API, JSON, CSV

Pili Hu

Recap and overview

- Data type:
  - Simple: int, float, str
  - Composite: list [], dict {}, tuple ()
- Functions, classes and modules
  - .member
  - .member()
- Built-in functions (operators)
  - Arithmetic
  - Math module
  - Str functions
    - .format()
- Control flow:
  - if
  - while
  - for

World is flat from now on!

Efficient learning paradigm

- Objective driven
- Jump between topics
- Use search
  - FAQ on open book repo
  - Issue tracker
  - Slack group
  - Google
  - Stack Overflow
- Ask to get quick feedback

Jupyter notebook

Concepts

- Virtual environment
- Jupyter notebook

They do not depend on each other but we usually use them together.

Exercise

- Setup virtual environment
- Install dependencies
- Open Jupyter notebook
- Can enter and exit Jupyter notebook
- Run one of your past example in Jupyter notebook

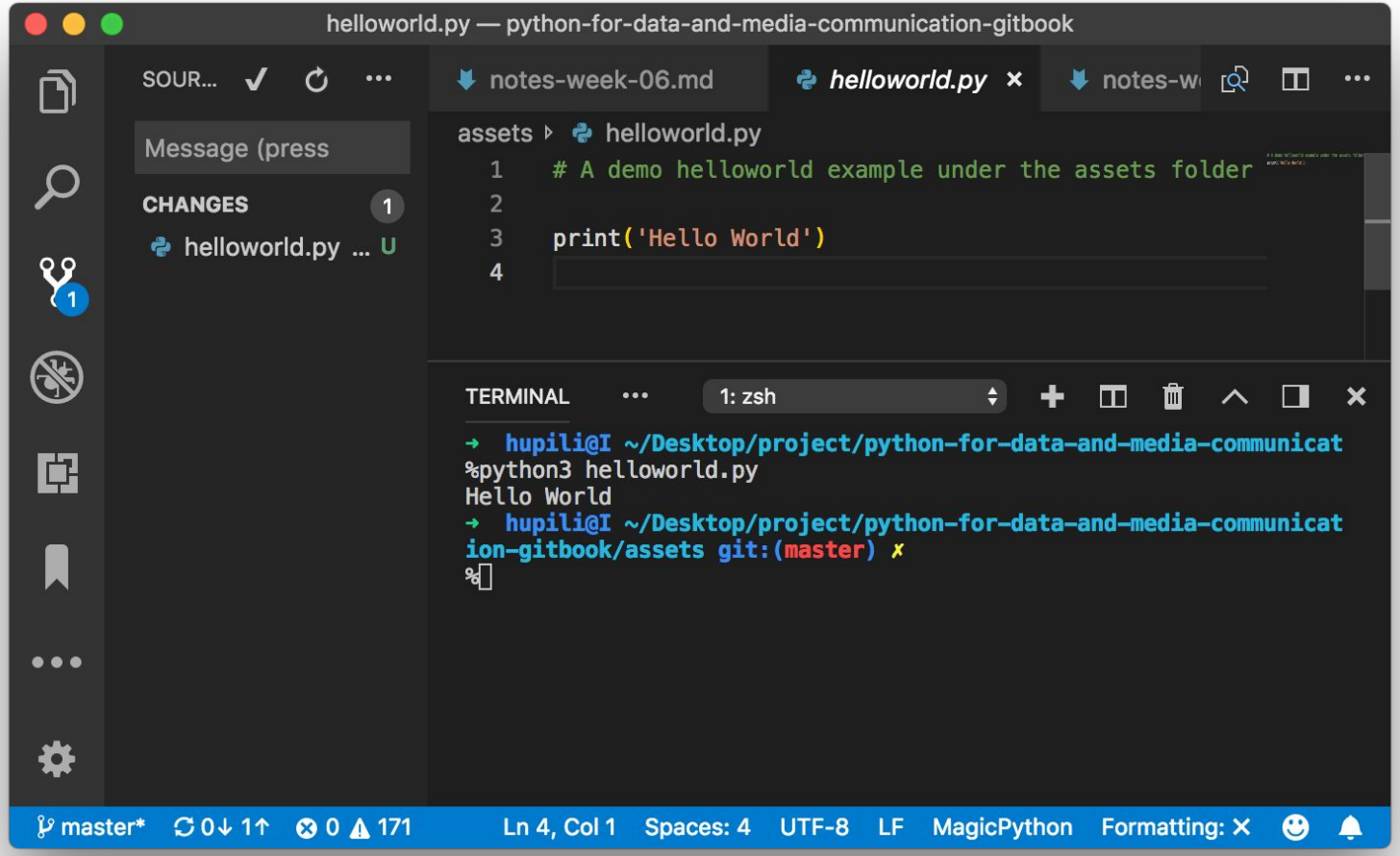
Several ways to write/ run Python

- Terminal + (any) Text Editor
- Interactive Python shell
  - >>>
- **Jupyter notebook**
- IDE:
