Python tutorial:
Installation & basic syntax

Research Group Meeting
07/25/2017
Mustafa Dogan

Tutorial outline

- Software installation
- Basic syntax & arithmetic operations (+,-,*,/)
- Strings
- List, set, dictionary, array, and data frame
- If statements
- Loops: for, while, enumerate
- Load and save data (`csv`, `text`)
- Plotting & data visualization

Python programming language

- Widely used high-level programming language for general-purpose programming
- a design philosophy which emphasizes code readability
- a syntax which allows programmers to express concepts in fewer lines of code
- constructs intended to enable writing clear programs

The core philosophy of the language `import this`

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Flat is better than nested.
- Sparse is better than dense.
- Readability counts.
- Special cases aren't special enough to break the rules.
- Although practicality beats purity.
- Errors should never pass silently.
- Unless explicitly silenced.
- In the face of ambiguity, refuse the temptation to guess.
- There should be one—and preferably only one—obvious way to do it.
- Although that way may not be obvious at first unless you're Dutch.
- Now is better than never.
- Although never is often better than right now.
- If the implementation is hard to explain, it's a bad idea.
- If the implementation is easy to explain, it may be a good idea.
- Namespaces are one honking great idea—let's do more of those!

Downloading Software

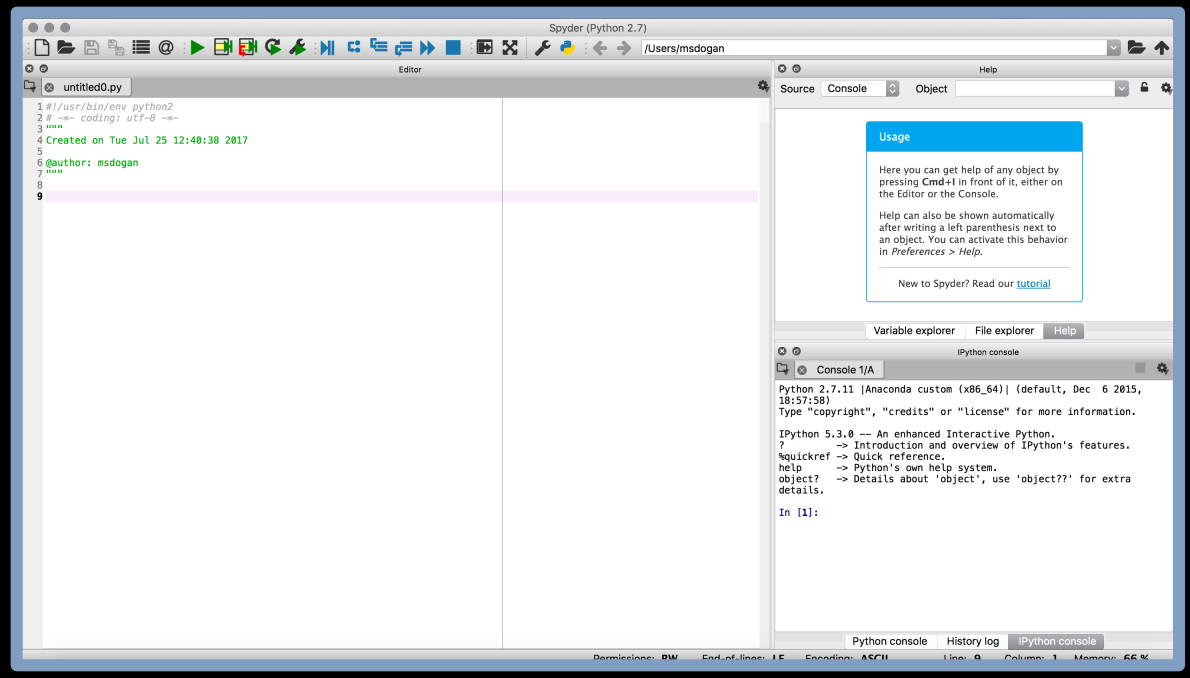
- Anaconda distribution includes almost all packages you will need to run Python
- Link to software (or just google Anaconda)

https://www.continuum.io/downloads

(Python 3+ is recommended)

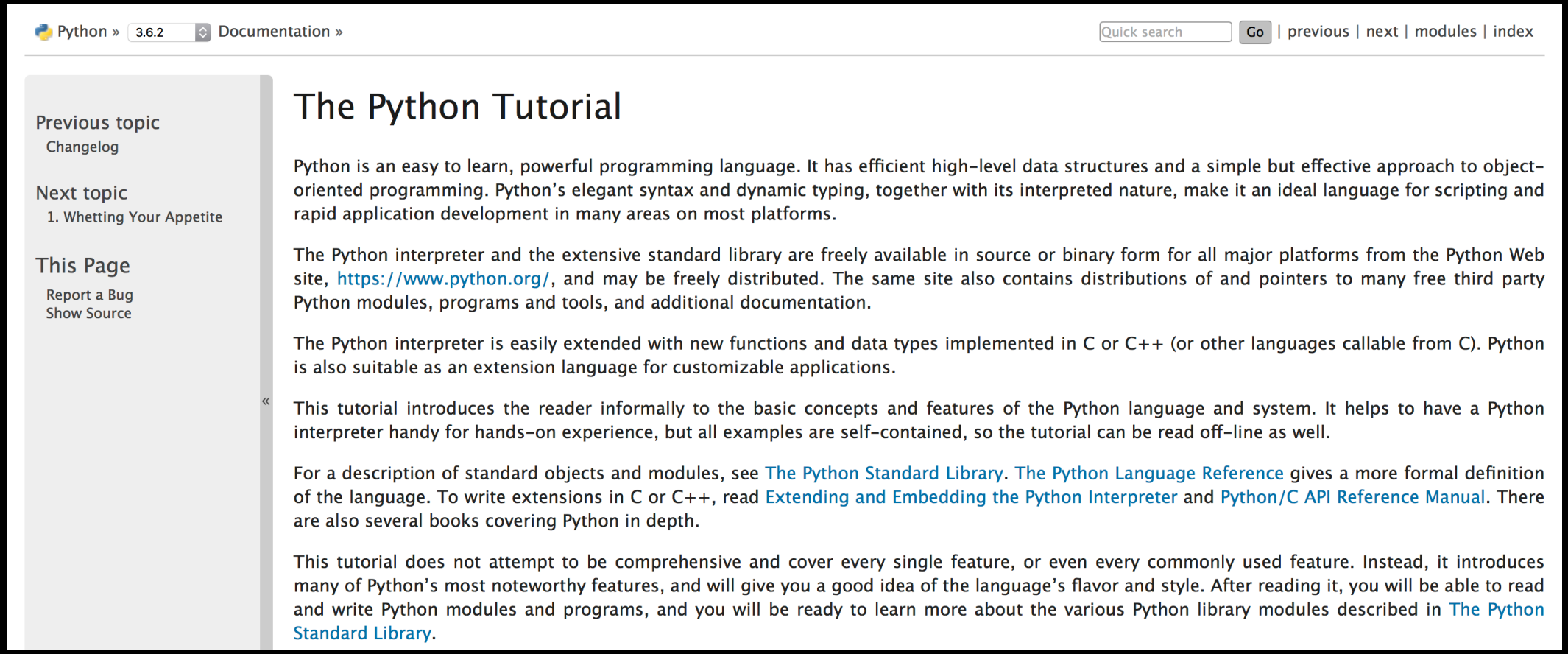
Using Python with Spyder IDE

- After installing Anaconda distribution open spyder
- Spyder is an open source cross-platform integrated development environment (IDE) for scientific programming in the Python language
- Similar to Matlab, there is script editor and console



