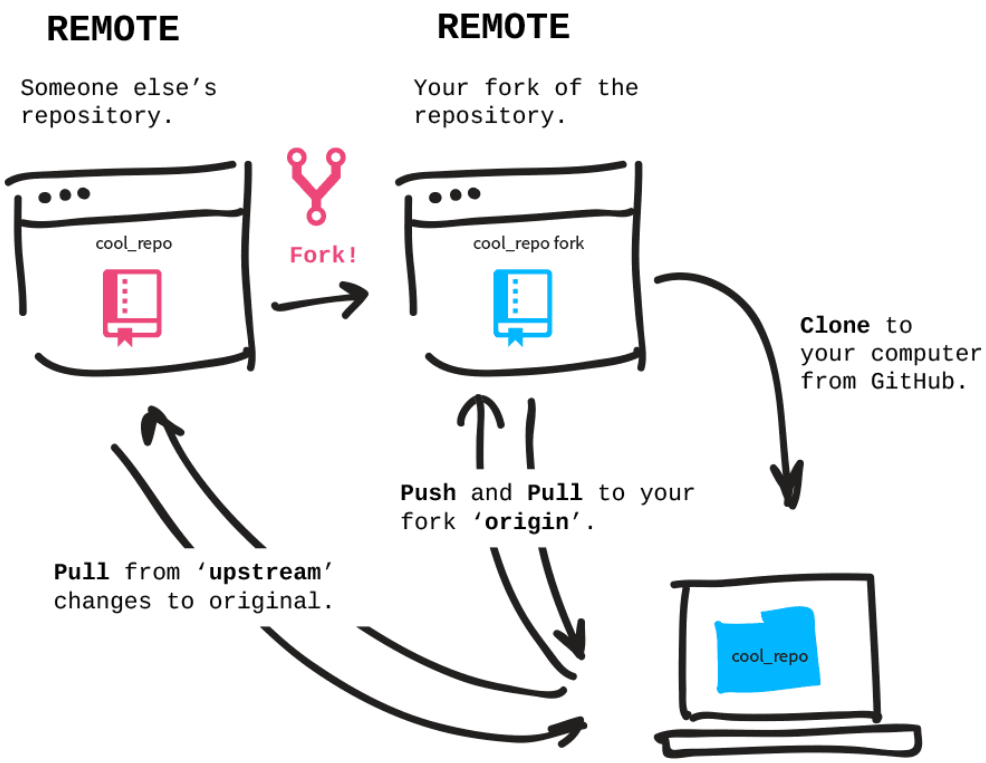
- There's muuuuuuch more than this
  - Rebase, stash, reset, tags...
  - Sometimes, add & commit is just what you need
  - When collaborating, branches will help, and conflicts will appear
- Jupyter notebooks and git... Painful, avoid conflicts at all cost
- You **will screw up** – it's not nice when it happens, but it's not the end of the world
- When in doubt, http://justinhileman.info/article/git-pretty/

And, from now on...

```
$ git clone https://github.com/vfp1/bts-mbds-data-science-foundations-2019.git
# Some time passes...
$ git fetch  # There are changes!
$ git checkout master  # Just in case
$ git merge --ff-only origin/master  # If in error, you probably made some commits to master
```

Or, in case that you preferred a fork…

- https://help.github.com/en/articles/syncing-a-fork



Our journey through the SciPy ecosytem

SciPy ecosystem

https://www.scipy.org/
https://scipy-lectures.org

NumPy
Base N-dimensional array package

SciPy library
Fundamental library for scientific computing

Matplotlib
Comprehensive 2D Plotting

IP[y]: IPython

IPython
Enhanced Interactive Console

Sympy
Symbolic mathematics

pandas
Data structures & analysis

SciPy ecosystem

https://www.scipy.org/
https://scipy-lectures.org

NumPy
Base N-dimensional array package

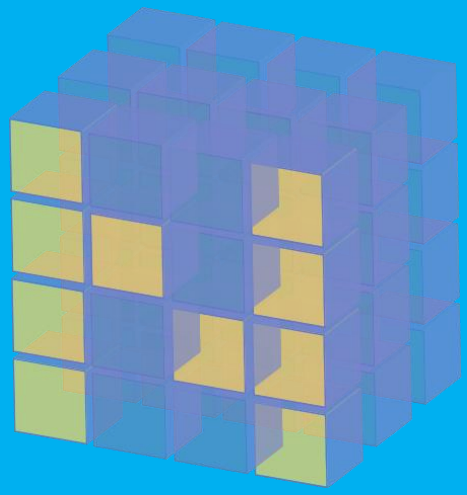
SciPy library
Fundamental library for scientific computing

Matplotlib
Comprehensive 2D Plotting

IP[y]: IPython
IPython
Enhanced Interactive Console

Sympy
Symbolic mathematics

pandas
Data structures & analysis



NumPy

NumPy Arrays

https://numpy.org/

An **array can contain**:

- Values of an experiment/simulation at discrete time steps
- A signal recorded by a measurement device, e.g. sound wave
- The pixels of an image, grey-level or colour
- 3D data measured at different XYZ positions, e.g. MRI scan
- Others...

NumPy

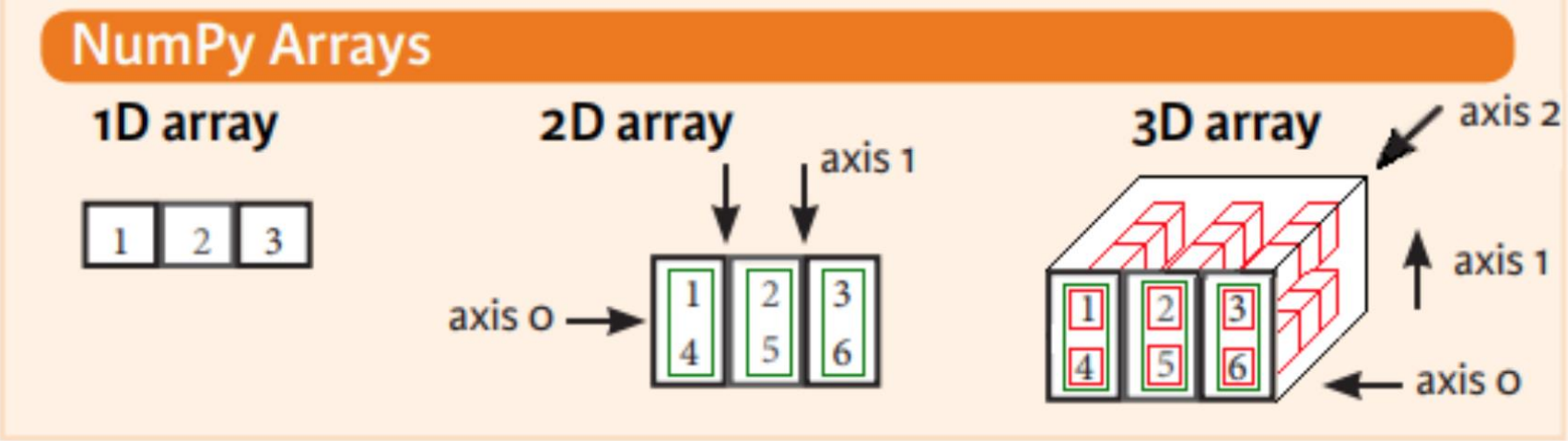
https://numpy.org/

The n-dimensional array (ndarray) computation provides extremely fast computing as we will see in the notebook

Python objects:	**• high-level number objects: integers, floating point** **•containers: lists (costless insertion and append), dictionaries (fast lookup)**
NumPy provides:	•extension package to Python for multi-dimensional arrays •closer to hardware (efficiency) •designed for scientific computation (convenience) •Also known as array oriented computing

