

SBWL 1: Data Processing 1 (PI2.0)

Stefan Sobernig

October, 2 2018

Data Science

Data Science

- What is Data Science?
- What problems does Data Science address?
- How do Data Scientists work?
- What tools do Data Scientists use?

What is Data Science?

- "There's a joke running around on Twitter that the definition of a data scientist is 'a data analyst who lives in California'— [Malcolm Chisholm?, @nivertech?]
- "Data Scientist (n.): Person who is better at statistics than any software engineer and better at software engineering than any statistician." [Josh Wills]
- "A data scientist is that **unique blend of skills** that can both unlock the insights of data and tell a fantastic story via the data," — [DJ Patil]
- "Data scientists are involved with gathering data, massaging it into a tractable form, making it tell its story, and presenting that story to others," — [Mike Loukides]
- "taking all aspects of life and turning them into data." [Kenneth Neil Cukier and Viktor Mayer-Schoenberger]
- "Data science teams need people with the skills and curiosity to ask the big questions." [DJ Patil]

What is Data Science?

... bottomline: there is no single definition, but some main recurring terms:

- **about "datafication"**
- **unique blend of skills (teamwork!)**
- **gathering data**
- **massaging data**
- **telling a story about the data**

... plus some recurring mention of common skills...

Datafication

A growing area of private and social life become reflected in computerised data to be turned into "valuable" insights.

- user tracking on the Web
- self-quantification
- cyber-physical ("smart") information systems: smart vehicles, smart stores, etc.
- "smart" marketing

... plus some recurring mention of common skills...

Data Scientists' Skills

	Data analyst	Data scientist
Analyt. skills	Analytical thinking	Excellent in math and statistics
	Apply established analysis methods	Visualisation, new approaches
Tech. skills	Data modelling, databases	Data modelling, databases
	Use of analysis tools	Data mining
	Programming skills of advantage	Algorithm development, method abstraction
Domain knowledge	Detailed domain knowledge	Background domain knowledge
	Project management	Creativity
	Communication skills	Team work

Data Scientists' Skills

"3 sexy skills of data geeks" (Nathan Yau, *Rise of the Data Scientist*, 2009)

- Statistics (data analyses as known to you; see course on "Data Analytics")
- Visualization (plots, visualisation tooling like dashboards, etc.; Data Science Lab)
- **Data munging (scraping, parsing, formatting, and cleaning data) (This course)**

What problems does Data Science address?

Example for data journalism

- focus on politics, economics and sports
- **Who will win the presidency? (USA 2016)**
- **Should Travelers Avoid Flying Airlines That Have Had Crashes in the Past?**
- **A Statistical Analysis of the Work of Bob Ross**
- **2014 World Cup Predictions**

The screenshot shows the FiveThirtyEight website interface. At the top, there are navigation tabs for Politics, Sports, Science & Health, Economics, and Culture. The main content area features a large article titled "World Cup Anxiety Reaches Its Boiling Point For The USMNT" by Michael Caley, accompanied by a photo of soccer players. To the right, there are sections for "THE LATEST" with three article teasers, "INTERACTIVES" with "NFL Predictions" and "MLB Predictions", and "Upcoming games" with a table of matchups and win probabilities.

Game	Probability
Detroit over Carolina	63%
Philadelphia over Arizona	65%
Pittsburgh over Jacksonville	80%
Buffalo over Cincinnati	51%

Dataset for published articles

Data Science as a Process

What does a Data Science process look like?

Example of a "classic" data-driven process: ETL in dataware housing

- ETL refers to a process in database usage and especially in data warehousing that:
- *Extracts* data from homogeneous or heterogeneous data sources
- *Transforms* the data for storing it in proper format or structure for querying and analysis purpose (includes cleansing of deduplications, inconsistencies, dealing with missing data,...)
- *Loads* it into the final target (database, more specifically,

operational data store, data mart, or data warehouse)

See., e.g. *Matteo Golfarelli, Stefano Rizzi. Data Warehouse Design: Modern Principles and Methodologies. McGraw-Hill, 2009.*

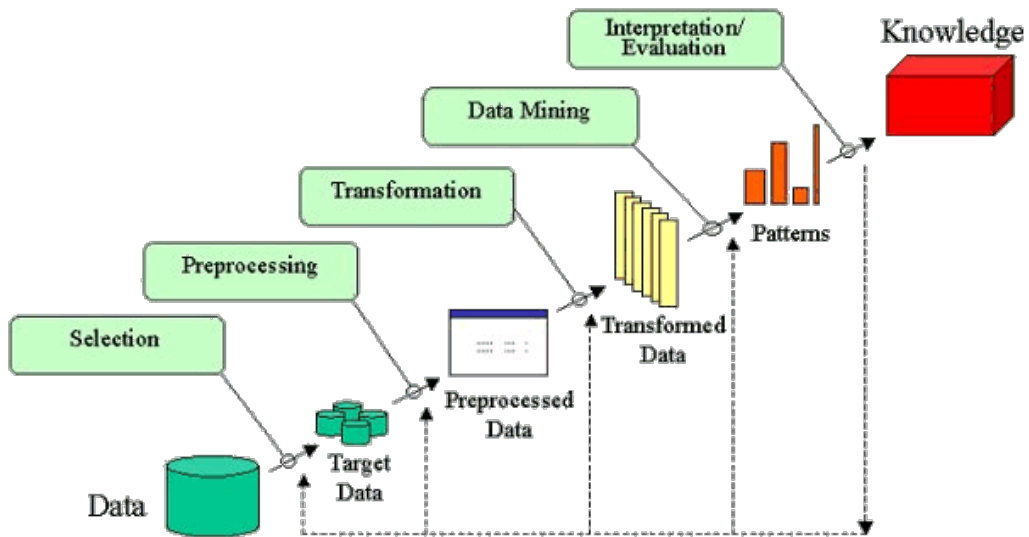
What does a Data Science process look like?

"Classic" views are challenged by datafication:

- The "classic view" typically assumes: **fixed, static processing pipelines** vs. iterative, dynamic pipelines in DS
- Typically assumes **complete/clean data** at the "load" stage vs. messy data in DS
- *Data cleansing* sometimes viewed as a part of a Transform step, sometimes not

What does Data Science Process look like?

"Knowledge Discovery in Databases (KDD)" process (often used in the course of Data Mining)



Source: Howard Hamilton

What does a Data Science Lifecycle look like?