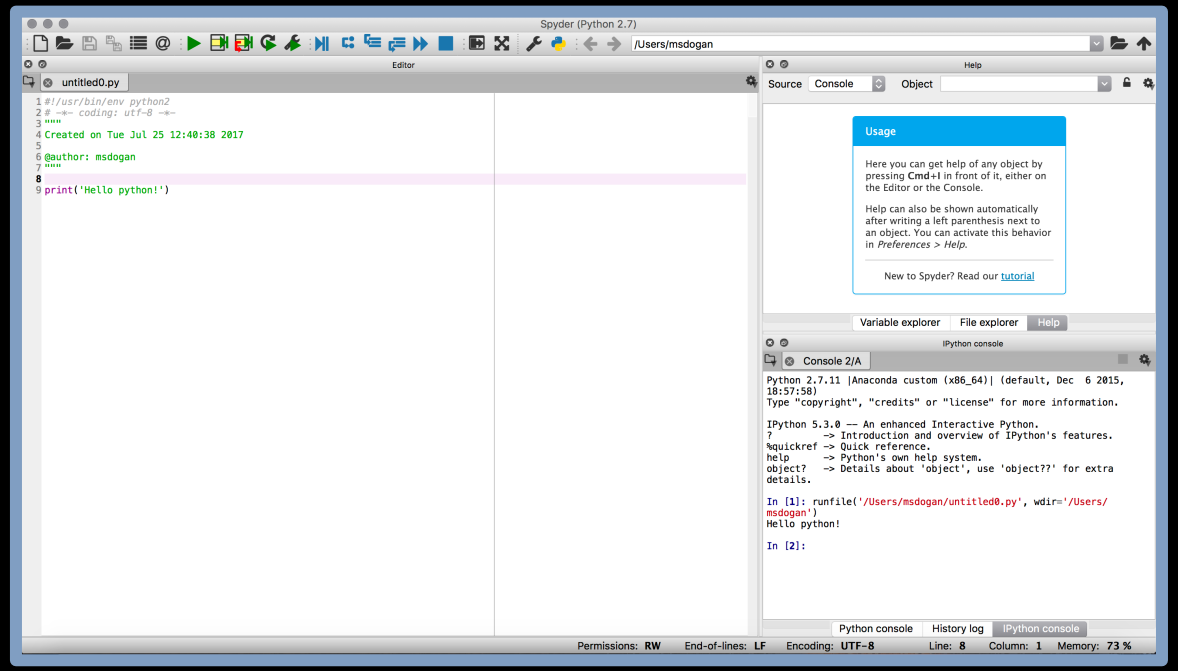
Official tutorial website

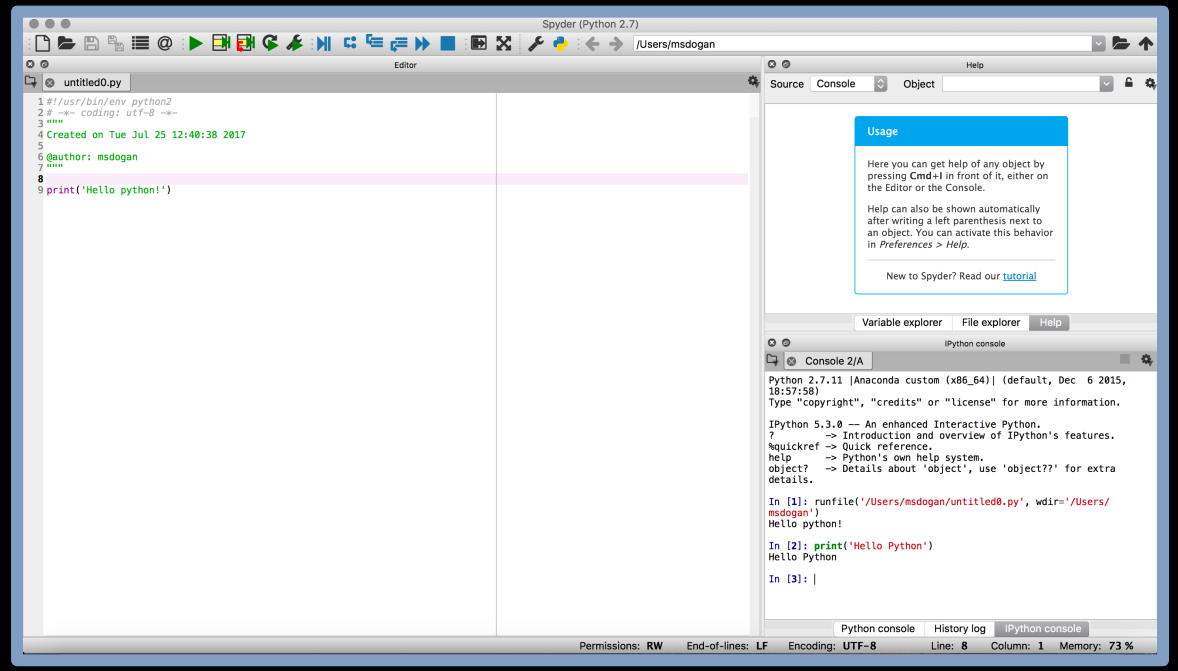
https://docs.python.org/3/tutorial/index.html



First Command: Hello Python!

- Type `print('Hello Python')` to script editor and run (F5)
- Type `print('Hello Python')` to console and hit 'enter'

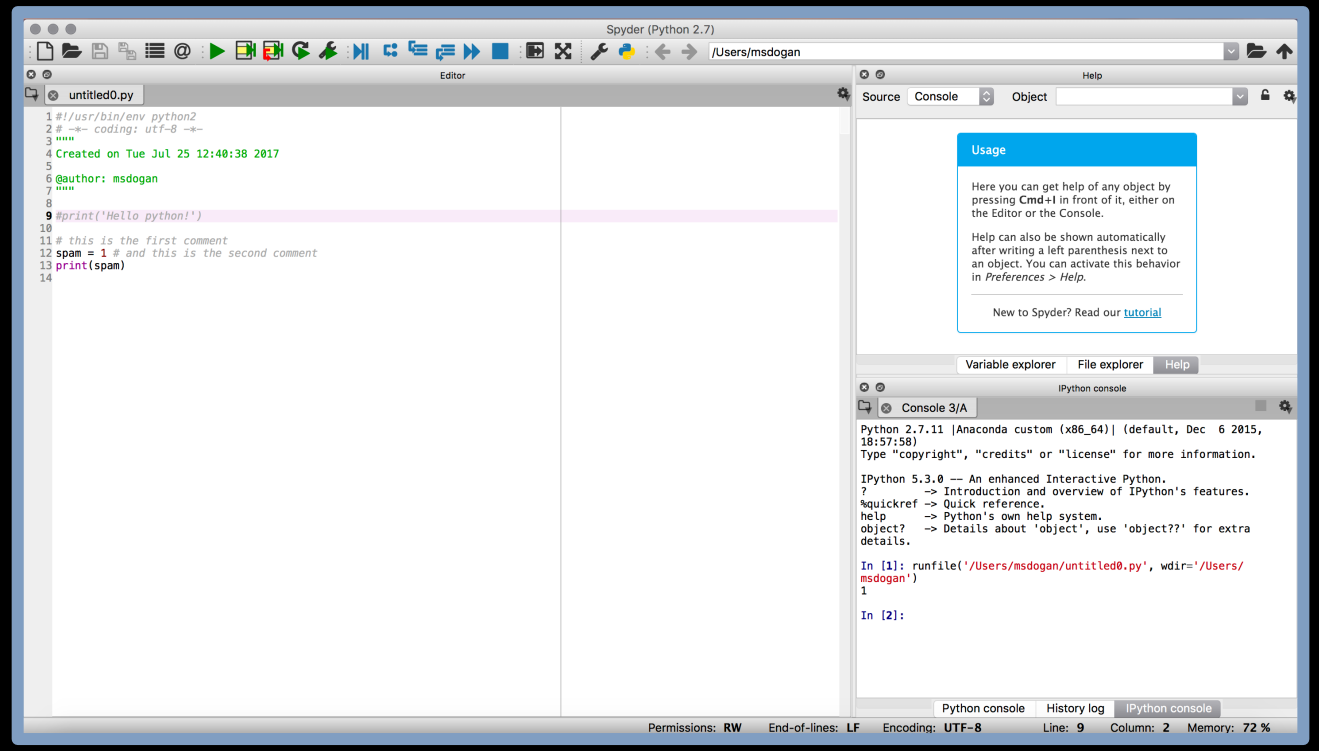




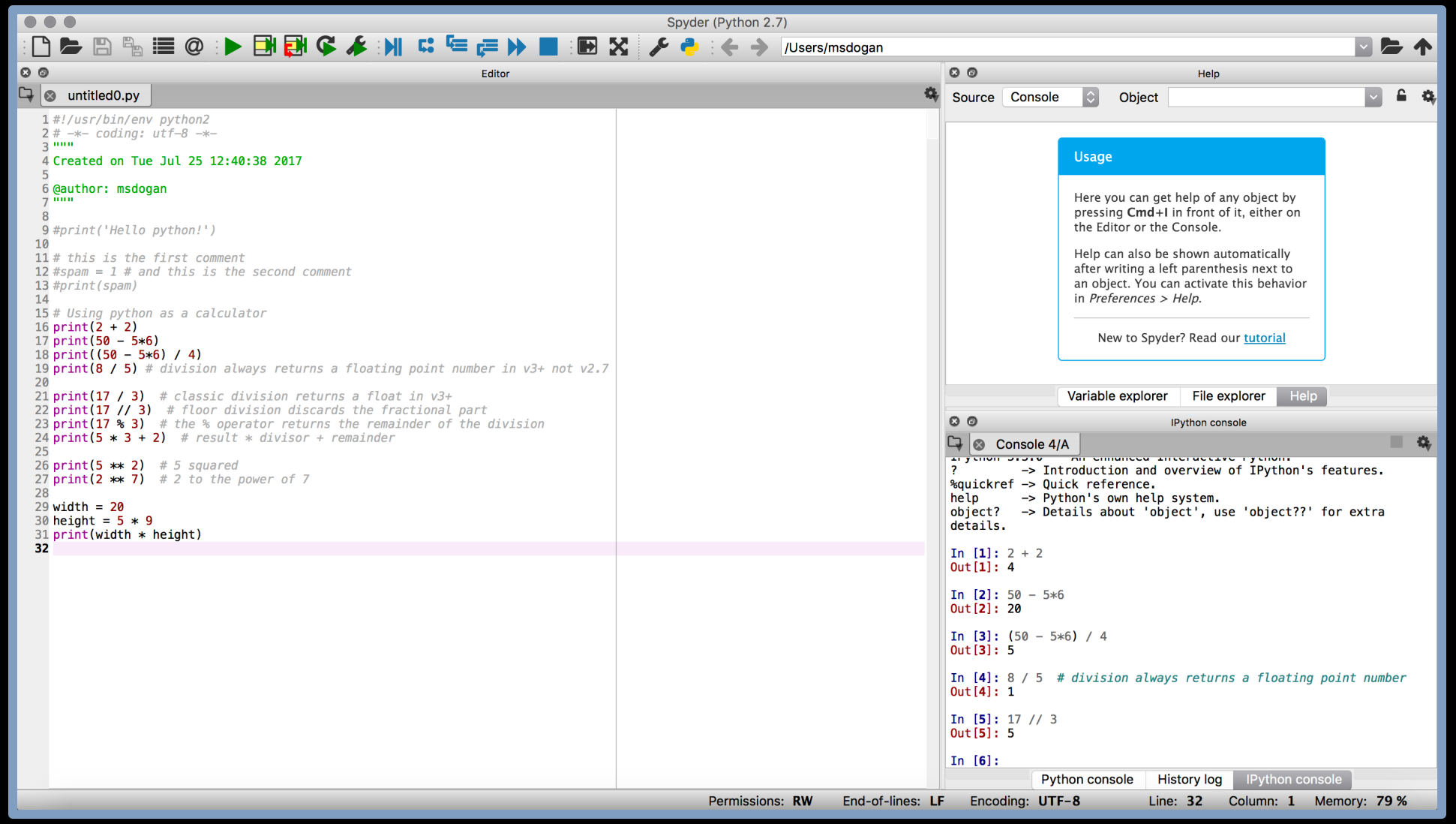
Script editor **Console**

Creating & printing variables & inserting comment (#)

```
# this is the first comment
spam = 1 # and this is the second comment
print(spam)
```