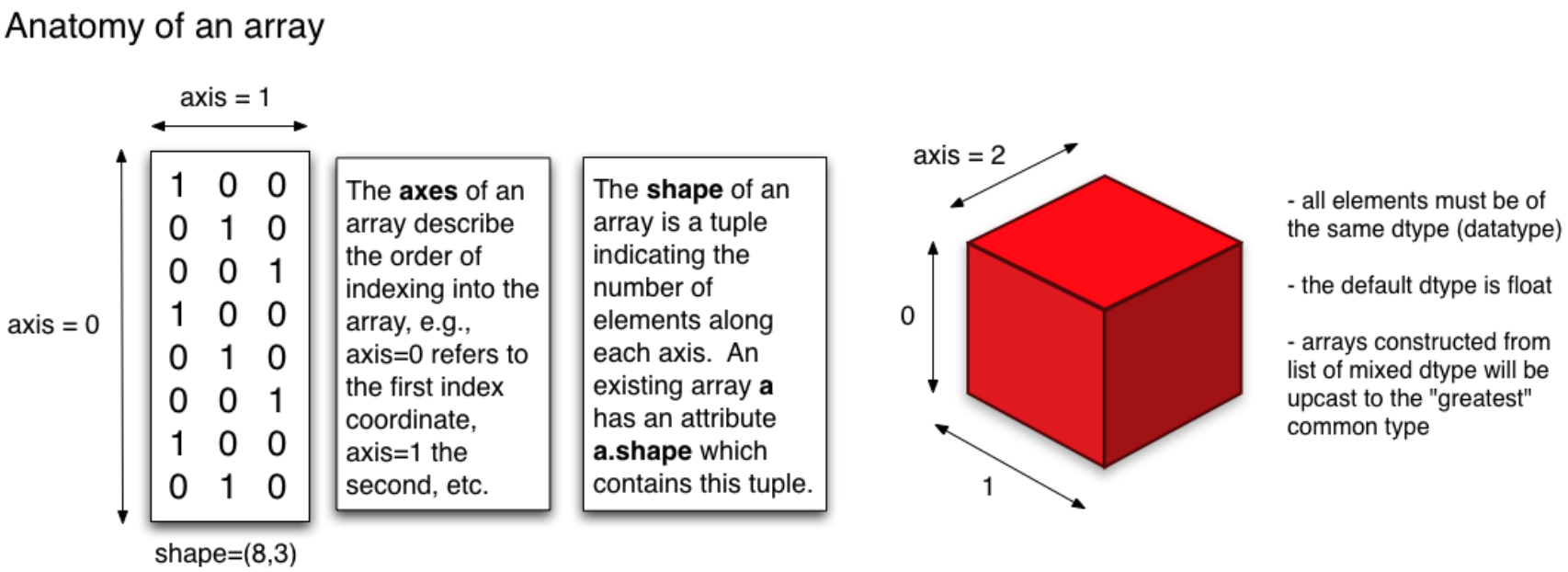
NumPy Arrays

https://numpy.org/



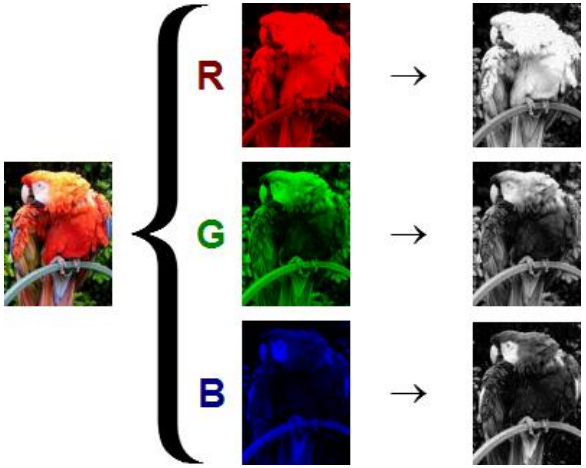
NumPy Arrays

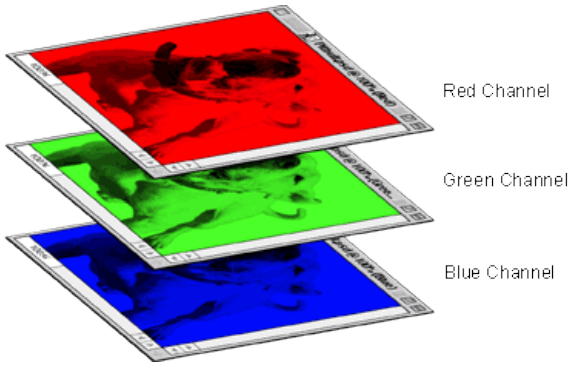
https://numpy.org/

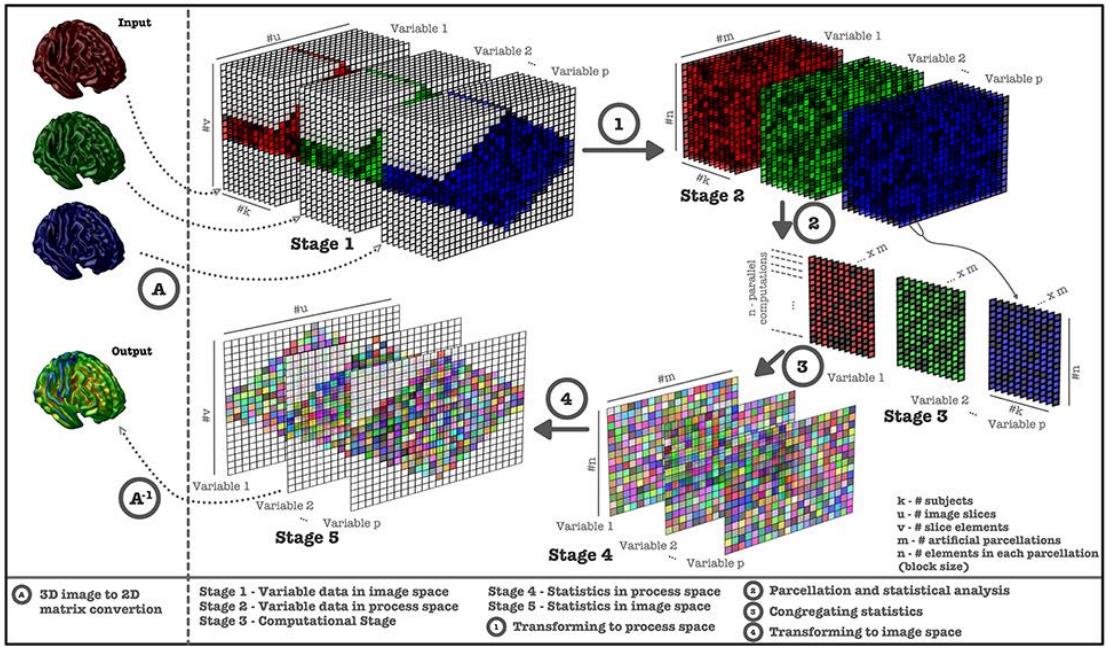


3D array examples

https://numpy.org/







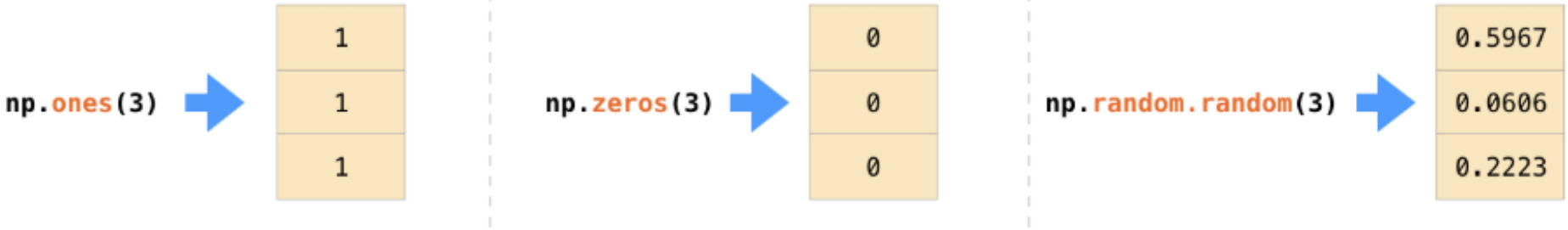
NumPy visualized (1D)

http://jalammar.github.io/visual-numpy/

```
data = np.array([1,2,3])
```

Generate NumPy arrays (1D)

http://jalammar.github.io/visual-numpy/



Array computation (1D)

http://jalammar.github.io/visual-numpy/

```
data = np.array([1,2])
```

data

1
2

```
ones = np.ones(2)
```

ones

1
1

```
data + ones
```

=

data

1
2

+

ones

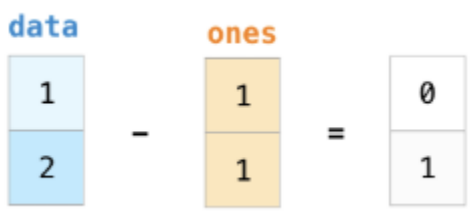
1
1

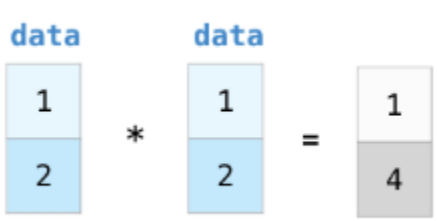
=

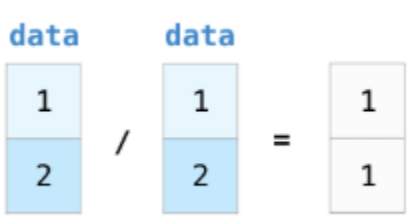
2
3

Array Computation (1D)