Towards a "Data Science workflow"

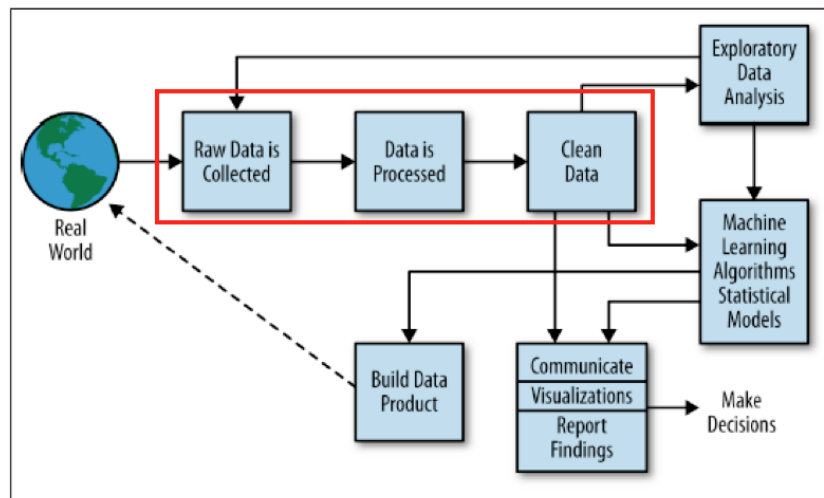
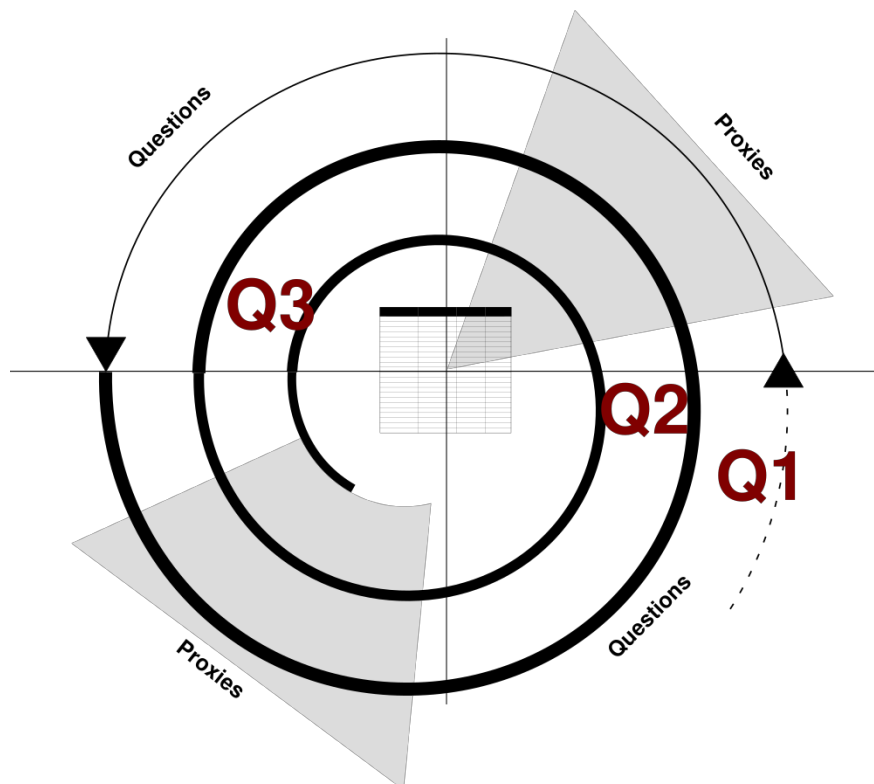


Figure 2-2. The data science process

Iterative Operationalisation



Danyel Fisher & Miriah Meyer. *"Making Data Visual"* (O'Reilly, 2018) (Chapter 2)*

Iterative Operationalisation (cont'd)

- Operationalisation involves searching for **proxies** (proxy tasks, proxy values) for the original question, standing-in for it at the level of the data set.
- Ex. data: a list of movies with ratings (e.g., IMDB) and a list of directors
- Q1: "Who are the best movie directors"?
- **Proxy** for best director: "Having directed many good movies"
- Q2: "What is a good movie"?
- **Proxy**: Good movie: "Having many good IMDB ratings"
- Q3: What is a "good" rating? How many ratings constitute "many" ratings?
- **Proxy**: distributions of rating scores and number of ratings per movie

Challenges in Data Science

WARNING: At each stage, things can go wrong! Any filtering/aggregation may bias the data!

- [...] data scientists [...] **spend a lot more time trying to get data into shape than anyone cares to admit—maybe up to 90% of their time.** Finally, they don't find religion in tools, methods, or academic departments. They are versatile and interdisciplinary*
- Yet far too much handcrafted work — what data scientists call "**data wrangling**," "**data munging**" and "**data janitor work**" — is still required. **Data scientists, according to interviews and expert estimates, spend from 50 percent to 80 percent of their time** mired in this more mundane labor of collecting and preparing unruly digital data, before it can be explored for useful nuggets.

"Data wrangling is a huge — and surprisingly so — part of the job," said Monica Rogati, vice president for data science at Jawbone, whose sensor-filled wristband and software track activity, sleep and food consumption, and suggest dietary and health tips based on the numbers. "It's something that is not appreciated by data civilians. At times, it feels like everything we do."* *New York times*

Challenges in Data Science (cont'd)

SECTIONS **The New York Times** SUBSCRIBE LOG IN

N.S.A. Suspect Is a Hoarder. But a Leaker? Investigators Aren't Sure.

Twitter's Fate: Marc Benioff of Salesforce Addresses Acquisition Talk

PAID POST: CHAUMET Napoleon Owned Jewels by This Legendary Maison

STATE OF THE MAINTENANCE: Un-Silicon Make It as

For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights

By STEVE LOHR AUG. 17, 2014



Monica Rogati, Jawbone's vice president for data science, with Brian Wilt, a senior data scientist. Peter DaSilva for The New York Times

Technology revolutions come in measured, sometimes foot-dragging steps. The lab science and marketing enthusiasm tend to underestimate the bottlenecks to progress that must be overcome with hard work and practical engineering.

The Data Science Lifecycle: your own experiences?

Which difficulties have you already experienced when working with data?