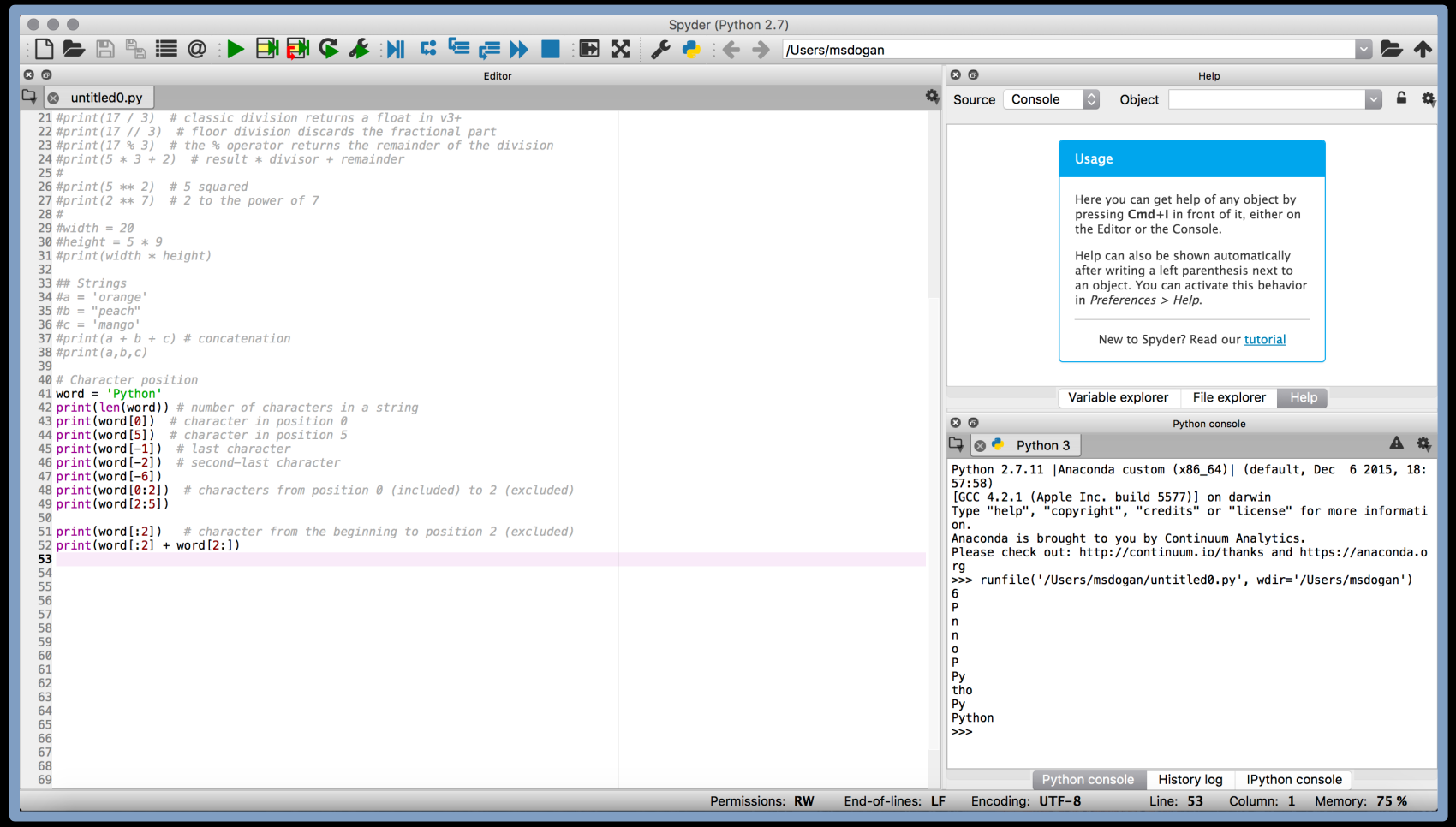


Using Python as a calculator



Strings

Strings must be enclosed in single ‘ ’ or double “ ” quotes

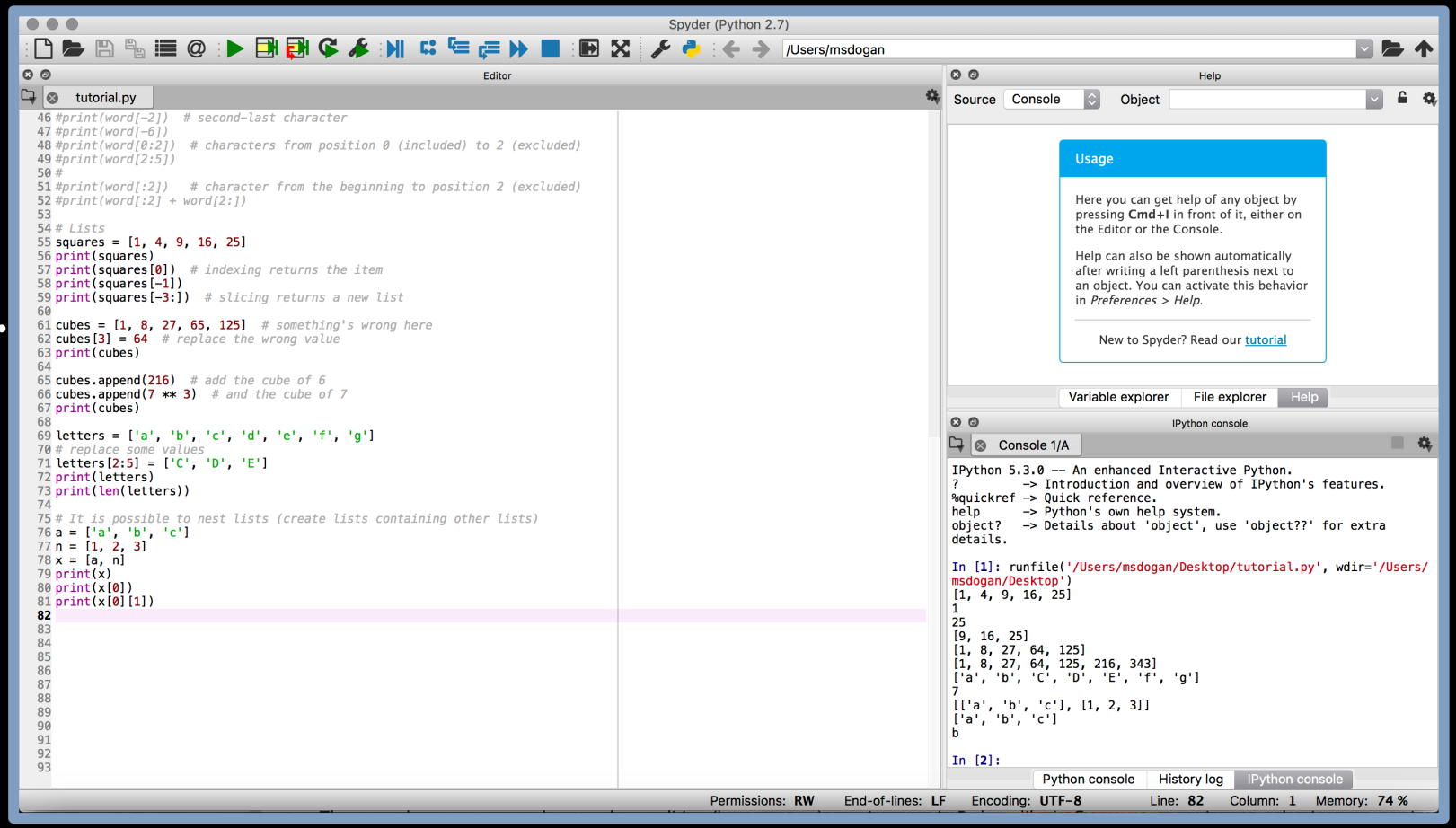


Ways to group, store and recall data

- List: `[ ]`
- Set: `{}`
- Dictionary: `{}`
- Array: `np.array()` (requires Numpy package)
- Dataframe: `pd.DataFrame()` (requires Pandas package)