http://jalammar.github.io/visual-numpy/

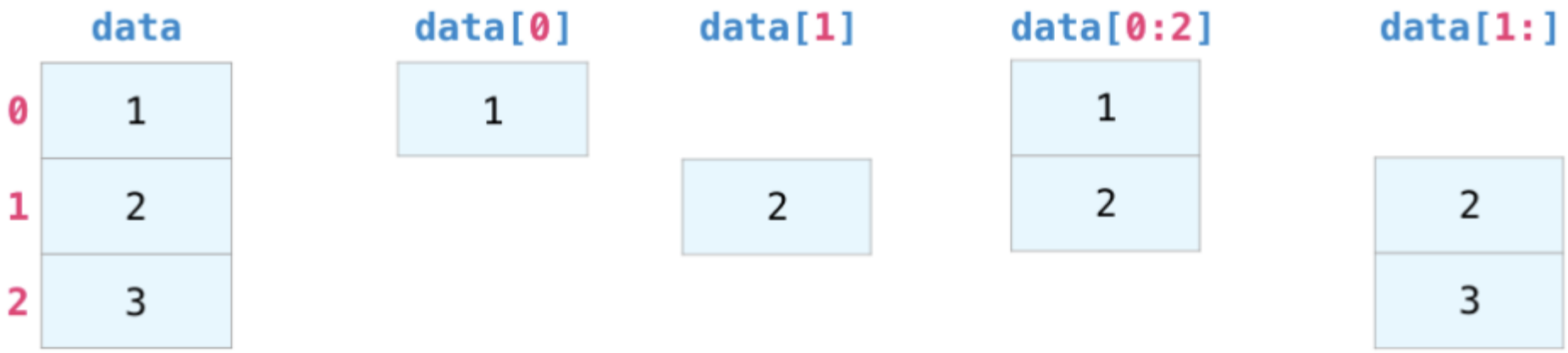






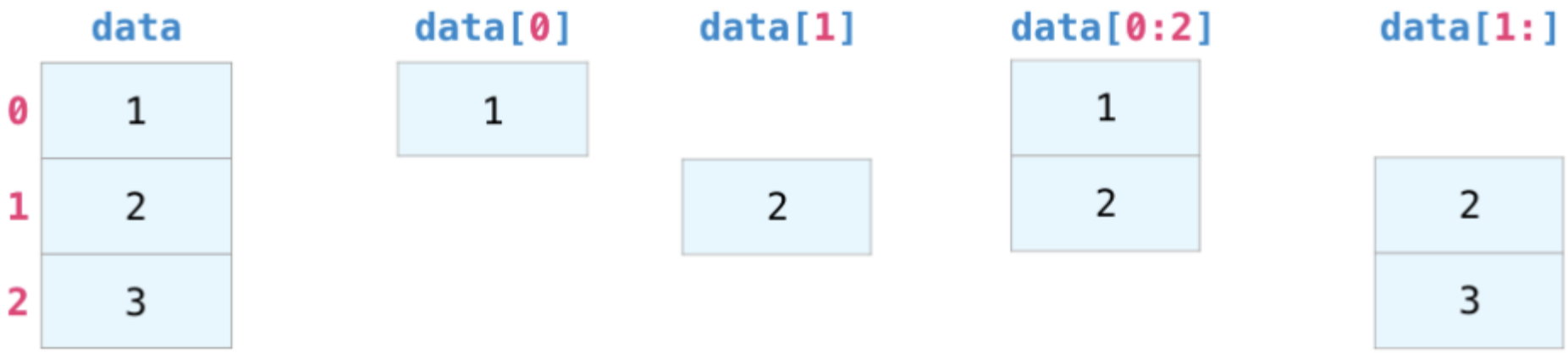
Indexing (1D)

http://jalammar.github.io/visual-numpy/



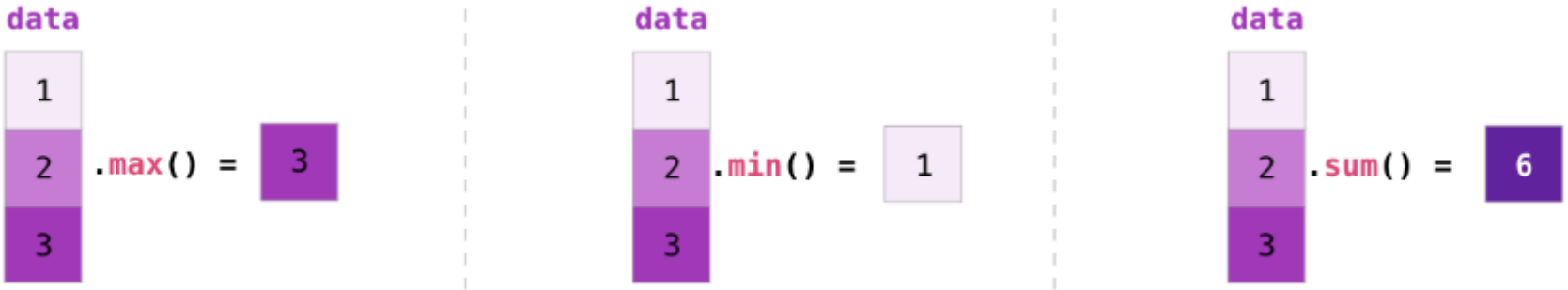
Indexing (1D)

http://jalammar.github.io/visual-numpy/



Aggregation (1D)

http://jalammar.github.io/visual-numpy/

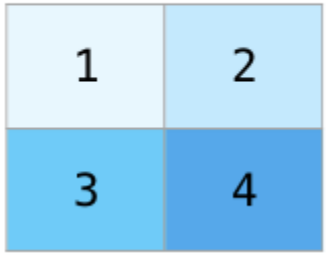


Matrices (2D)

http://jalammar.github.io/visual-numpy/

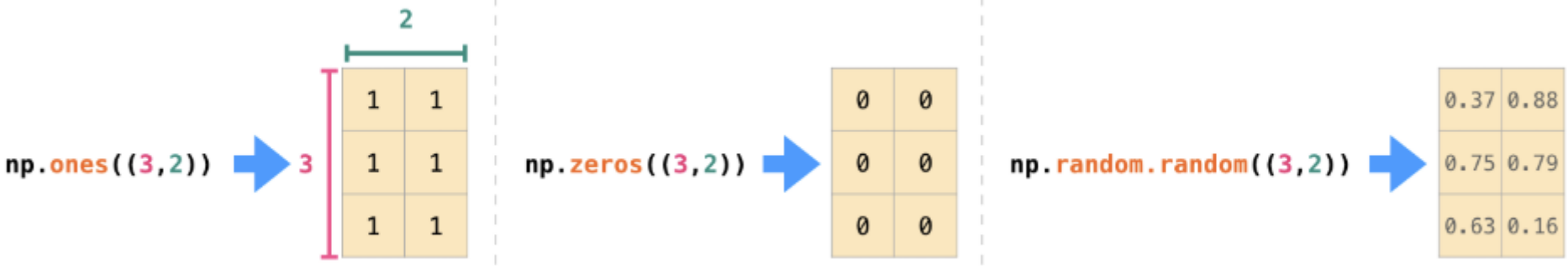
```
np.array([[1,2],[3,4]])
```

np.array([[1,2],[3,4]])



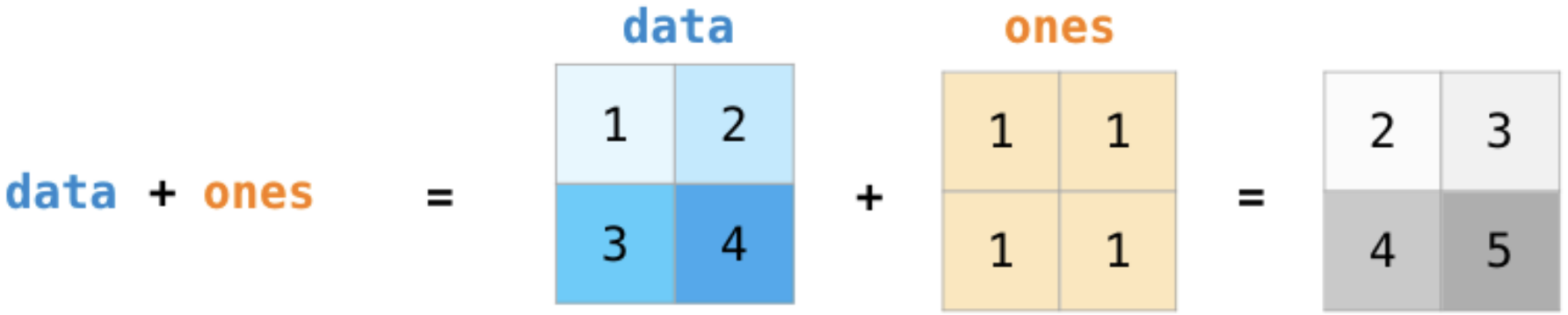
Matrices (2D)