1. ... ever had problems loading/ importing a file someone sent to you because of an unknown file format?
2. ... ever encountered something like this: "K♦snudl"?
3. ... ever encountered blanks in your data?
4. ... ever saw an observation (an insight, a trend) disappear when combining from different data sets (a.k.a. "Simpson's paradox")
5. ... **more on that in the next lectures!**

Data Science Lifecycle: Summary

Again, not a single definition, but some recurring terms:

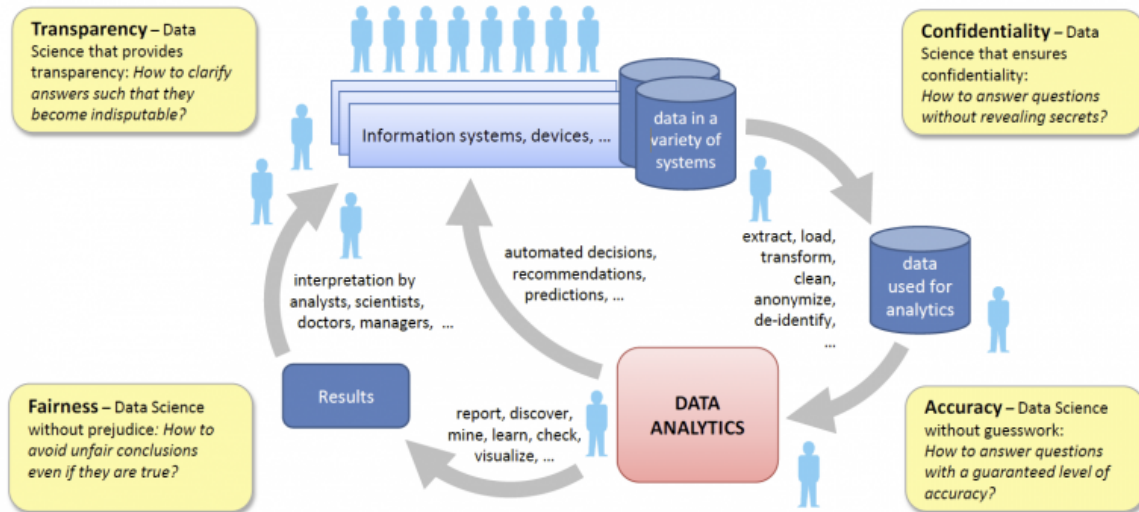
1. **find and collect all relevant data**
2. **identify issues & problems within the data**
3. **organise / transform / merge data**
4. systematically operationalise questions about the data: proxies
5. select a visualisation, a statistical technique, or a machine-learning technique as an outcome of operationalisation
6. provide interpretations and limitations of the results
7. communicate results

Data Science Ethics

Ethics in Data Science: FACT

- **Fairness** : How to avoid unfair conclusions even if they are true?
- **Accuracy** : How to answer questions with a guaranteed level of accuracy?
- **Confidentiality** : How to answer questions without revealing secrets?
- **Transparency** : How to clarify answers such that they become indisputable?

Ethics in Data Science: FACT (cont'd)



Source <http://www.responsible-datascience.org/>

Data Science Lifecycle: Summary

NOTE:

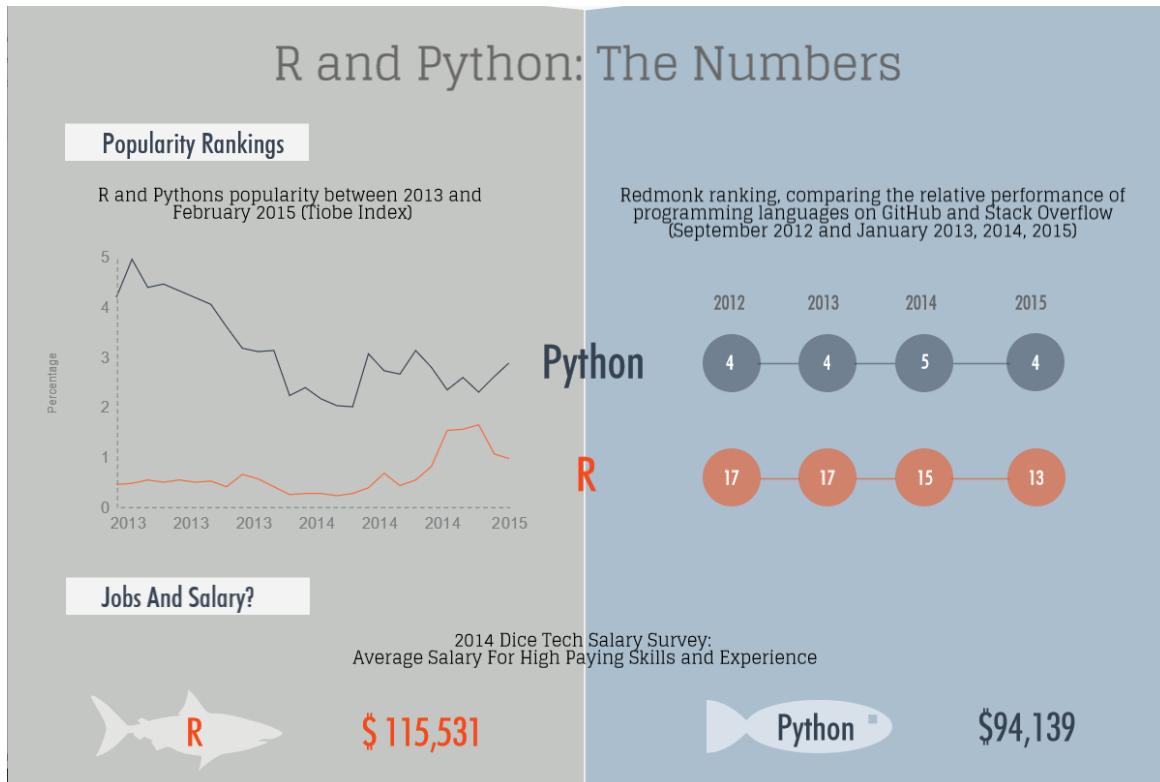
- Typically, Data Science is not a one-shot process, but an (iterative) lifecycle.
- Not ad hoc, but short-lived than building than classic processes: ETL, data mining.
- Typically, you need to revisit/ adjust your process, either for improving it or for maintenance (sources changing, source formats changing, etc.)
- Mind FACT in Data Science projects

Notice.

These steps may take **80% of the work** or more -> This is the focus of our course "**Data Processing I**"
!!!