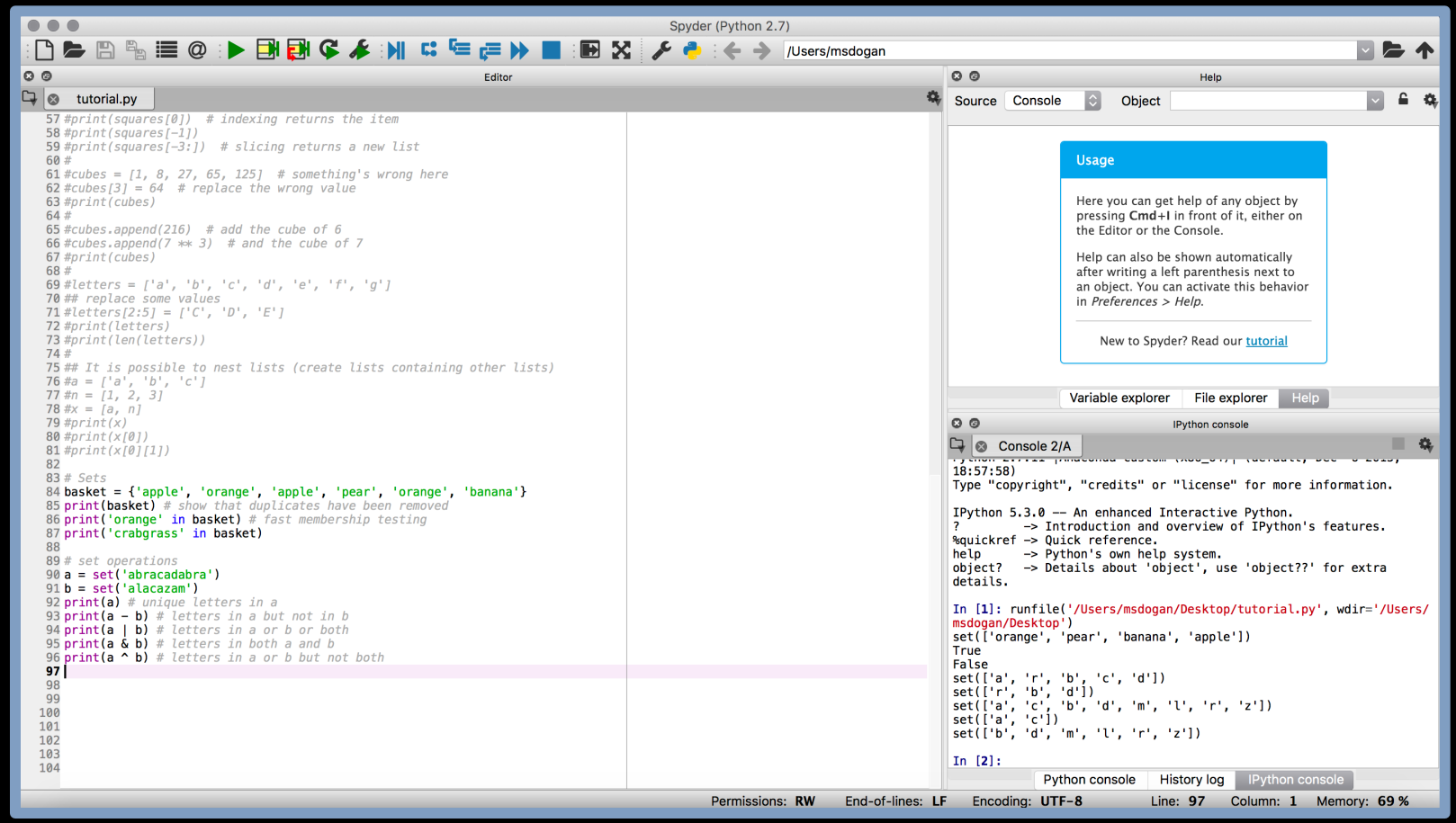
Lists

Python knows a number of *compound* data types, used to group together other values. The most versatile is the *list*, which can be written as a list of comma-separated values (items) between square brackets.



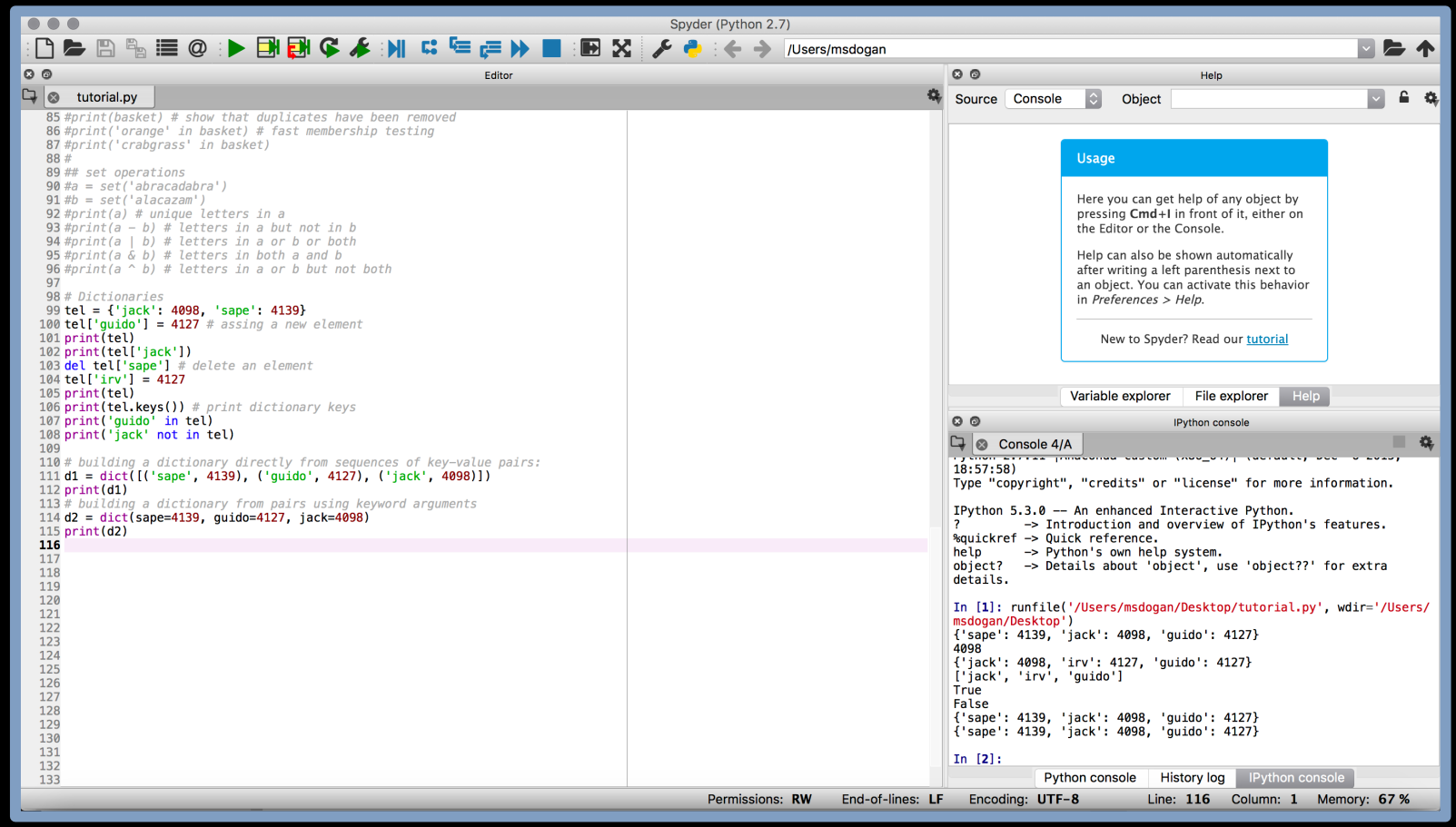
Sets

- A set is an unordered collection with no duplicate elements
- Sets support mathematical operations like union, intersection, difference, and symmetric difference



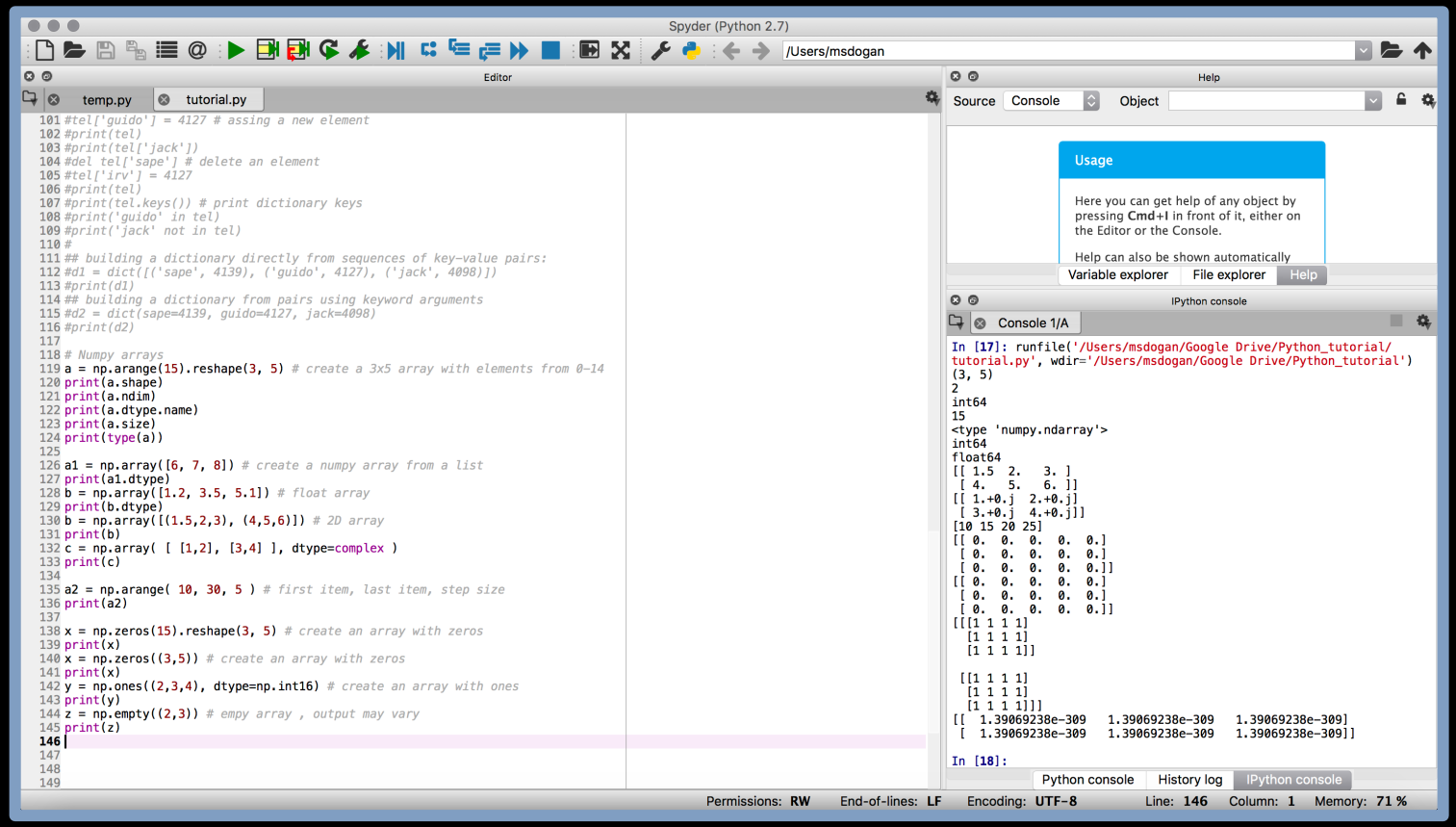
Dictionaries

- Dictionaries are indexed by *keys*
- The main operations on a dictionary are storing a value with some key and extracting the value given the key

