

1 Basic Report Writing

1.1 Subsection

1.1.1 Subsubsection

With some text in it.

1.1.2 Another subsubsection

Hello world. Here's some text in the first paragraph of this subsubsection.

Here's another paragraph in the same subsubsection.

2 Math

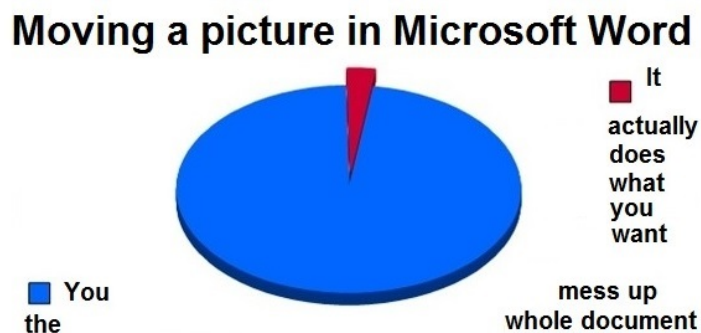
2.1 Some Random Equation

$$y = \sqrt[3]{\frac{5x}{x^2}}$$

This is an inline math equation: $y = x^2 + 2x + 4$...in the paragraph following Some Random Equation.

3 Figures and Images

Figure 1: A funny LaTeX meme



For more examples and settings, [click this link](#).

4 Lists

4.1 Non-numbered List

- First item
- Second item
 - First second item
 - Second second item
- ! Last item with custom exclamation point marker

4.2 Numbered Lists

1. Numbered
2. list of
3. things
 - (a) first thing
 - (b) second thing
- ! exclaimed item

NOTE this item

→ custom arrow bullet in list

4. list cont.

5 Tables

head1	head2	head3
cell4	cell5	cell6
cell7 W/kg	cell8 N	cell9 seconds

6 Code

```
1 import pandas as pd
2 from tools.loading import print_loading_bar
3
4
5 data_source_str = "wordle-solver/data/words_3000" # Designates data to process
6
7 # Open list of words
8 fh = open(data_source_str + "_raw.txt")
9 LENGTH = len(fh.readlines()) # get number of words
10 fh = open(data_source_str + "_raw.txt") # reopen file handle
11
12 # Load letter frequency lookup table
13 freqs = pd.read_csv("wordle-solver/data/lookup_char-freqs.csv", index_col=0)
14
15 # Initialize data storage
16 words = pd.DataFrame()
17
18 WORD_LENGTH = 5 # Word length setting (Wordle is currently a 5-letter game)
19
20 # Process list of words
21 for i, word in enumerate(fh):
22
23     print_loading_bar(i, LENGTH, title="Preparing word data: ", size=100, no_newline=True)
24
25     word = word.strip().lower()
26
27     if len(word) == WORD_LENGTH:
28         words.loc[i, 'word'] = word
```

```

29     score = 0
30     for j in range(WORD_LENGTH):
31         col = 'char'+str(j)
32         words.loc[i, col] = word[j]
33         score = freqs.loc[word[j], col] + score
34     words.loc[i, 'score'] = score
35
36 words = words.sort_values('score', ascending=False)
37 words.to_csv(data_source_str + "_clean.csv")

```

For more code examples and parameters, [click this link](#).