Tables Demonstration Idea

What makes a table

A table in markdown will look like a table you might get in email.
It's a bunch of data in columns. It might have titles or totals. It
might even have some breakdown. It doesn't have multi-line headers, some
cells that span multiple columns, centered columns or any of the stuff
you might see in HTML but not in email.

Here's a simple table:

```
  3   Elven Kings     under the sky
  9   Mortal Man      doomed to die
  1   Dark Lord       on dark throne
```

A table is text, separated out by blank lines before and after.

A table is a table because it has two or more columns and two or more
rows, where each column runs two or more spaces (or tabs) down the entire table.
I recommend aligning text on the left and numbers
by the decimal point. The first column should start on the left margin or else it could
be a code block. Tables are forgiving about exact column placement; think if you could read
it in an email. If it's unambiguous, it probably works. All the odd spacing will be ignored
in rendering anyway.

Header

Tables may have a *header*, which would be the column headings for the
values below it. There's only one header line; use an HTML escape if you want fancy.
The header may be just emphasized text for each column, aligned with the column:

```
*#*   *Owner*         *Notes*
  3   Elven Kings     under the sky
  9   Mortal Man      doomed to die
  1   Dark Lord       on dark throne
```

Or it can have plain names with a second line of dashes or equal signs, which line up:

```
  #   Owner          Notes
  -   ------------   ------------
  3   Elven Kings    under the sky
  9   Mortal Man     doomed to die
  1   Dark Lord      on dark throne
```

You can even skip some columns

```
 _#_                 _Notes_
  3   Elven Kings    under the sky
  9   Mortal Man     doomed to die
  1   Dark Lord      on dark throne
```

Footer

Tables may have a *footer*, or final line of the table.
This is usually used for totals. Like the header, each
element should line up with a column and can be emphasized:

```
  3   Elven Kings    under the sky
  9   Mortal Man     doomed to die
  1   Dark Lord      on dark throne
_13_  _3 Races_      _One Ring To Bind Them_
```

or use a line of `-` or `=` above the footer:

```
  3   Elven Kings    under the sky
  9   Mortal Man     doomed to die
  1   Dark Lord      on dark throne
===   =========      =========
 13   3 Races        One Ring to Bind Them
```

Computed Values

Computed values, like totaling a column, appear often in tables. Its
annoying in email how often someone adds a line but fails to update
the column. We can use a computed field in the total line to make the
right thing happen. Here's an example with
the `<+>` (sum) and `<#>` (count) fields:

```
    3   Elven Kings     under the sky
    7   Dwarf Lords     halls of stone
    9   Mortal Man      doomed to die
    1   Dark Lord       on dark throne
  ===   =========       =========
_<+>_   _<#> Races_     One Ring to Bind Them
```

Here are all the computed fields:

- `<+>` , sum all the numbers. It is permissive: `$ 12,345.67` is the number 12345.67, but `n/a` is 0.
- `<#>` , counts the number of non-blank entries.
- `<->` , only works if there are exactly two numbers, it is the first number minus the second.
- `<avg>` , the average. Think `<+>` / `<#>` .
- `<%>` , the weird percentage thing. This works *only* if you left enough spaces in the data so to insert a new column in the data *and* the column to the left of it is a bunch of numbers. Then this will insert a new data column with percentages.

Here's a slightly more complex example:

```
_Monster_  _Hoard_       _Hit Points Each_
Orcs         1,000       7
Ent            100       43
Nazgul           7       430
------     -------  ---  -------
<#> Types   <+>     <%>  <avg> Average HP per type
```

Which would be the same as:

```
_Monster_  _Hoard_            _Hit Points Each_
Orcs         1,000    90.3%   7
Ent            100     9.0%   43
Nazgul           7     0.6%   430
------     -------      ---   -------
3 Types      1,107     100%   160 Average HP per type
```

Subtotals

Subtotal lines may occur in tables where one column has

sorted or unsorted lists. It's a way to put together nested subtotals in a limited way for how we usually write documents.

A subtotal line has one or more computed values, e.g., `<+>` and is the line above a sorted or unsorted list. The computed value looks at all values in the list, including nested lists, except it doesn't look into nested list it finds a subtotal value.

Here's an example

```
_My Nested Column_        _Value 1_   _Value 2_
* Top Level Subtotal <#>        <+>         <->
  * SubItem 1                   300         400
  * SubItem 2, not subtotal       5
    * Sub Sub Item               40           1
```

- '<#>' counts all non-subtotalled, non-blank items below it, so it is 3.
- '<+>' is 345 (300 + 5 + 40)
- '<->' is 399 (400 - 1)