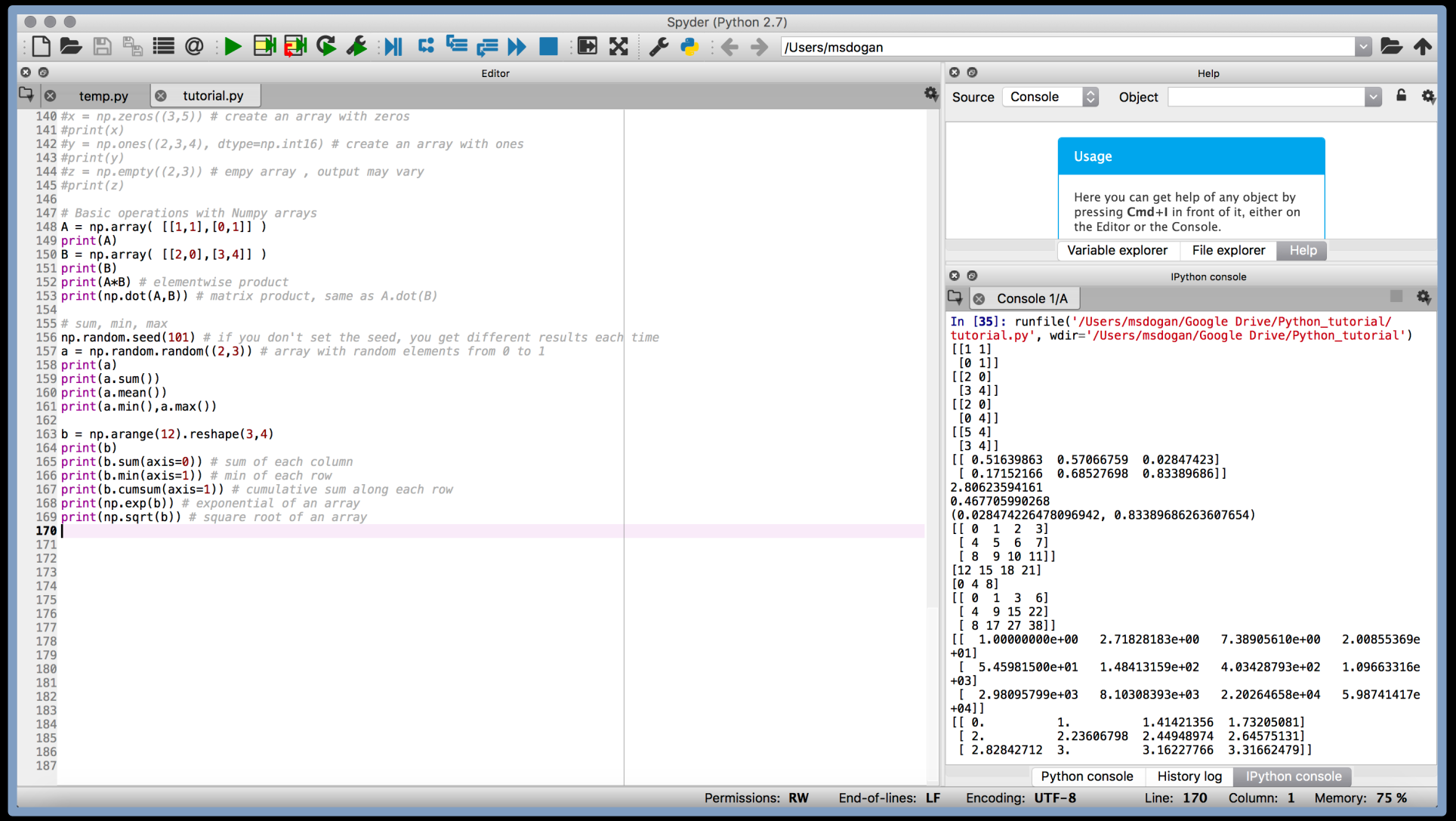


Numpy Arrays `np.array()`

- `import numpy as np`
- Homogeneous multidimensional array
- Table of elements, all of the same type

