# Table of Contents

|  |
| --- |
| Research | Current Trends | Use-cases | Project Cycle | Presentation | Best Practices |
| Situation Understanding |
| Data quality | Acquisition | Data Mining/Sql | Profiling/Descriptive Statistics |
| Cleaning | Transformation | Prelim Exploration/Slice & Dicing | Exploration/Visualization | Big Data |
| Methods Selection | Feature Engineering | Pre-processing |
| * Techniques & methods | Application Notes |
| Insight | Action | Results |
| * Data Architecture & Engineering | Databases |
| Bash | Surveys | open data | Scaling | ML/Cloud Tools | Software Eng. | Q&A |

# Google's Python Class

To get started, the Python sections are linked at the left -- [Python Set Up](https://developers.google.com/edu/python/set-up) to get Python installed on your machine, [Python Introduction](https://developers.google.com/edu/python/introduction) for an introduction to the language, and then [Python Strings](https://developers.google.com/edu/python/strings) starts the coding material, leading to the first exercise. The end of each written section includes a link to the code exercise for that section's material. The lecture videos parallel the written materials, introducing Python, then strings, then first exercises, and so on. At Google, all this material makes up an intensive 2-day class, so the videos are organized as the day-1 and day-2 sections.

This material was created by [Nick Parlante](http://www-cs-faculty.stanford.edu/%7enick/) working in the engEDU group at Google. Special thanks for the help from my Google colleagues John Cox, Steve Glassman, Piotr Kaminksi, and Antoine Picard. And finally thanks to Google and my director Maggie Johnson for the enlightened generosity to put these materials out on the internet for free under the [Creative Commons Attribution 2.5](http://creativecommons.org/licenses/by/2.5/) license -- share and enjoy!

## findall and Groups

|  |
| --- |
| Step > Value |
| Assumptions: |
| Additional Resources: |

The parenthesis ( ) group mechan

  # Open file  
  f = open('test.txt', 'r')  
  # Feed the file text into findall(); it returns a list of all the found strings  
  strings = re.findall(r'some pattern', f.read())