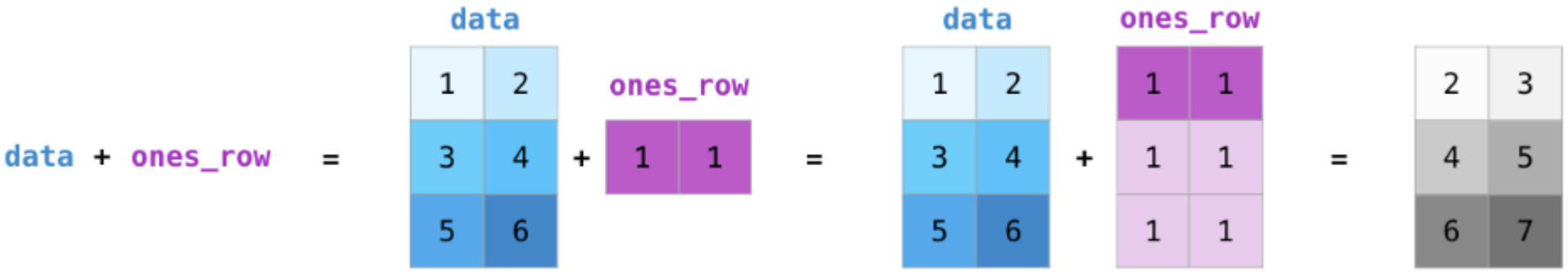
http://jalammar.github.io/visual-numpy/



Matrix computation (2D)

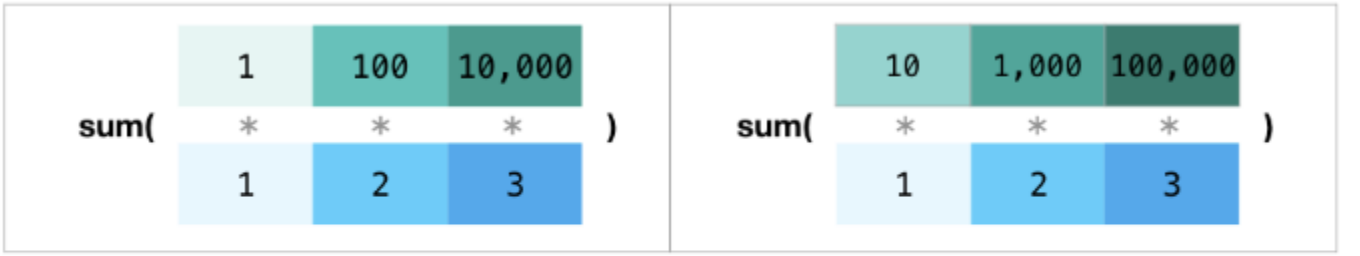
http://jalammar.github.io/visual-numpy/





Matrix Dot Product (2D)

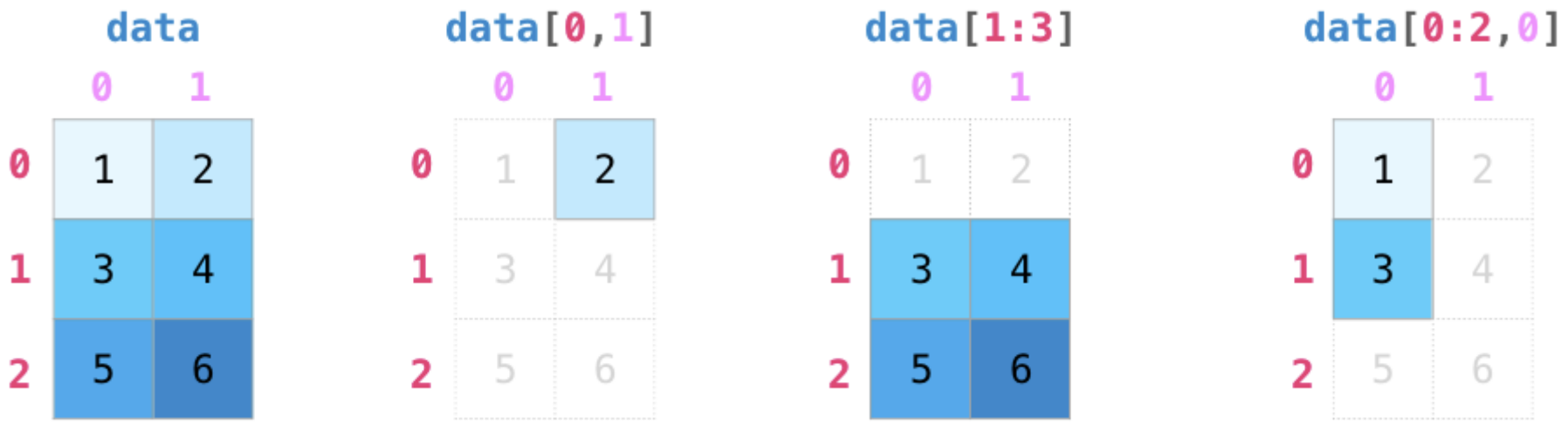
http://jalammar.github.io/visual-numpy/



1*1 + 2*100 + 3*10,000 | 1*10 + 2*1,000 + 3*100,000 = 30201 | 302010

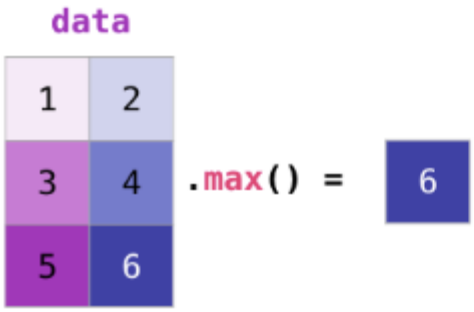
Matrix Indexing (2D)

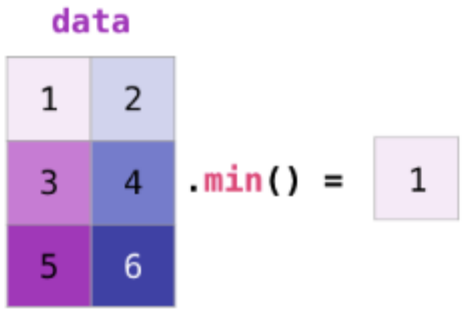
http://jalammar.github.io/visual-numpy/

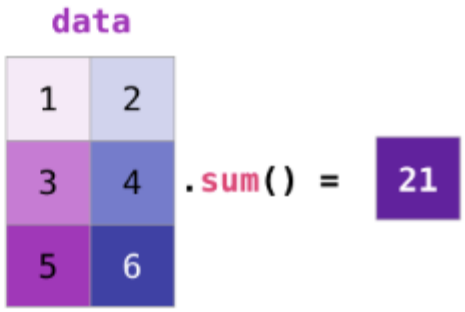


Matrix Aggregation (2D)