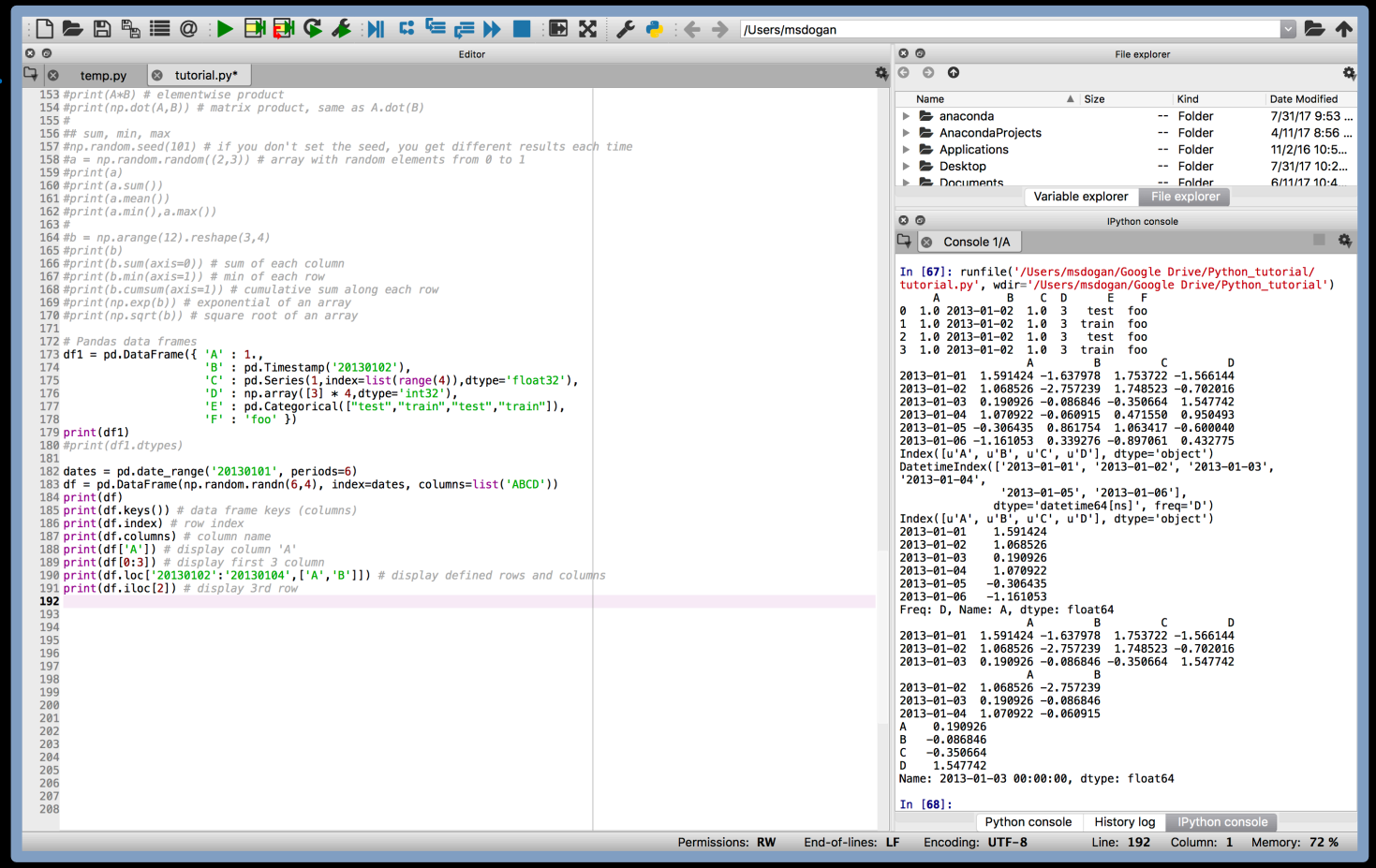


Some operations w/ Numpy arrays

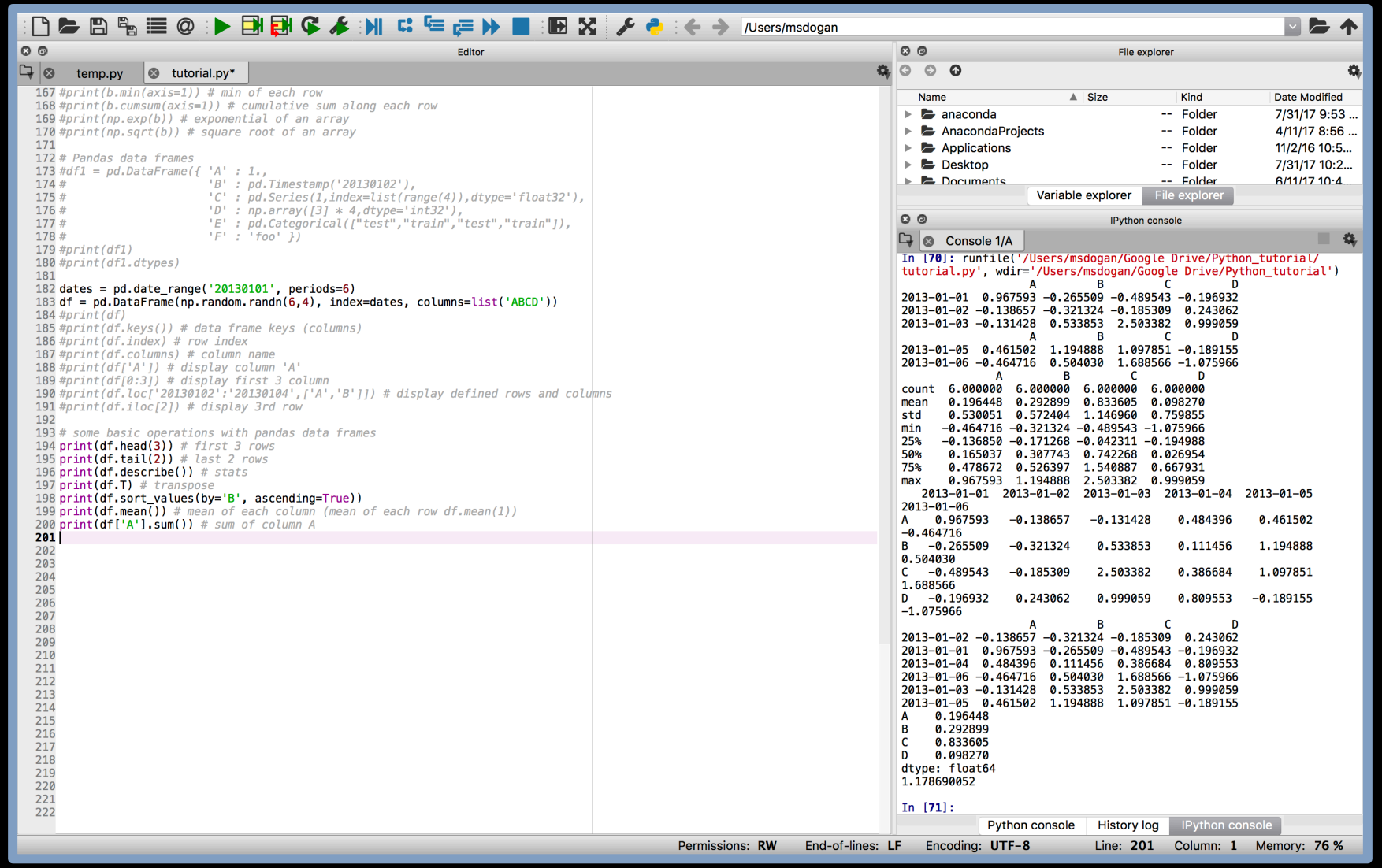


Pandas data frames `pd.DataFrame()`

- `import pandas as pd`
- Tabular data structure with labeled columns & rows
- Dictionary-like container for series objects

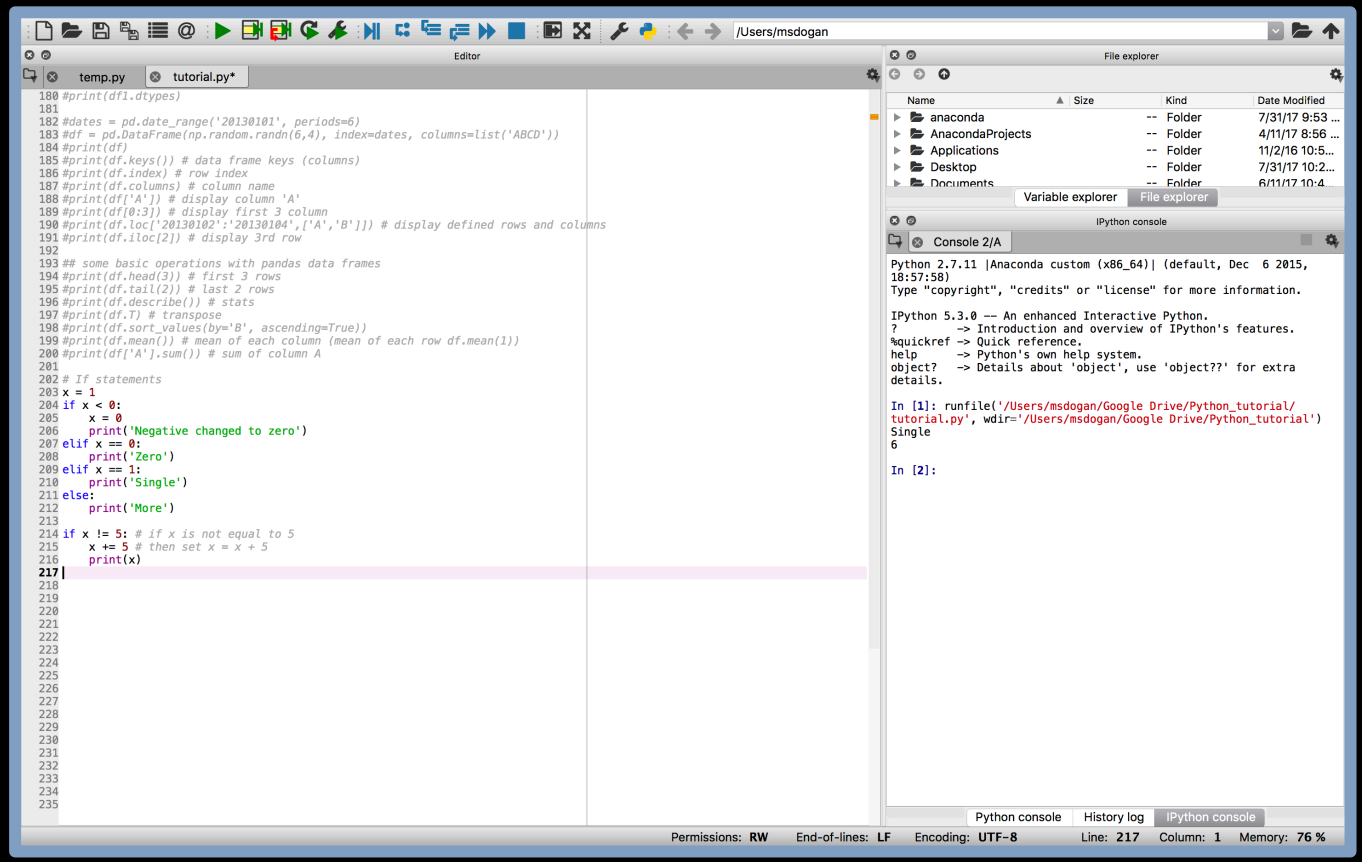


Some operations w/ Pandas dataframes



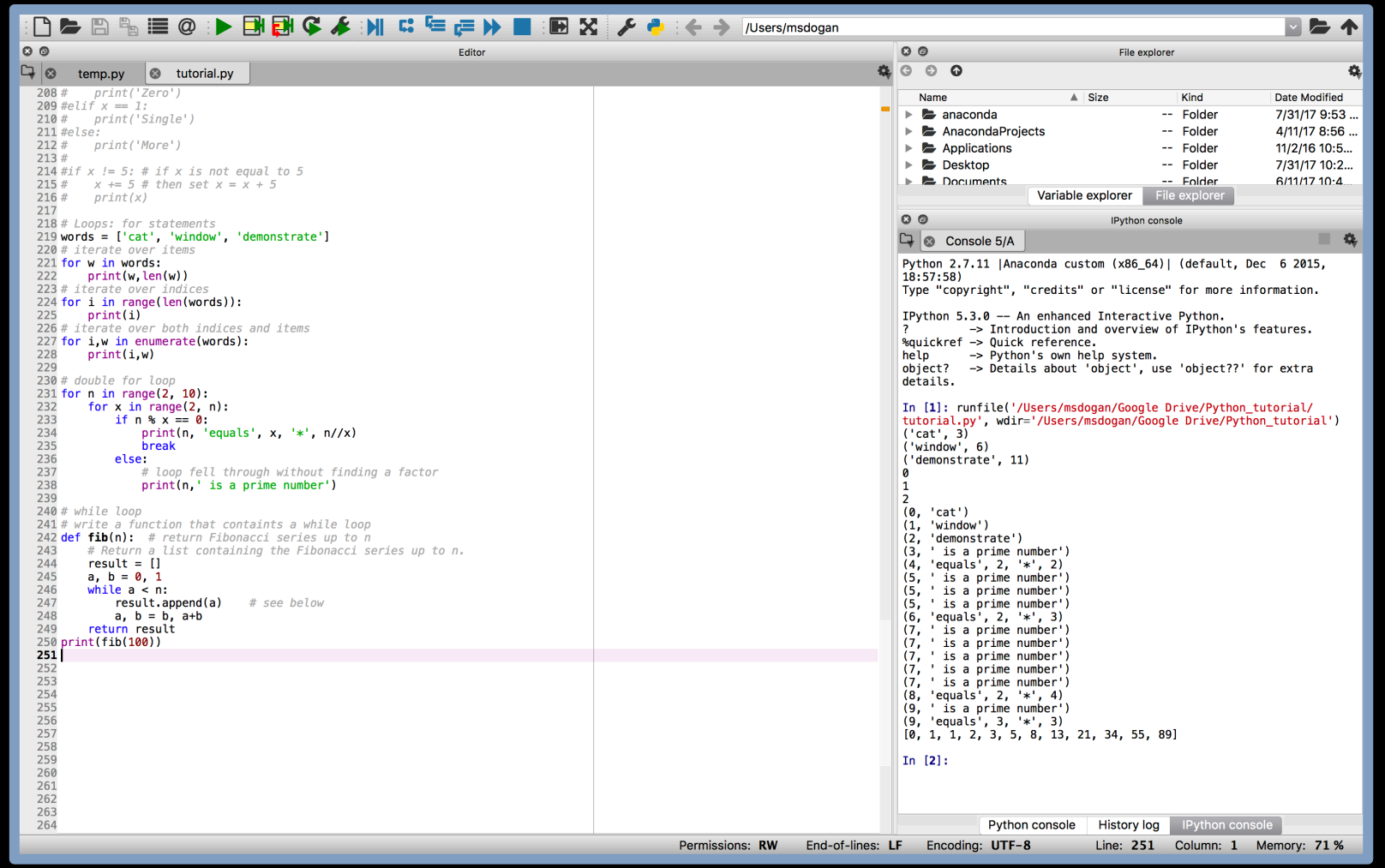
`if` statements: `if, elif, else`

- Less than: <
- Less than or equal: <=
- Greater than: >
- Greater than or equal: >=
- Equal: ==
- Not equal: != or <>



