Markdown is the most widely adopted way to format documents using plain text syntax. Here are the essentials.

STYLIZE TEXT

italic	`*italic*`
bold	`**bold**`
~~strikethrough~~	`~~strikethrough~~`
code in-line	`` `code in-line` ``
code block	` ```code block``` `
> text	blockquote

HEADERS AND BREAKS

# Text	H1
## Text	H2
### Text	H3
---	Horizontal Rule

LISTS

Lists can be unordered or ordered, which is determined by the symbol used to list them. Unordered lists have either **-**, **+**, or *****, while ordered lists require any integer followed by a **.** or **)** character.

An ordered list:

```
1. First item
2. Second item
3. Third item
```

Unordered List:

```
- First item
- Second item
- Third item
```

Combining the two:

```
1. First ordered list item

1. Second ordered list item (which
   shows as 2.)

1. (Markdown ignores the written
   number in favor of enumerating
   by integer)

   - Indented lists require four
     spaces in most specifications

   - Sublists are implementation
     specific

   - Try it out and render to be
     sure it works

1. Continuing the list after an
   indentation
```

TABLES

Tables are a common extension of the official markdown definition (CommonMark). They can be formatted in quite a lot of ways:

```
| Column Title      | Another Column          | One More |
|-------------------|:---------|:---------|
| **Bolded text**   | Columns don't need to be aligned |          |
| Or even filled in |                         |   code   |
```

Renders as:

Column Title	Another Column	One More
bolded text	Columns don't need to be aligned	
Or even filled in		`code`

That can be painful to look at. Here's a prettier example with alignment based on the placement of the **:** in the table structure:

```
| Tables   |      Are      |  Cool |
|:---------|:-------------:|------:|
| col 1 is | left-aligned  |    $1 |
| col 2 is |   centered    |    $2 |
| col 3 is | right-aligned |    $3 |
```

Tables	Are	Cool
col 1 is	left-aligned	$1
col 2 is	centered	$2
col 3 is	right-aligned	$3

Tables are always a little tricky to remember. Use available online tools to help build them, like: https://csvtomd.com

LINKS

There are two ways to create links. The most commonly used format includes brackets followed by parentheses: **[]()**. To render a link to an image, begin with a **!**.

Examples.

```
[Link inline this way](https://opensource.com)

[Or add a title for the link](https://opensource.com "Google's Homepage")

[Here is a relative link within a repository](../blob/master/LICENSE)

![I link to an image](path/to/image.png)
```

The second format involves brackets, **[][]** or **[]**, followed by a reference formatted with a bracket and colon, **[]:**, anywhere else in the document.

Examples.

```
[Reference-style link][Case-insensitive Reference Text]

[Numbers are commonly used][1]

[Text can link on its own as a reference]

[case-insensitive reference text]: https://opensource.com

[1]: https://opensource.com

[text can link on its own as a reference]: https://opensource.com
```

URLs, in and out of angle brackets, will automatically get turned into links with most implementations.

Example.

```
Both https://opensource.com and <https://opensource.com> will render as
links, as well as opensource.com on some renderings.
```