http://jalammar.github.io/visual-numpy/

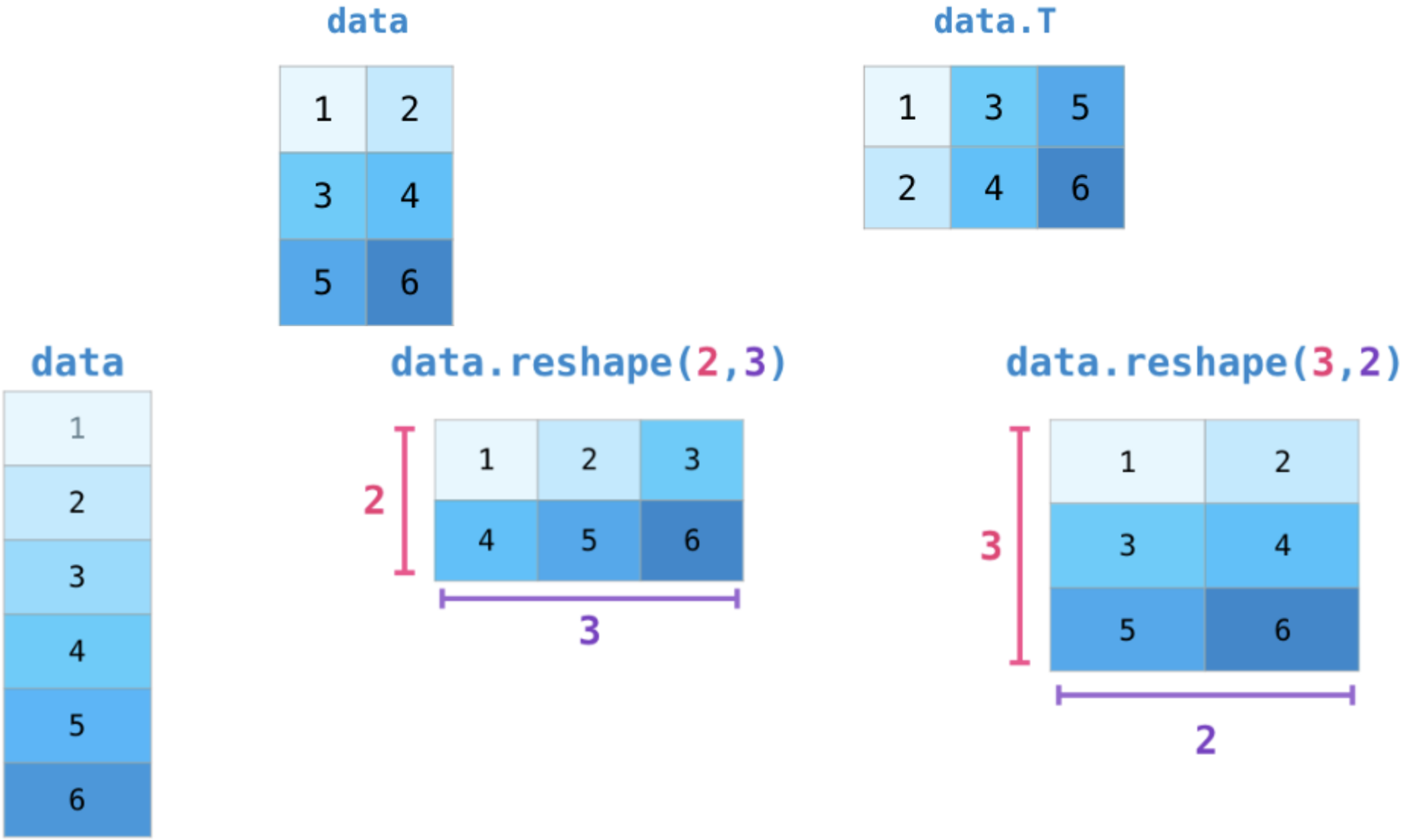






Transposing and Reshaping (2D)

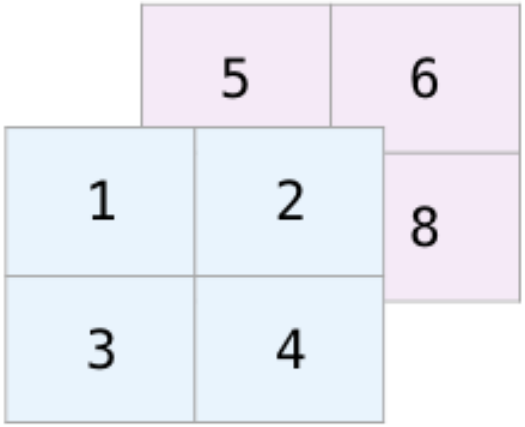
http://jalammar.github.io/visual-numpy/



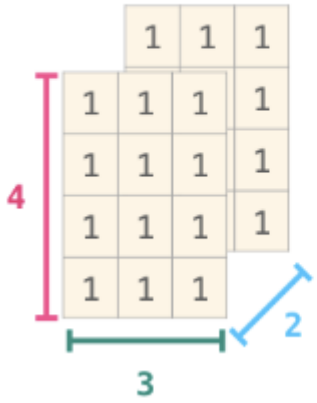
nDarray (3D)

http://jalammar.github.io/visual-numpy/

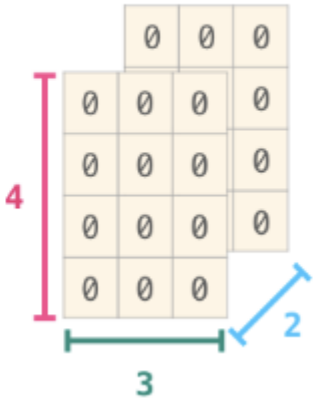
```
np.array([ [[1,2],[3,4]],
           [[5,6],[7,8]] ])
```



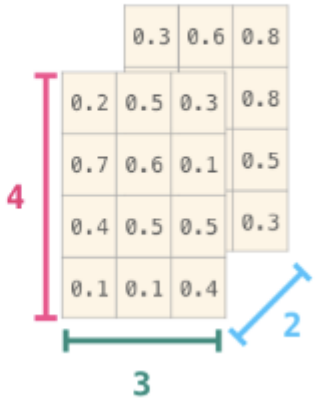

```
np.ones((4,3,2))
```



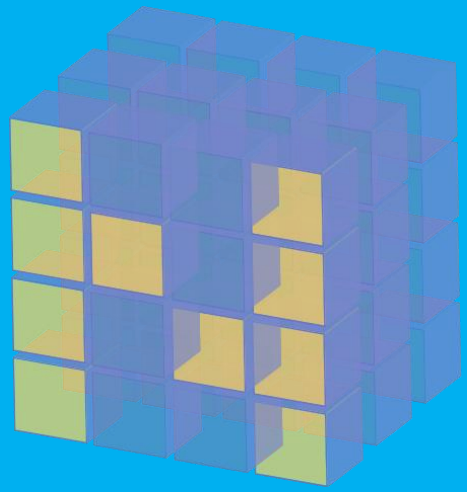

```
np.zeros((4,3,2))
```




```
np.random.random((4,3,2))
```



Use cases of



NumPy

Mathematical operations

http://jalammar.github.io/visual-numpy/

$$MeanSquareError = \frac{1}{n}\sum_{i=1}^{n}(Y\_prediction_i - Y_i)^2$$

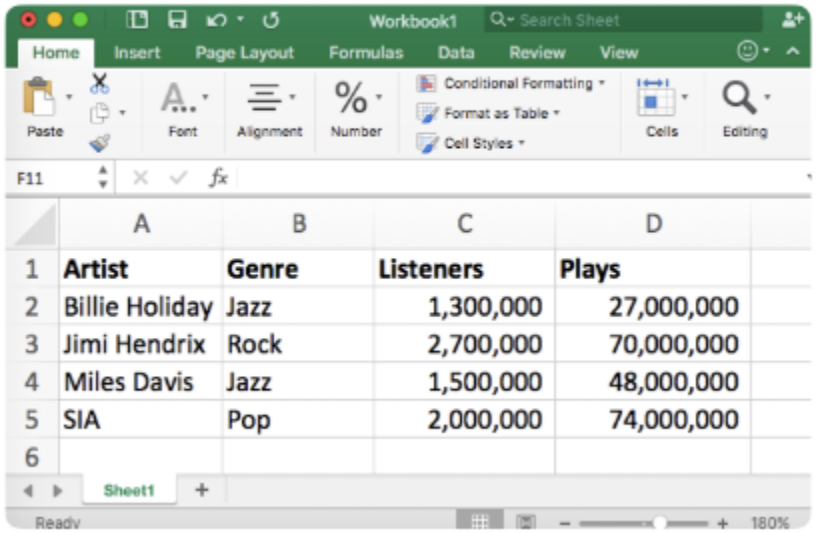
```
error = (1/n) * np.sum(np.square(predictions - labels))
```

```
                                  predictions   labels
error = (1/3) * np.sum(np.square(  | 1 |     -  | 1 |  ))
                                   | 1 |        | 2 |
                                   | 1 |        | 3 |
```

Pandas!!!!!