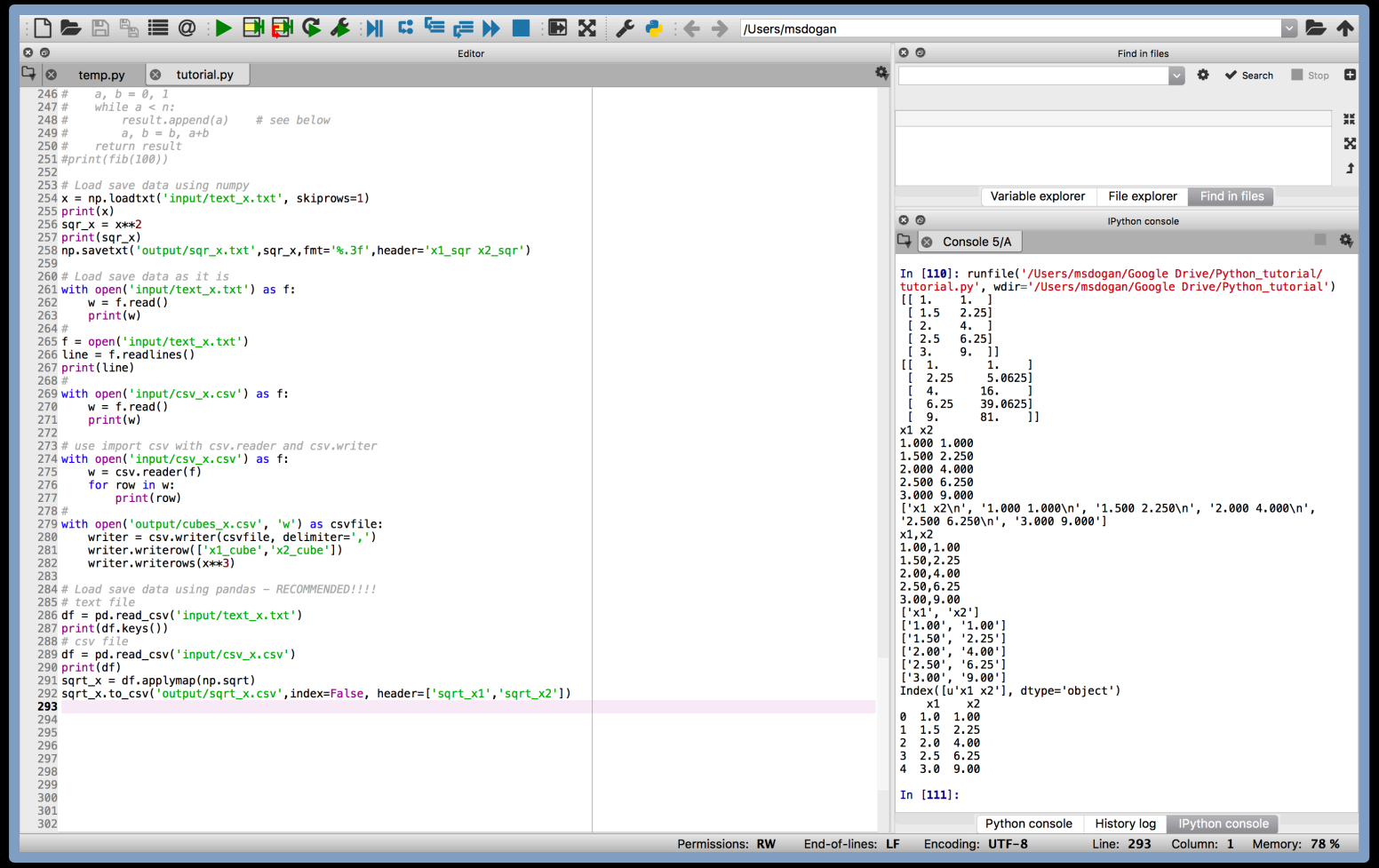
Loops: `for` & `while` statements

- For loops: Iterate over the items of any sequence (a list or a string), in the order that they appear in the sequence
- While loops: continue until exit criteria met



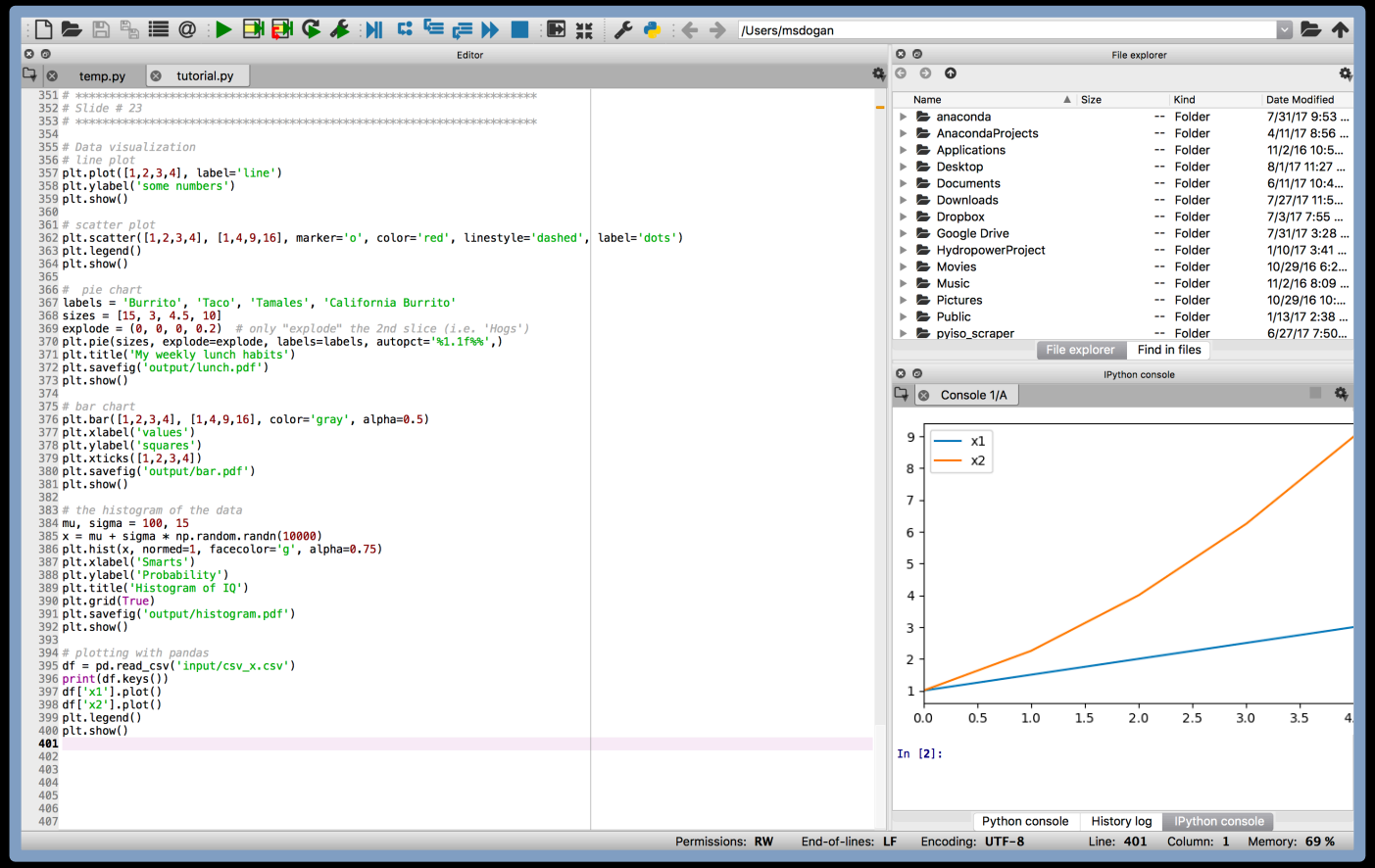
Loading & saving data

- Load / save data from / to `csv` and `txt`
- A few different ways: numpy, csv, and pandas



Data visualization

- `import matplotlib.pyplot as plt`
- Line plot
- Scatter plot
- Pie chart
- Bar chart
- Histogram
- Bonus: pandas plot



Linear Programming (LP) with CVXPY

- CVXPY is a Python-embedded modeling language for convex optimization problems
- It allows you to express your problem in a natural way that follows the math
- User's guide: http://www.cvxpy.org
- Install CVXPY by running the following command:
  - Windows/Mac OS: `pip install cvxpy`