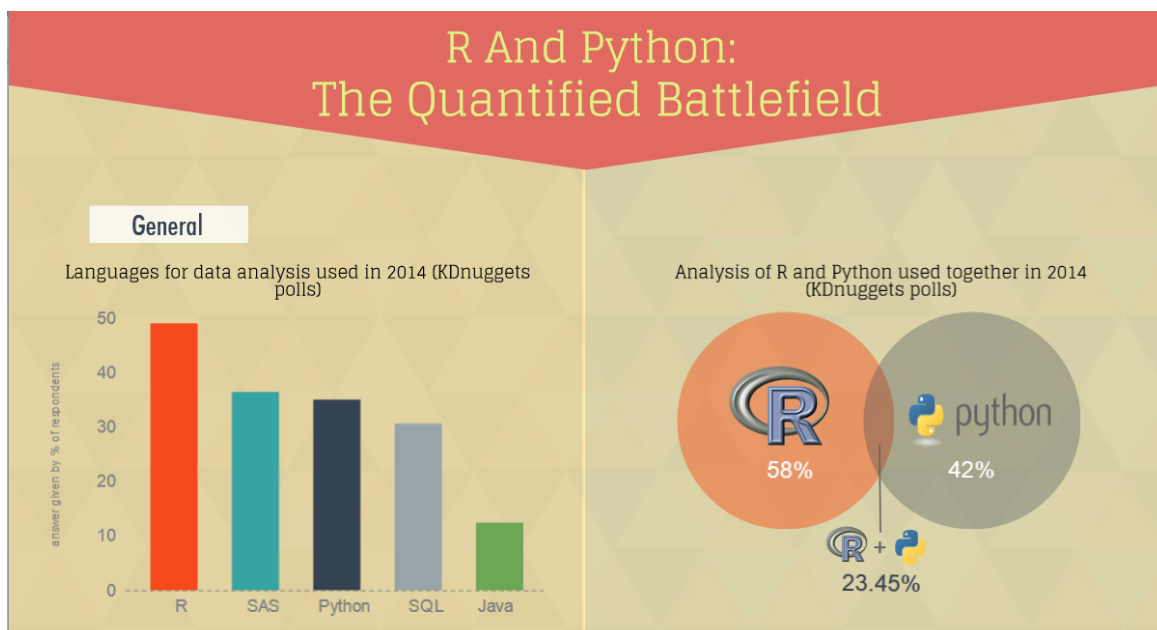
Data Science Tools

Data Science Tools: Python and R





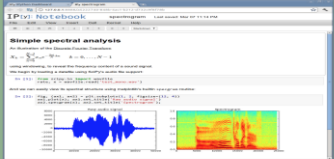
Source <https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis>

Python and R



Source <https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis>

Python and R

Graphical Capabilities	+	IPython Notebook
<p>A picture says more than a thousand words</p> <p>Visualized data can be understood more efficiently and effectively than the raw numbers alone.</p> <p>R + visualization = perfect match </p> <p>ggplot2 To make pretty graphs, including the opportunity to use grammar of graphics to create layered, customizable plots</p> <p>lattice To easily display multivariate relationships</p> <p>rCharts To create, customize and publish interactive javascript visualizations from R</p> <p>googleVis To use Google Chart tools to visualize data in R</p> <p>ggvis To implement interactive grammar of graphics, while rendering in a web browser</p> <p>e.g.: Visualizing Facebook friends with R</p> 		<p>Bundle your analysis in one file</p> <p>The IPython Notebook makes it easier to work with Python and data.</p> <p>Simplify your workflow when working with data in Python</p> <p>It's a combination of:</p> <p>Interactive python exploration, prewritten programs, text, and equations for documentation in one environment</p> <p>Share notebooks with colleagues without having them install anything.</p> <p>The IPython notebook drastically reduces the overhead of organizing code, output, and notes files, which allows to spend more time doing real work.</p> 

Source <https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis>

Why Python and R

The Python vs R debate confines you to one programming language. You should look beyond it and embrace both tools for their respective strengths. Using more tools will only make you better as a data scientist. [TheNextWeb]

- Data Processing 1 (SBWL 1): Python
- Data Analytics (SBWL 2): R
- Data Processing (SBWL 3): Python

Python & Jupyter



Outline

- Why Python?
- Python installation
- Working with Python
- Working with Jupyter
- Brief Python3 tutorial

Why Python?

Python is a dynamic general-purpose language with which one can archive fast results in only a few lines of code.

- functional and object oriented programming language
- dynamic typing
- many data science libraries
- large and lively community

Companies

- Youtube, DropBox, Google, Quora, Reddit, Yahoo Maps

See also a [verified list of companies using Python](#)

Versions 2.7 vs. 3.x

Python is currently available in two versions: Version 2.x and 3.x.

We are using Python 3 in this course

- better support for unicode
- Python 3.x is the present and future of the language
- see [Python 2 vs 3](#) for a discussion

Examples:

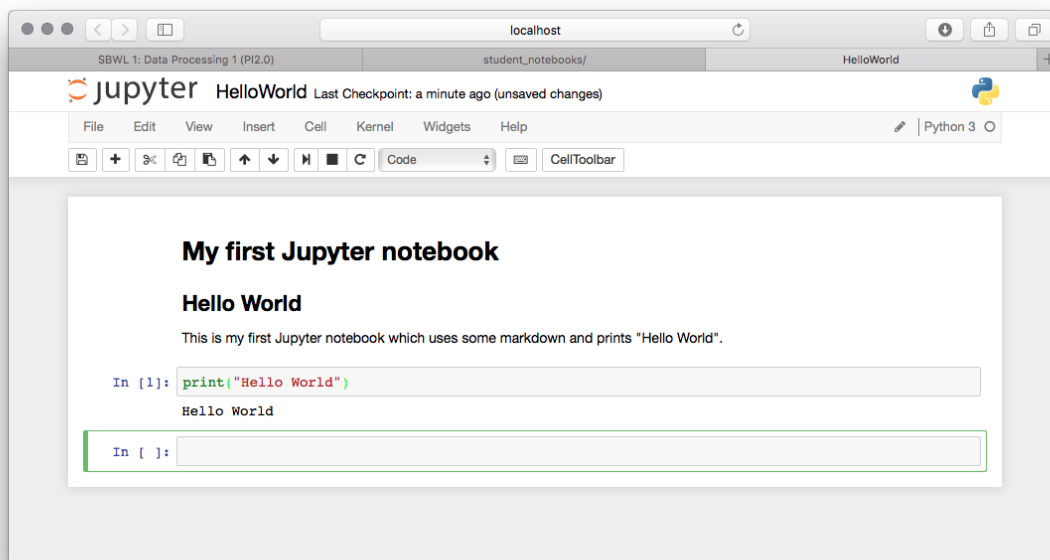
Python 2

- print 5
- `3 / 2 --> 1`
- `3 // 2 --> 1`
- `3/2.0 -->1.5`

Python 3

- print(5)
- `3 / 2 --> 1.5`
- `3 // 2 --> 1`

Jupyter Notebook



Brief Python3 Tutorial

Jupyter Notebook Version

The following slides are also available as Jupyter notebook *python3-intro.ipynb*.