http://jalammar.github.io/visual-numpy/

music.csv



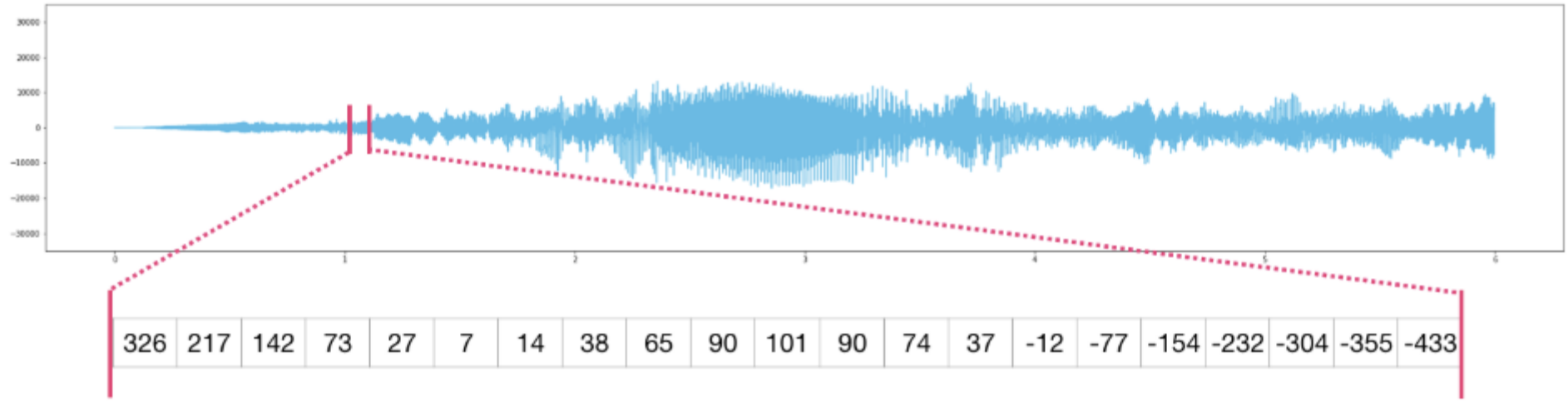
	A	B	C	D
1	Artist	Genre	Listeners	Plays
2	Billie Holiday	Jazz	1,300,000	27,000,000
3	Jimi Hendrix	Rock	2,700,000	70,000,000
4	Miles Davis	Jazz	1,500,000	48,000,000
5	SIA	Pop	2,000,000	74,000,000
6				

pandas.read_csv('music.csv')

	Artist	Genre	Listeners	Plays
0	Billie Holiday	Jazz	1,300,000	27,000,000
1	Jimi Hendrix	Rock	2,700,000	70,000,000
2	Miles Davis	Jazz	1,500,000	48,000,000
3	SIA	Pop	2,000,000	74,000,000

Audio & Timeseries

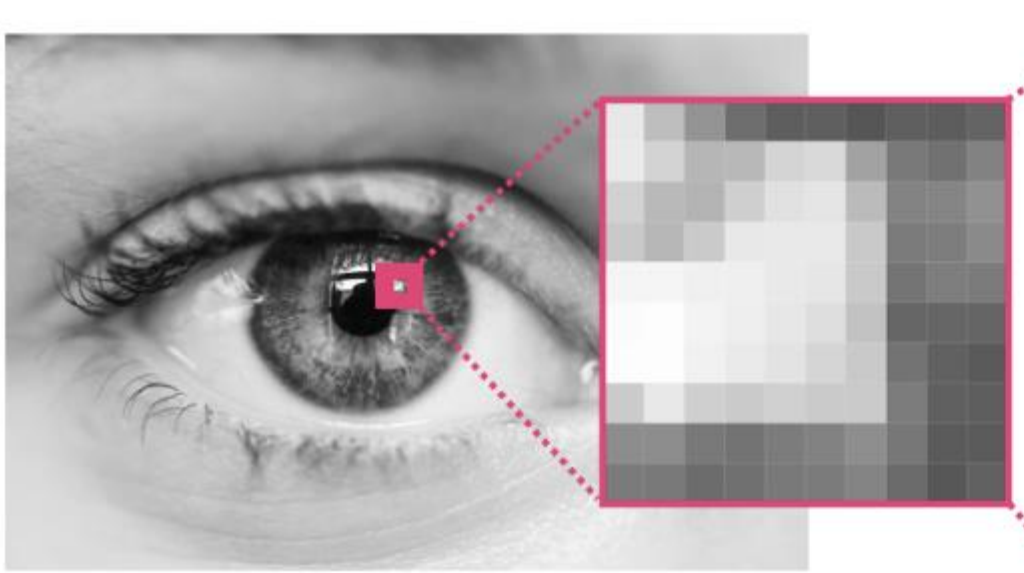
http://jalammar.github.io/visual-numpy/



Usually a 1D array (if it is not a MEL spectrogram).
NumPy is extremely useful here for slicing and selecting parts of the audio.

Images

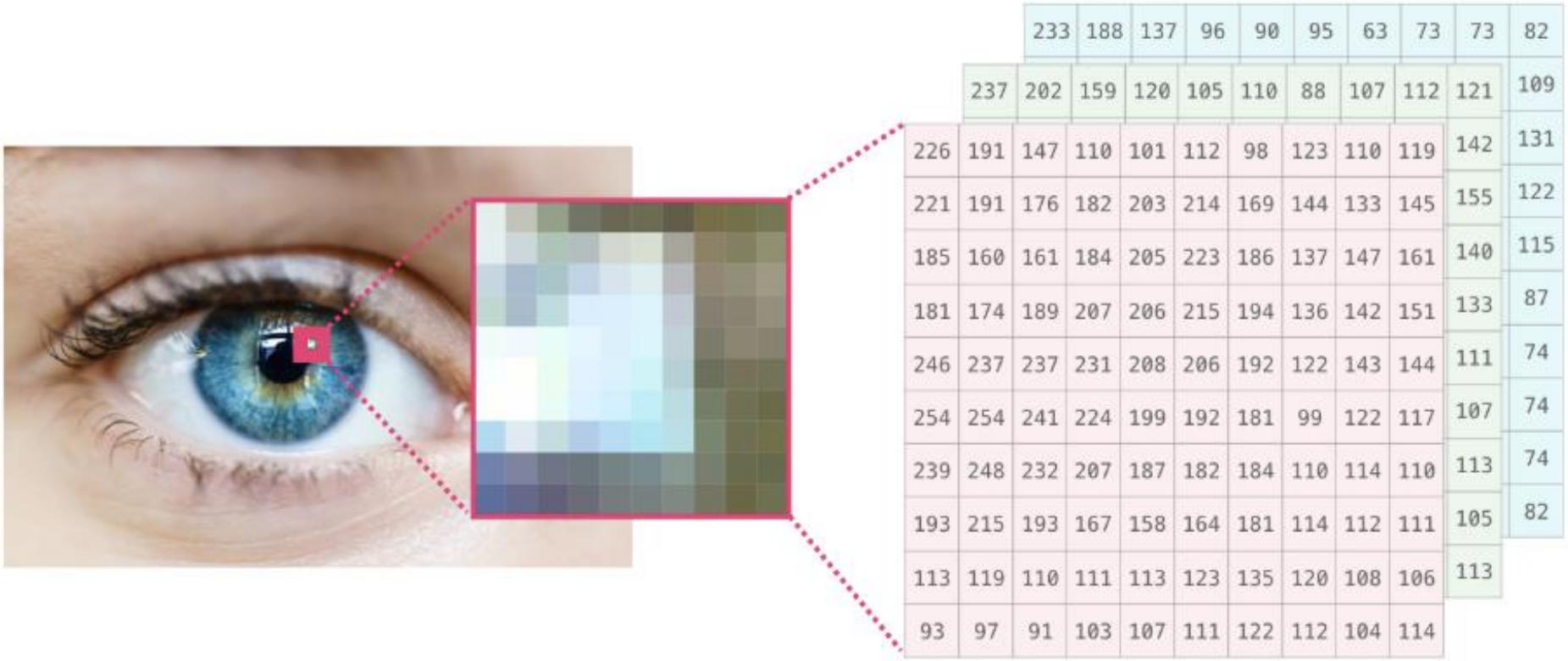
http://jalammar.github.io/visual-numpy/



230	194	147	108	90	98	84	96	91	101
237	206	188	195	207	213	163	123	116	128
210	183	180	205	224	234	188	122	134	147
198	189	201	227	229	232	200	125	127	135
249	241	237	244	232	226	202	116	125	126
251	254	241	239	230	217	196	102	103	99
243	255	240	231	227	214	203	116	95	91
204	231	208	200	207	201	200	121	95	95
144	140	120	115	125	127	143	118	92	91
121	121	108	109	122	121	134	106	86	97