Giant Complicated Example

Here's a complex example by way of explanation:

```
Loot                                Gold Pieces     % Loot
------                              ---------       -
* Grabbed by <#> Brave Adventurers     <+>          <%>
    1. Blammo, Wizard                  <+>
        * Rings
            Ring of Caffiene         2,000
            Immolation Ring          4,000
        * Weapons
            Multi-dagger of Swiss      500
    2.  Ogg, Half-troll                <+>
        * Weapons
            * Rail spar                  5
            * Spear of Destiny +2    9,000
        * Armor
            * Fur of Pimping +1      3,000
    3.  Pleace, Cleric                 <+>
        * Weapons
            * A Grazing Mace +3     10,000
```

```
            * Halibut of Smite        2,000
* Loose Change                        <+>
    * Cash Found                      7,000
    * Villager's Reward              10,000
    * Gem Value                        <->
        * Gems                         <+>
            * Big Diamond             2,500
            * 6 small rubies          1,200
            * bag of garnets            800
        * Jeweler's Fee                 900
-----                                    -   ------
Total Haul                           <+>   <%>
```

Look at it piece by piece:

Blammo

```
1. Blammo, Wizard                   <+>
    * Rings
        Ring of Caffiene          2,000
        Immolation Ring           4,000
    * Weapons
        Multi-dagger of Swiss       500
```

- The `<+>` makes the Blammo, Wizard line a subtotal. The line has an unordered list immediately below it.
- The `<+>` adds all values below, none of which are subtotals.
- `<+>` becomes 6,500 (2,000 + 4,000 + 500).

Equipment Found

```
* Grabbed by <#> Brave Adventurers      <+>          <%>
```

- The computed values make this a subtotal line. It has an ordered list immediately below it.
- The ordered list contains three subtotals, which are already computed, as computations are deepest first.
- The `<#>` finds three children (Blammo, Ogg, and Pleace), all of which are subtotals.
  It does not count inside the subtotals and so `<#>` is set to 3.
- The `<+>` adds the values below it, three subtotal values, set to 30,505 (6,500 + 12,005 + 12,000)
- The `<%>` looks to the previous column, calculates percentages for each of the three values: 21.3% (6,500/30505), 39.3% (12,005/30,505), 39.3% (12,000/30,505). These are inserted into the blank

column space next to Blammo, Ogg, and Pleace, as well as a “100.0%” on this line.

Loose Change

```
4.  Loose Change                    <+>
    * Cash Found                  7,000
    * Villager's Reward          10,000
    * Gem Value                    <->
        * Gems                     <+>
            * Big Diamond          2,500
            * 6 small rubies       1,200
            * bag of garnets         800
        * Jeweler's Fee              900
```

- There a several subtotal lines. Compute from the deepest level first.
- Gems is a subtotal line. The `<+>` becomes 4,500 (2,500 + 1,200 + 800)
- Gem Value is a subtotal line. It has two values below it, the 4,500 subtotal of Gems and the Jeweler’s Fee of 900. The `<->` becomes 3,600 (4,500 - 900).
- Loose Change is a subtotal line. The `<+>` becomes 20,600 (7,000 + 10,000 + 3,600). The details under Gem Value aren’t looked at when computing this because Gem Value is a subtotal line with a value in this column.

The Total Line

```
-----                                  -   ------
Total Haul                           <+>   <%>
```

- In `<+>` adds up the 2 subtotal values (Grabbed By and Loose Change) and becomes 51,105 (30,505 + 20,600).
- The percentage calculates the two percentages, 56.7% (30,505 / 51,105) and 40.3% (20,600 / 51,105).
  This *replaces* the `100%` on the ‘Grabbed By’ line with `56.7%`, adds the 40.3% to the ‘Loose Change’
  line, and a `100.0%` to the total line.

Here’s A Boring Rendering of this table

Loot	Gold Pieces	% Loot
Grabbed by 3 Brave Adventurers	30,505	56.7%

1. Blammo, Wizard	6,500	21.3%
Rings		
Ring of Caffiene	2,000	
Immolation Ring	4,000	
Weapons		
Multi-dagger of Swiss	500	
2. Ogg, Half-troll	12,005	39.3%
Weapons		
Rail spar	5	
Spear of Destiny +2	9,000	
Armor		
Fur of Pimping +1	3,000	
3. Pleace, Cleric	12,000	39.3%
Weapons		
A Grazing Mace +3	10,000	
Halibut of Smite	2,000	
Loose Change	20,600	40.3%
Cash Found	7,000	
Villager's Reward	10,000	
Gem Value	3,600	
Gems	4,500	
Big Diamond	2,500	
6 small rubies	1,200	
bag of garnets	800	
Jeweler's Fee	900	

Total Haul	51,105	100.0%

Final Questions

```
"How do I center the titles, pick the number formatting, make multi-line headers, skip c
```

You don’t. The idea plan is to mostly make really simple tables. Anything at all complicated probably needs HTML escapes. This isn’t a big mini-language, its just a way to render tables easily.

```
"My table doesn't render, I just get junk."
```

Something broke. Just like Markdown, the default action is to just render the text. If columns seem pushed together, look for some data that gets too close to the next column. Save a copy and try removing the header, footer, or some of the lines until you find the offending character.

```
"This makes my life great!"
```

Thank you.