A useful helper: The print operator

```
print('test')
```

Basic Data Types

Basic data types are the essential building blocks for handling information in Python

- Strings
- Integers
- Floats
- Boolean

Strings

Any text between two matching quotes (either single ' ' or double quote " ")

Examples

```
'data'  
"science"  
'I study at WU Vienna'
```

Exercise.

Create some strings and play with the different quotes

see also Chapter 3.1.2 in the Python tutorial ([en](#) , [de](#))

Integers

Integers are whole numbers

```
Terminal> python3 -c 'print( type( 1 ) )'  
<class 'int'>
```

Some examples:

```
1  
0  
-5
```

Floats

Floats are decimal number types.

```
Terminal> python3 -c 'print( type( 2.2 ) )'  
<class 'float'>
```

Some examples:

```
1.0  
15.4
```

Numbers with leading zero

Python does not support numbers with a leading zero

```
0034
```

```
Terminal> python3 -c '0034'  
SyntaxError: invalid token
```

Operations for Numbers: Addition

- "+" Addition

```
5+4
```

```
Terminal> python3 -c 'print(5+4)'  
9
```

Operations for Numbers: Subtraction

- "-" Subtraction

```
10-34
```

```
Terminal> python3 -c 'print( 10-34 )'  
-24
```

Operations for Numbers: Multiplication

- "*" Multiplication

```
5*4
```

```
Terminal> python3 -c 'print(5*4)'  
20
```

```
2.5 *3
```

```
Terminal> python3 -c 'print( 2.5*3 )'  
7.5
```

Operations for Numbers: Division

Python 3

- "/" (floor division)
- "/" (true division)

```
4/8
```

```
Terminal> python3 -c 'print(4/8)'  
0.5
```

see also Chapter 3.1.1 in the Python tutorial ([en](#) , [de](#))

Strings vs. Integers

Question.

The "==" operator compares if two values are equal. What happens if we execute the following command?

```
5=="5"
```

```
Terminal> python3 -c 'print( 5=="5" )'
False
```

Notice.

If a number is entered within quotes, the value is processed as string.

Float vs. Integers

Try the following

Question.

The "==" operator compares if two values are equal. What happens if we execute the following command?

```
5==5.5
```

```
Terminal> python3 -c 'print( 5==5.5 )'
False
```

Booleans

A boolean data type has only two possible values: True or False

- named after **George Boole**
- truth value of logic and boolean algebra
- used to test conditions and to control the program flow

```
Terminal> python3 -c 'print( type( True ) )'
<class 'bool'>
```

Data Containers

- Data containers can hold multiple data points.
- Data containers are **data types** again

Python provides the following containers:

- Variables
- lists
- dictionaries

Variables

Variables are a means to store and reference data

- container that holds information
- sole purpose is to label and store data in memory

Python does not require type declarations (unlike Java), defining variables is thus as simple as:

```
VARIABLE_NAME = ASSIGNMENT
```

Number assignments

For instance. Assigning the value of 1 to variable *a*

```
a = 1
```

String assignments

For instance. Assigning the value of "Data Science" to variable *title*

```
title = "Data Science"
```

Operations with variables

One can also combine operations with variables

```
x = 5
y = 10
c = x*y
print(c)
```

```
Terminal> python3 -c 'x=5;y=10;c=x*y; print(c)'
50
```

Operations with variables


```
a = 'Data'
b = 'Science'
print(a+b)
```

```
Terminal> python3 -c 'a = "Data"; b = "Science"; print( a+b )'
DataScience
```


Lists

Do you remember?

Data structures example: List



- Array or list *list := [a₁, a₂, a₃, a₄, ... a_n]*
- typical use cases
 - grocery list (butter, milk, *G := ["milk", „butter“, „soup“]*
 - vectors *N := [11, 12, 13, 14, 15]*
 - sequence of numbers
 - *many more*
 - *nested arrays*
 - *matrixes*
 - *tables, cubes*

8


Lists

A list is a group of items

You can create a list in Python by placing the items in square brackets ([]) and separating the items with a comma.

```
[ item1, item2, item3, ..., itemN ]
```

```
Terminal> python3 -c 'print( type( [] ) )'  
<class 'list'>
```

Lists: Example

```
[ 'Milk', 'Eggs', 'Lettuce' ]  
#or  
[ 12.5, 8.0, 61.3, 87.5 ]
```

Lets store the list in a variable so that we can reuse it later in the code

```
list = [ 12.5, 8.0, 61.3, 87.5 ]  
print(list)  
[ 12.5, 8.0, 61.3, 87.5 ]
```

see also Chapter 3.4 and 5 in the Python tutorial ([en](#) , [de](#))

Lists Concatenation

`./src/listex.py`

```
a1=['a','b','c']  
a2=['d','e']  
a3=a1+a2  
print(a3)
```

```
Terminal> python3 ./src/listex.py  
['a', 'b', 'c', 'd', 'e']
```


Notice.

"_" , "*" , "/" are not allowed as operations for lists

Iterating over lists

Do you remember?

Loops/Iterations (control flow statement)




- loops allow code to be executed repeatedly

loops

```
for item in list:
    //do something with item
```

- Typical use cases
 - iterate over items in a list
 - repeat code x times (e.g., login at ATM with failure)
 - repeat code every x minutes (e.g., periodically check emails)

14


Iterating over lists

`./src/listex2.py`

```
list=[11,22,33,44,55]
for item in list:
    print(item)
```

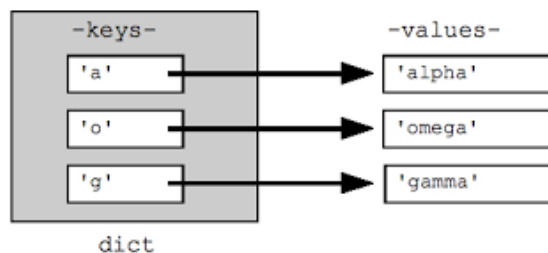
```
Terminal> python3 ./src/listex2.py
11
22
33
44
55
```

Dictionaries

A Python dictionary is a more complex data container than a variable or a list.