Images

http://jalammar.github.io/visual-numpy/



Language

http://jalammar.github.io/visual-numpy/

Model Vocabulary

#	
0	the
1	of
2	and
…	…
71,289	dolophine

Language

http://jalammar.github.io/visual-numpy/

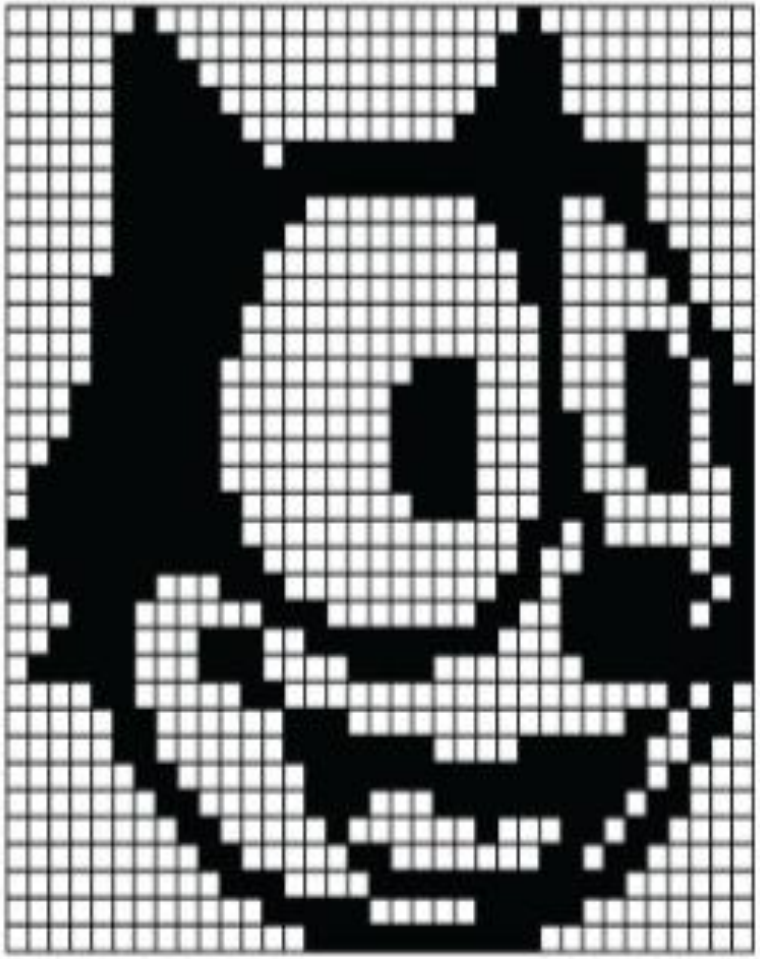
Model Vocabulary

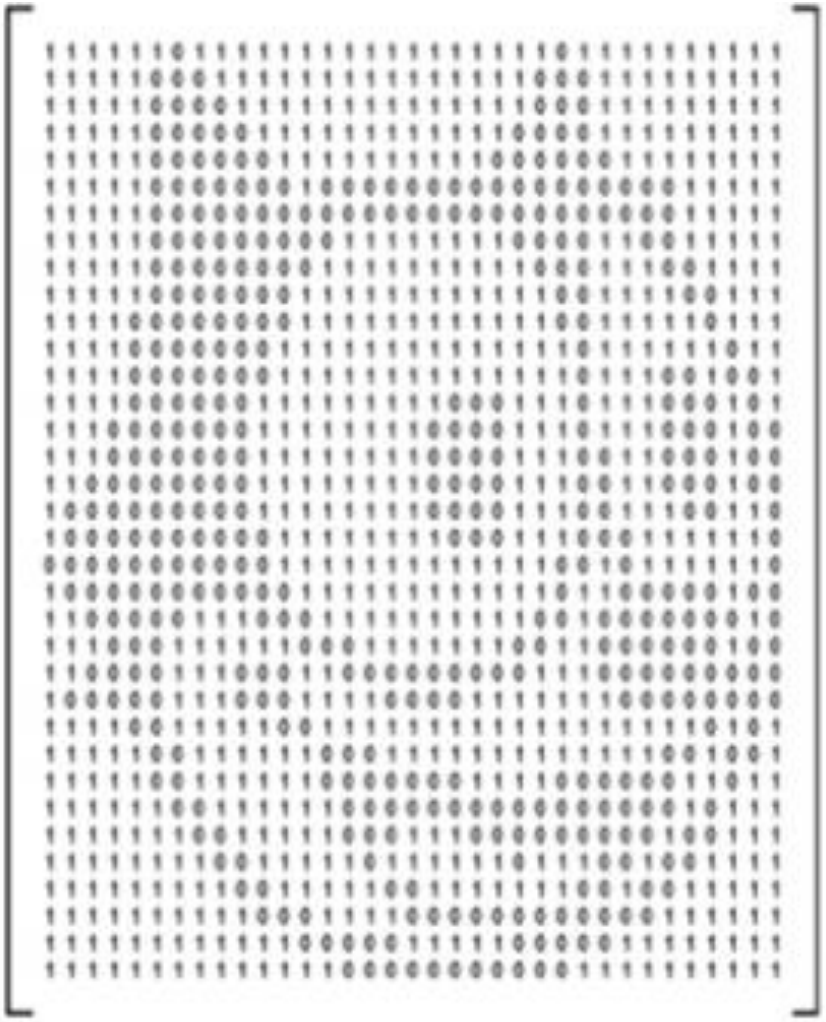
#	
0	the
1	of
2	and
…	…
71,289	dolophine

have	the	bards	who	preceded	me	left	any	theme	unsung
38	0	29104	56	7027	745	225	104	2211	66609

Let’s work with images using NumPy and SciPy

What is an image?





Basic image operations

Geometric transformation

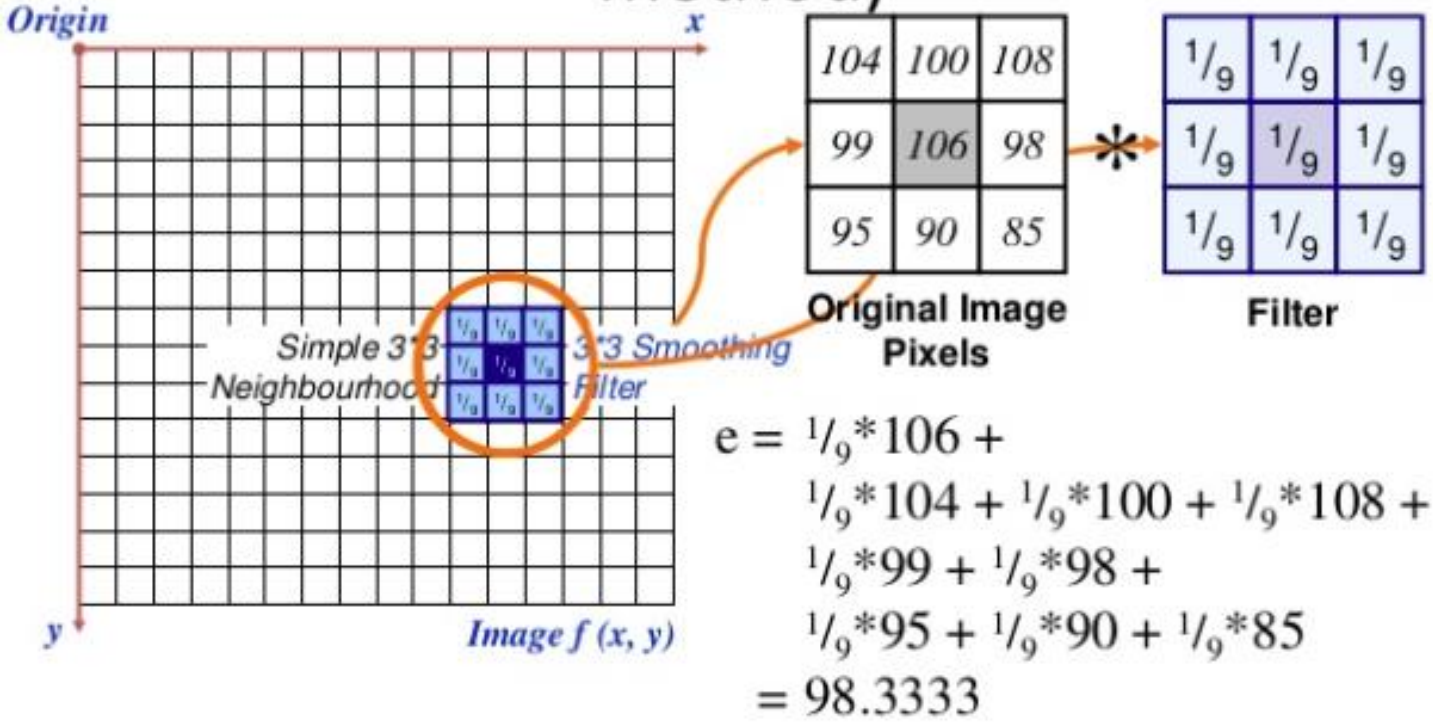
Basic image operations

Filtering through kernel convolutions

Basic image operations

Filtering through kernel convolutions

Smoothing Image(Gaussian blur method)