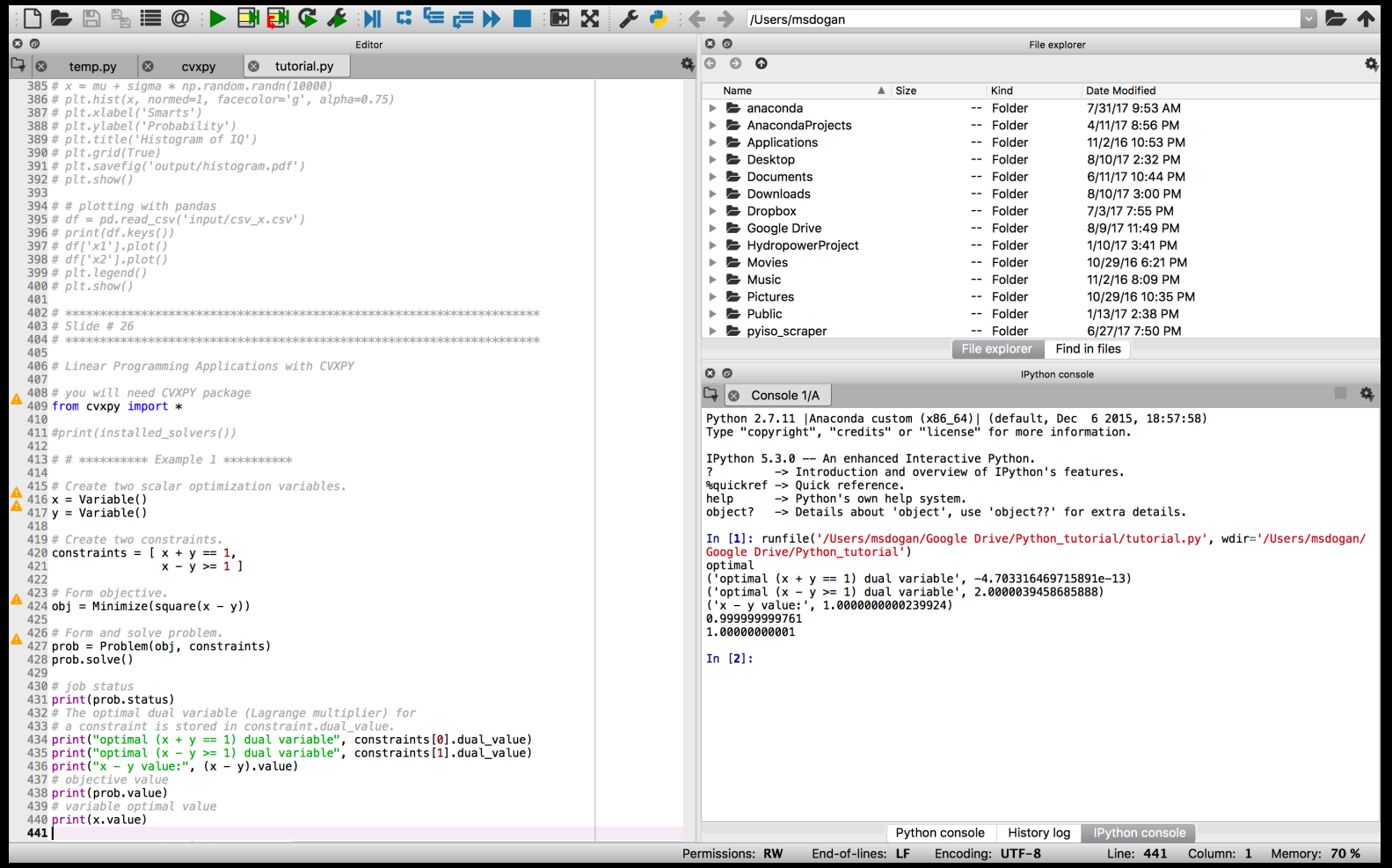
CVXPY optimization package

- Variables
  - Variables can be scalars, vectors, or matrices.
  - `x = Variable()`, `y = Variable(5)`, or `z = Variable(4, 7)`
- Constraints
  - You can use ==, <=, and >=
  - constraints are elementwise
  - `constraints = [ x + y == 1, x - y >= 1 ]`
- Parameters
  - Parameters are symbolic representations of constants
  - `c = Parameter(sign='negative'), c.value = 5`
- Objective
  - `obj = Minimize(square(x - y)), obj = Maximize(3*car + 5*truck)`

Example 1

Objective:
$minimize\ (x - y)^2$

Subject to:
$x + y = 1, x - y \geq 1$



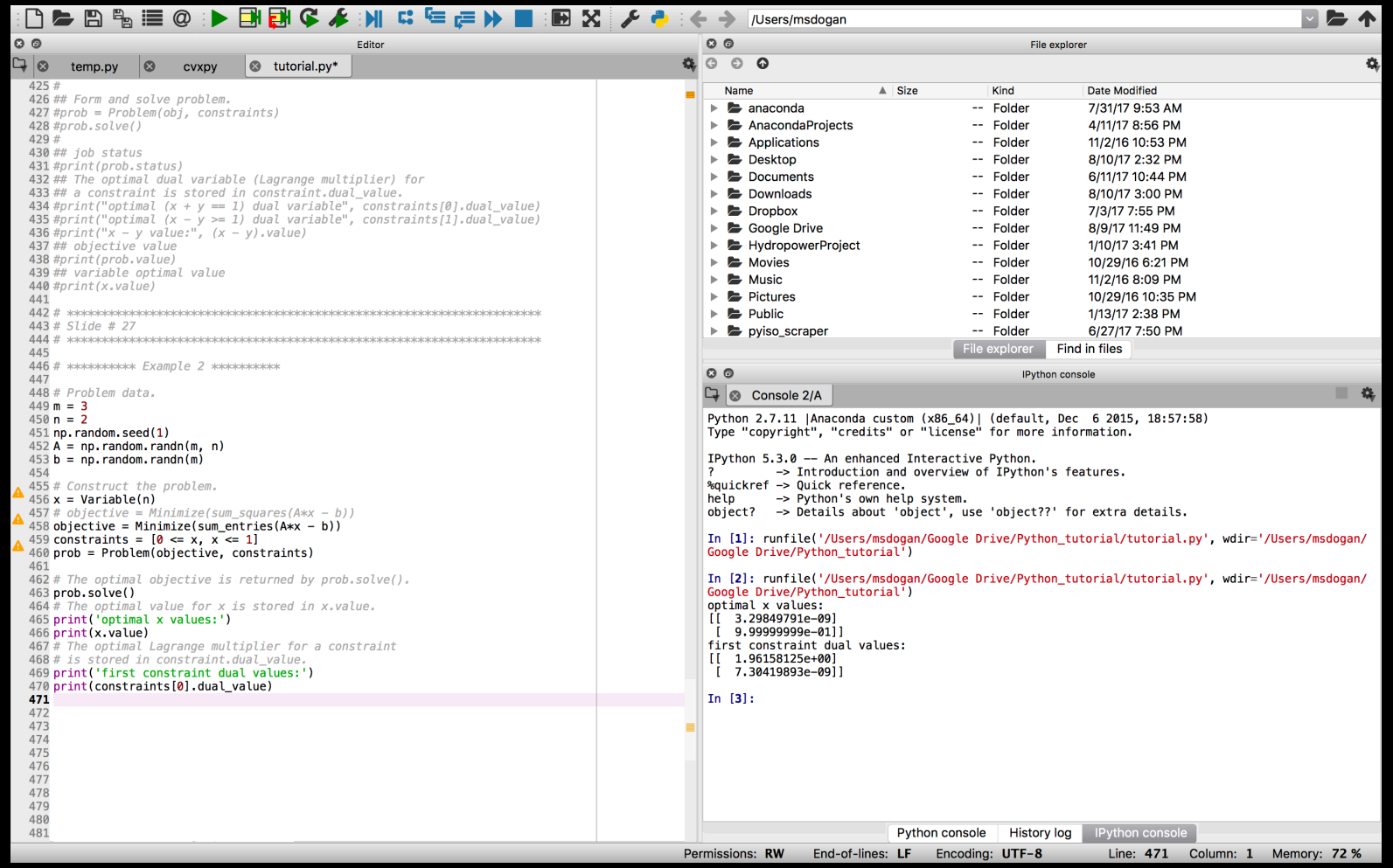
Example 2

- CVXPY problem with vectors and matrices
- Retrieve:
  - objective value
  - optimal values of variables
  - dual values of constraints

Example:

$$min \sum (Ax - b)$$

s.t. $0 \leq x \leq 1$



ECI 153 Clyde Motor Company

Deterministic Optimization and Design — Jay R. Lund
UC Davis — Fall 2015

Today we have a graphical example of a product-mix type problem frequently solved by linear programs.

Clyde Motor Company Example
A company has 3 factories that together produce two products: cars and small trucks.

Each factory specializes in certain components required for both products.
Plant 1 makes truck bodies.
Plant 2 makes car bodies.
Plant 3 makes shared components and assembles trucks and cars.

Each plant has a limited capacity.
Each product is sold for a fixed price with a fixed profit per unit.

Data: Plant	Product 1	Product 2	Capacity Available (units/hour)
1	1	0	4
2	0	2	12
3	3	2	18
Unit Profit	$3	$5	--

Step 1: **Define the Problem:**
- -) Max profit
- -) S.T. limited plant capacities
- -) How many cars and trucks should be produced?

Steps 2 and 3: **Express the Problem Mathematically:**

Max $Z = 3X_1 + 5X_2$ = profit per hour of operation, say.

Subject To:

(1) $1X_1 \leq 4$
(2) $2X_2 \leq 12$
(3) $3X_1 + 2X_2 \leq 18$
(4) $1X_1 \geq 0$
(5) $1X_2 \geq 0$

X_1 = Number of small trucks produced.
X_2 = Number of cars produced.

Explanation of constraints:

(1) $1X_1 \leq 4$
(2) $2X_2 \leq 12$
(3) $3X_1 + 2X_2 \leq 18$
} capacity constraints on production

(4) $X_1 \geq 0$
(5) $X_2 \geq 0$
} cannot produce negative products

37