- key: the *word* you lookup
- value: result for the lookup

```
{ key1: value, key2: values }
```



```
Terminal> python3 -c 'print( type( {} ) )'  
<class 'dict'>
```

Dictionaries: Example

`./src/dict.py`

```
wordCounts={ 'Data':10, 'Science': 1, 'Course':5 }  
print(wordCounts)  
  
#access key-value  
print( wordCounts['Data'] )
```

```
Terminal> python3 ./src/dict.py  
{'Data': 10, 'Science': 1, 'Course': 5}  
10
```

Dictionaries: Values

The values of a dictionary itself can be:

- data (e.g. Integers, Strings, Booleans)
- lists
- dictionaries

Dictionaries: Values

`./src/dict2.py`

```
course={ 'title': 'DataProcessing1 (WS17)',  
         'authors':['A. Polleres', 'J. Umbrich'],  
         'wordCounts': {'Data':10, 'Science':10}  
       }  
  
value=course['wordCounts']  
print(value)  
print( type(value) )
```

```
Terminal> python3 ./src/dict2.py  
{'Data': 10, 'Science': 10}  
<class 'dict'>
```

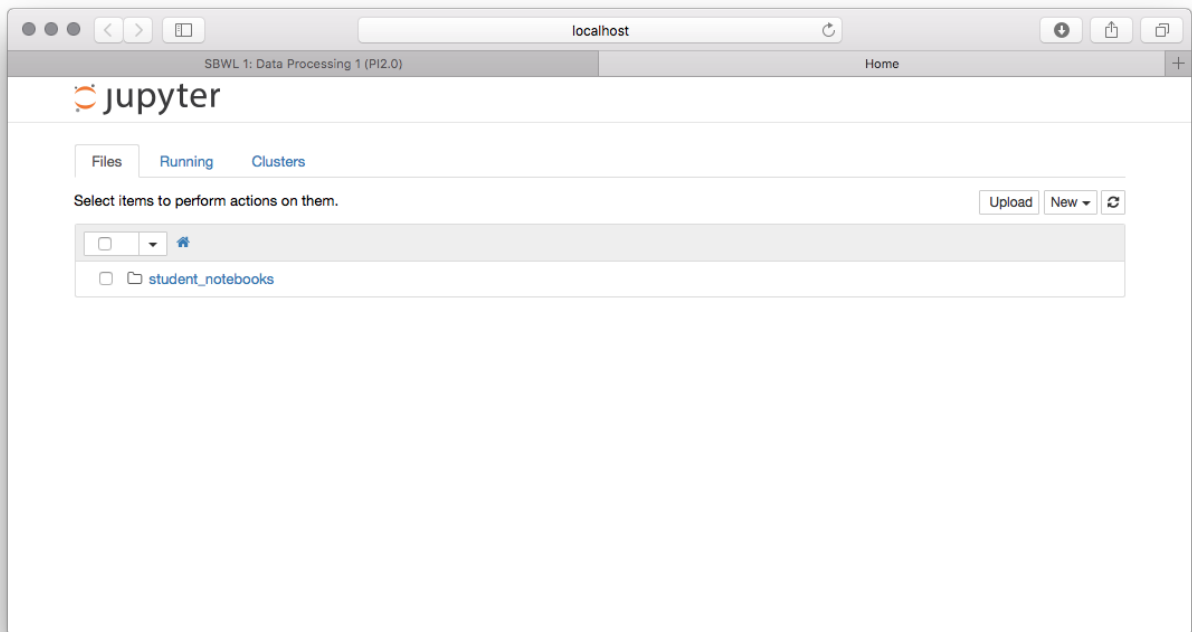
Jupyter

Jupyter

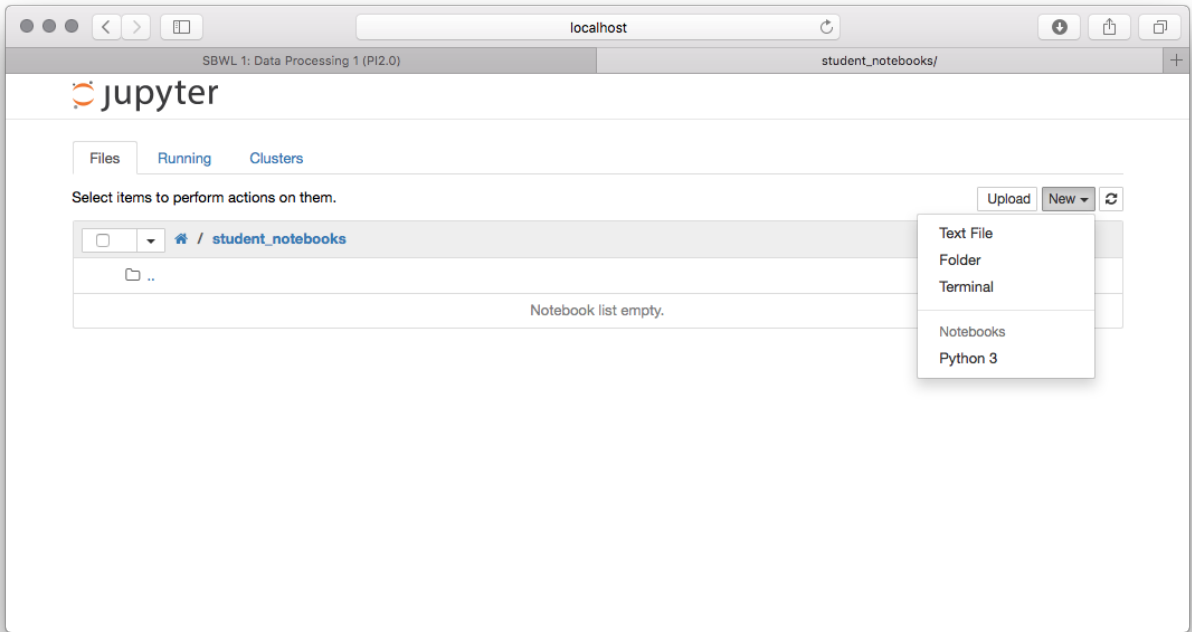
The Jupyter Notebook is a web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more. [[Jupyter.org](https://jupyter.org)]