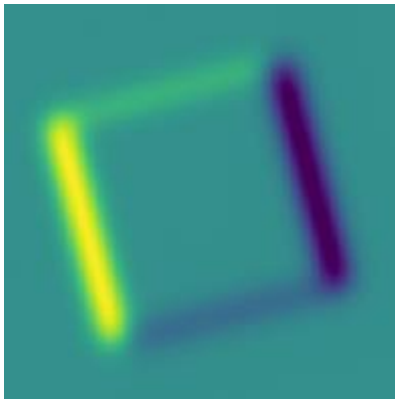
$$e = 1/9*106 + 1/9*104 + 1/9*100 + 1/9*108 + 1/9*99 + 1/9*98 + 1/9*95 + 1/9*90 + 1/9*85 = 98.3333$$

The above is repeated for every pixel in the original image to generate the smoothed image

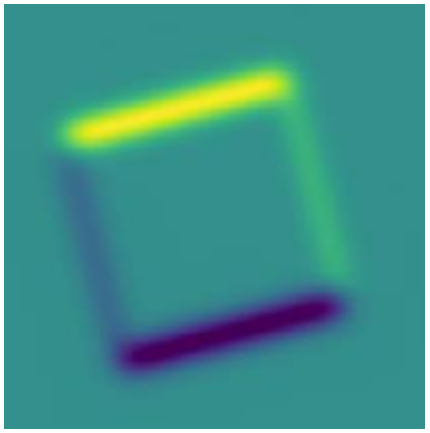
Basic image operations

Sobel filter

-1	0	1
-2	0	2
-1	0	1

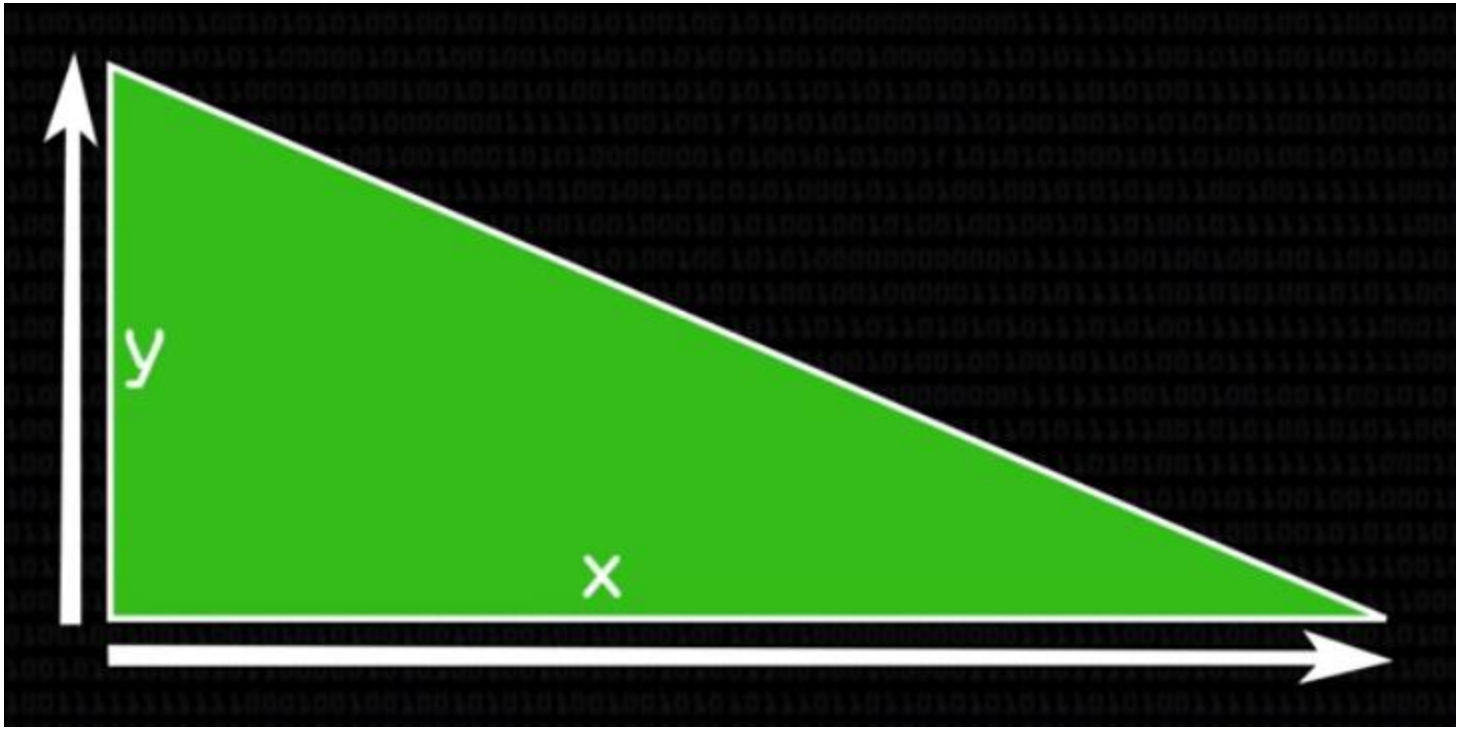


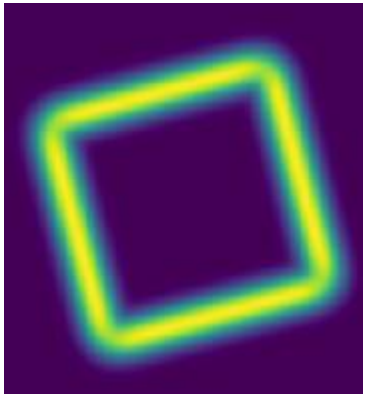
-1	-2	-1
0	0	0
1	2	1



Basic image operations

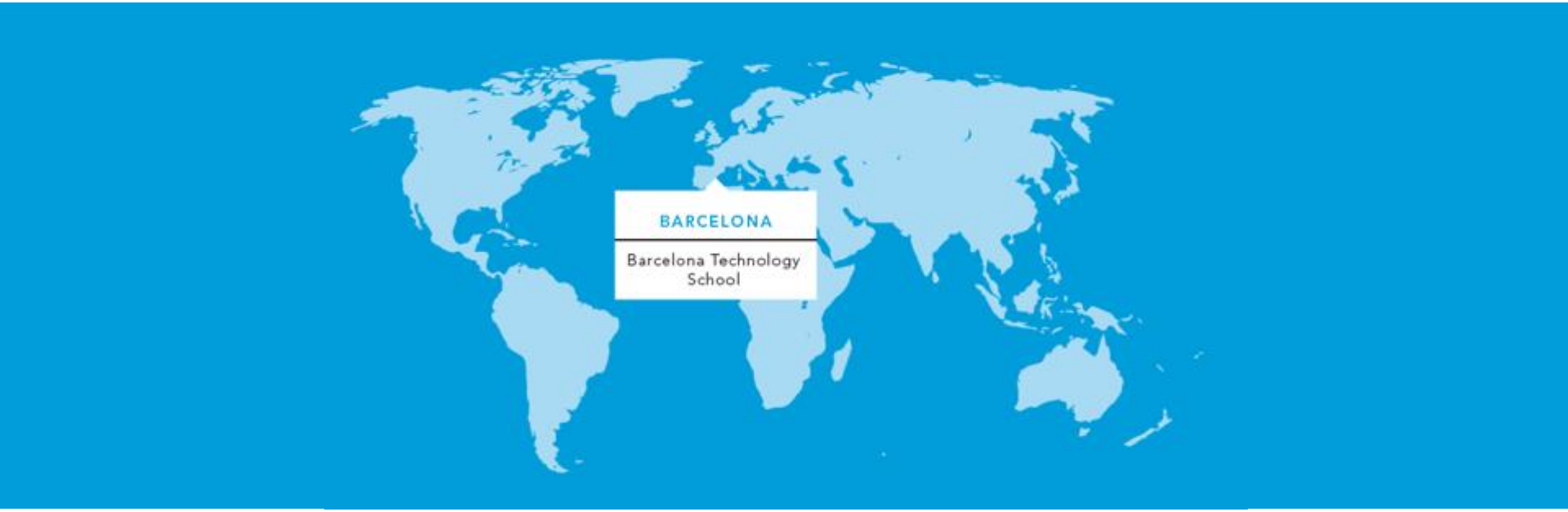
Sobel filter





Let’s get to code!

Go to the notebook


BARCELONA
Barcelona Technology School