  - IDLE
  - VSCode

    Spider

    PyCharm

VSCode: embedded terminal is useful



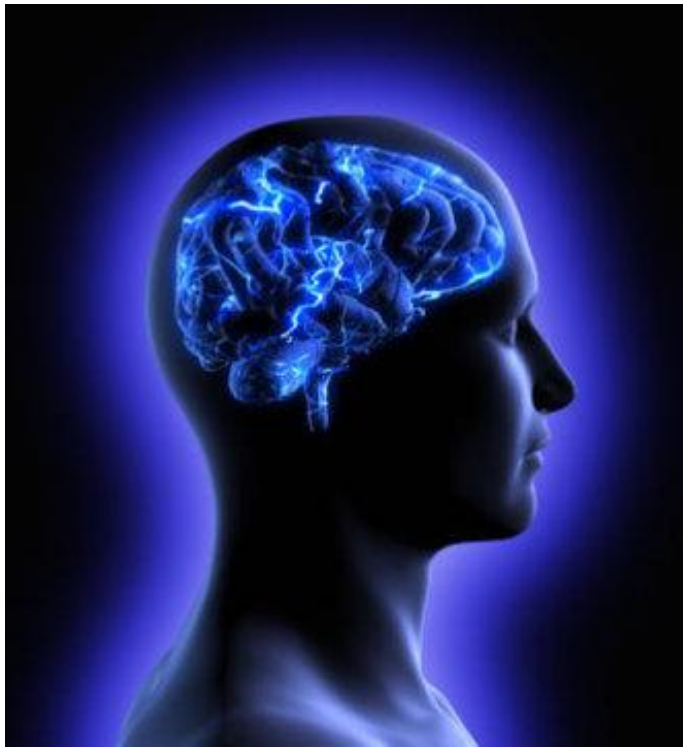
A word on Terminal and shell

- Terminal -- “the window” -- the access point for the end user.
- Shell -- “the worker” -- the program that runs a REPL to react to user command.

Input/ Output methods

Serialization and deserialization

序列化



Serialization

Deserialization

paper

audio

gesture

Order matters! → serialise

Python

Internal structure:

- Int
- Float
- Str
- Tuple
- List
- Dict

External format:

控制面板

- Console input/ output
- Txt
- Json
- Csv
- Xml

Exercise

File I/O

- Read and write file

分隔符

Delimiter Separated Values (DSV)

- Delimiter can be anything that is not part of valid data
- Common delimiters are:
  - Comma → CSV
  - Tab → TSV

Best practice of CSV

- Keep a header row
  - This makes the dataset self descriptive
- Put your data in:
  - list of list structure: [[], [], ...] → good to use with "csv"
  - list of dict structure: [{}, {}, ...] → good to use with "pandas"