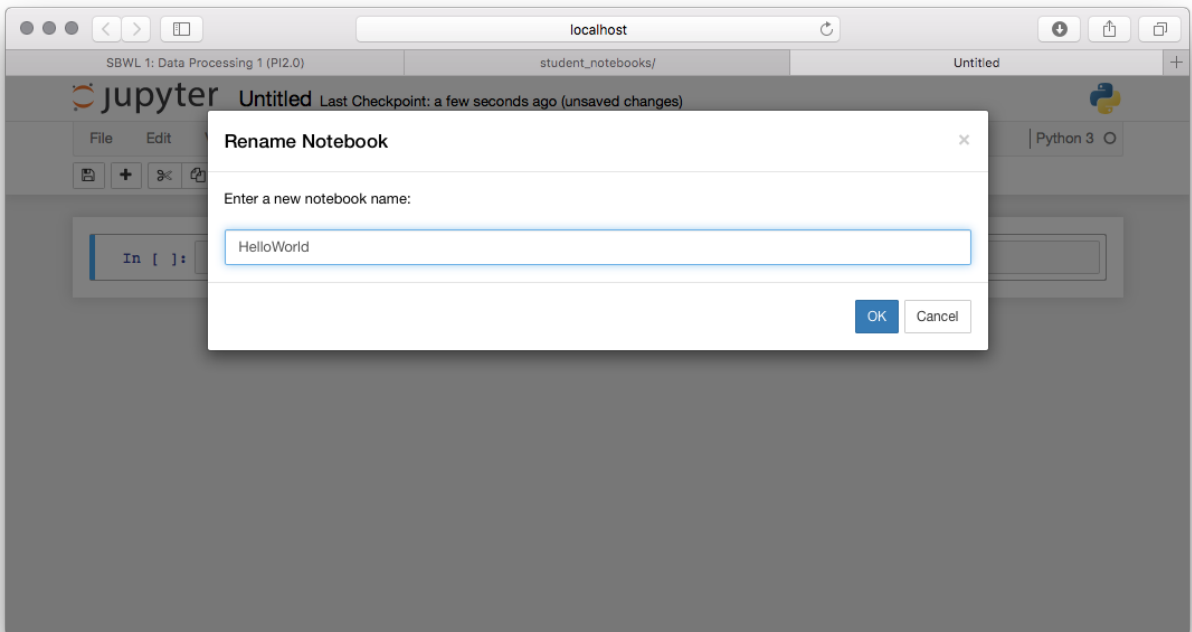
Jupyter UI



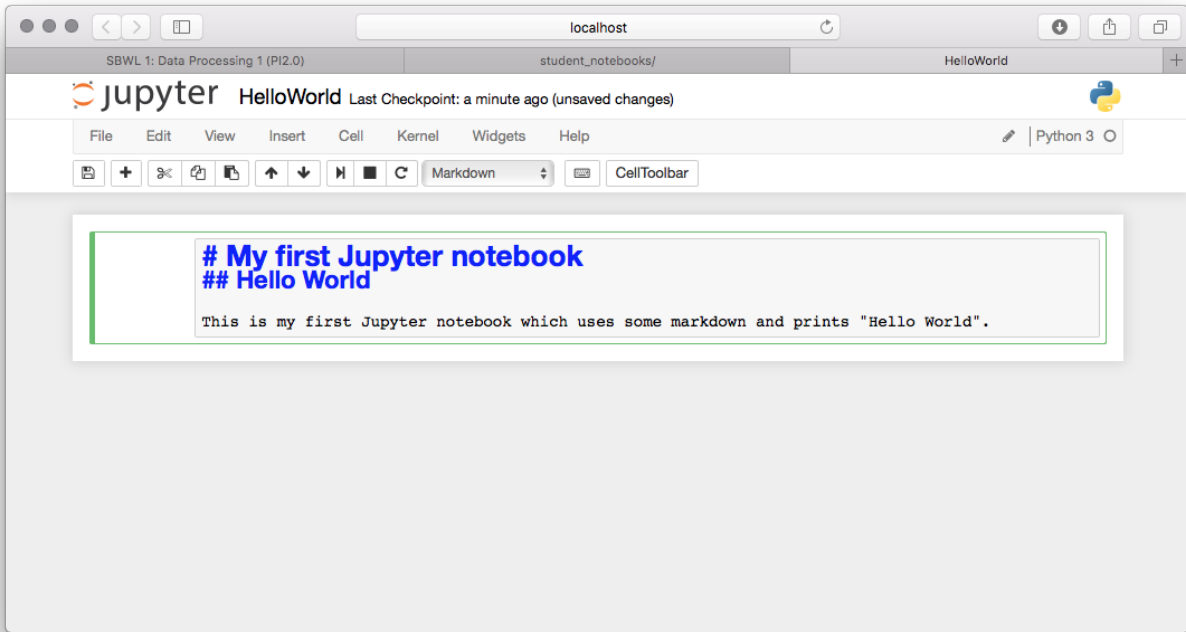
Jupyter: Create a new Notebook



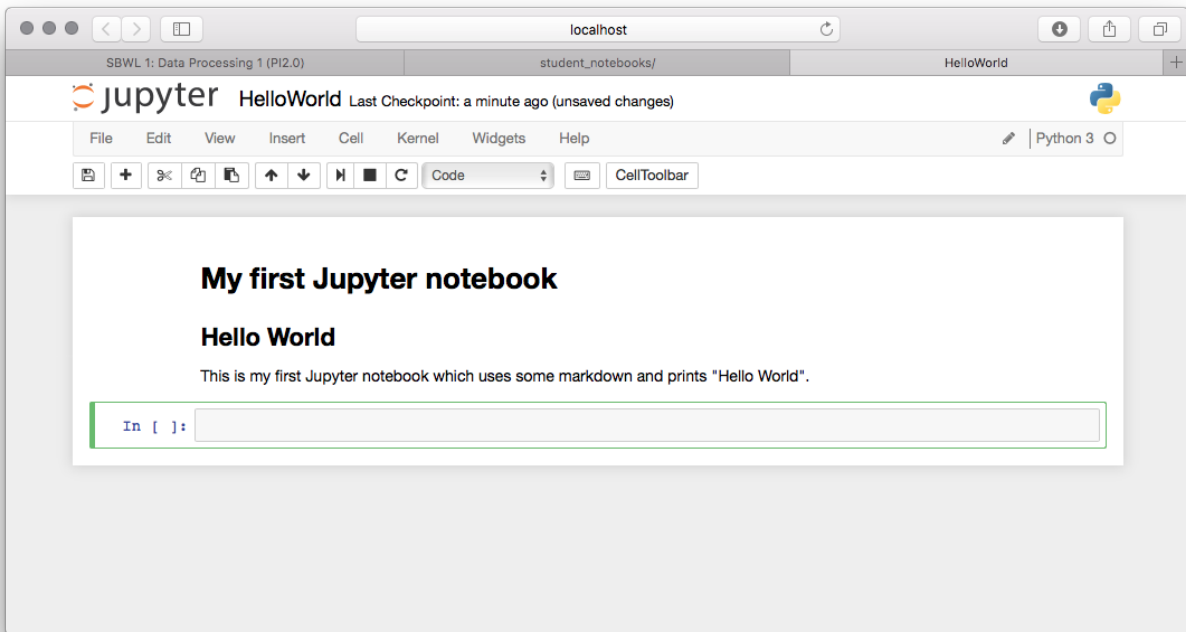
Jupyter: Set a title



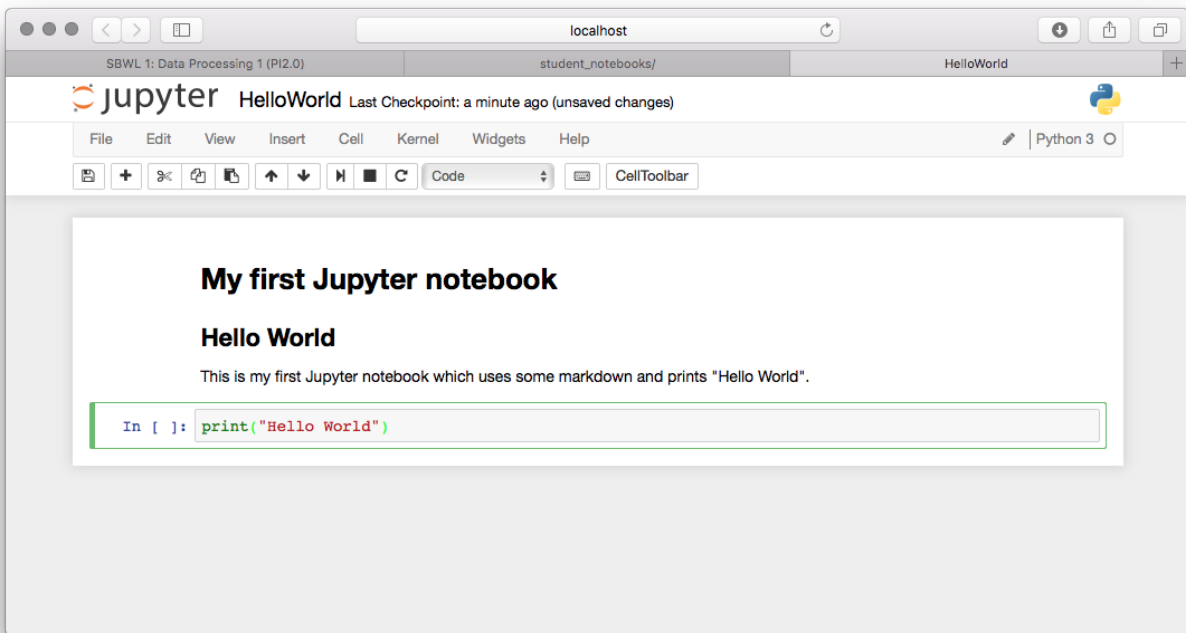
Jupyter: Markdown Cells



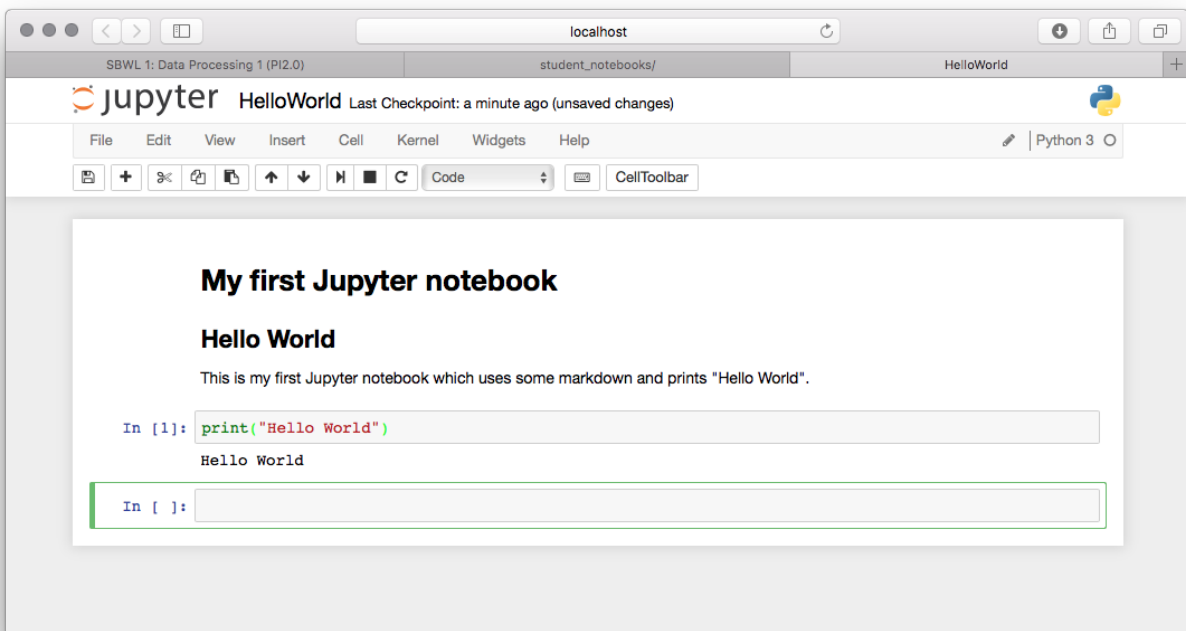
Jupyter: Markdown Cells



Jupyter: Code Cells



Jupyter: Running Code



Markdown?

Markdown is a text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML).[\[Official homepage\]](#)

See a good introduction at help.github.com

Markdown Cheatsheet

Headers and text formating

```
# The largest heading
## The second largest heading
##### The smallest heading

**This is bold text**
*This text is italicized*
> This is a quote
```

Markdown Cheatsheet

Lists

```
- George Washington
- John Adams
- Thomas Jefferson
```

```
1. James Madison
2. James Monroe
3. John Quincy Adams
```

Lets Try

Further Reading material

- [Learning Python](#) by Mark Lutz and David Ascher (O'Reilly)
- [Official Python 3 Tutorial \(english\)](#)
- [Official Python 3 Tutorial \(german\)](#)
- [A gallery of interesting IPython Notebooks](#)