Exercise

csv

- Write mortgage schedule to CSV file

JSON Syntax

http://www.json.org/

object
- **{}**
- **{** *members* **}**

members
- *pair*
- *pair* **,** *members*

pair
- *string* **:** *value*

array
- **[]**
- **[** *elements* **]**

elements
- *value*
- *value* **,** *elements*

value
- *string*
- *number*
- *object*
- *array*
- **true**
- **false**
- **null**

string
- **""**
- **"** *chars* **"**

chars
- *char*
- *char chars*

char
- *any-Unicode-character-except-"-or-\-or-control-character*
- **\"**
- ****
- **\/**
- **\b**
- **\f**
- **\n**
- **\r**
- **\t**
- **\u** *four-hex-digits*

number
- *int*
- *int frac*
- *int exp*
- *int frac exp*

int
- *digit*
- *digit1-9 digits*
- **-** *digit*
- **-** *digit1-9 digits*

frac
- **.** *digits*

exp
- *e digits*

digits
- *digit*
- *digit digits*

e
- **e**
- **e+**
- **e-**
- **E**
- **E+**
- **E-**

In natural language:

- Basic types:
  - Number
  - String
  - Boolean
  - Null
- Composite types:
  - Array/ List: []
  - Dict/ Object: {}

Visually:

```
{
    "title": {
        "text": "Hong Kong Population by gender"
    },
    "legend": {
        "data": ["Male", "Female"],
        "align": "left",
        "top": 20,
        "left": 10
    },
    "xAxis": [{
        "type": "category",
        "data": [
            1991,
            1996,
            2001
        ],
        "axisTick": {
            "alignWithLabel": true
        }
    }],
```

Exercise

json

- Load the mortgage parameters from json file

Exercise

Integrated practice

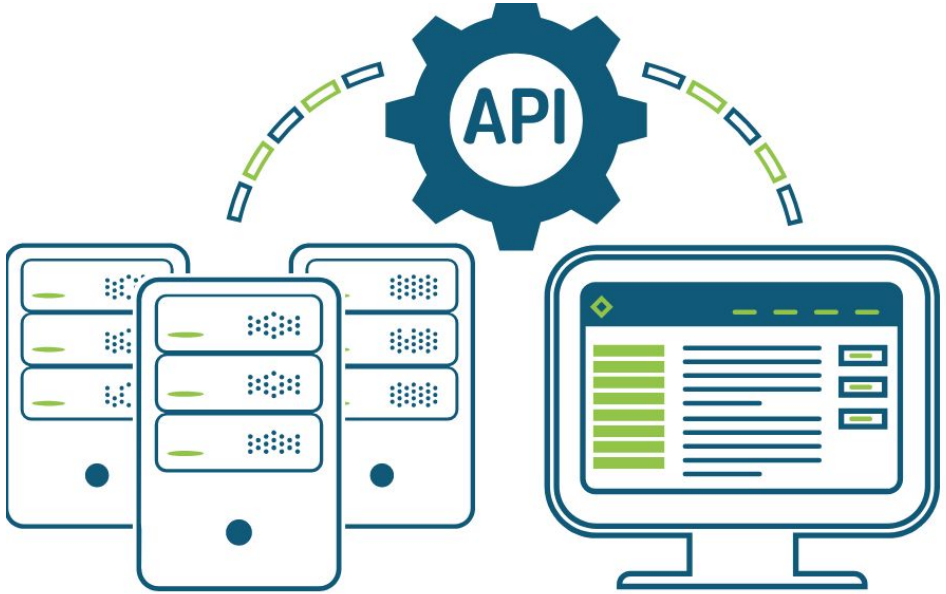
- Read a list of mortgage parameters
- Output the mortgage schedule to CSVs, one parameter set one file

Application Programming Interface (API)

Application Programming Interface

Regard API as:

- A magic black box
- An oracle
- A protocol



Common forms of API:

- **Local function call**
- Remote procedure call
- **HTTP API**
  - Restful style
  - RPC style

Exercise

Local function call

(take home exercise)

- Try geopy module to get the coordinates of certain places and distances between two points.

Exercise

HTTP API

- Try USGS earthquake API
- Search for earthquakes around your hometown in past 100 years.
- Save the record to a CSV for further analysis.

Exercise

Integrated exercise
(Bonus)

Implement an earthquake bot.

- Get data from USGS
- Post tweets using python-twitter module.