

Listapalooza!

*Tips-n-tricks with true and pseudo lists**

PRESENTED BY

Nita Beck
Nita Beck Communications



**The director's cut...*



ABOUT ME

- Tech comm business owner
- Flare consultant
- Information architect / tech comm strategist
- **Passionate about**
 - Helping my clients to their best work
 - Supporting the Flare user community (aka the *Flare hive brain*) worldwide



ABOUT THIS PRESENTATION

- My goals
 - Encourage the use of true HTML lists for most bullet and numbered lists
 - Inspire ideas for things that act like lists, but aren't
- Not a workshop, but
 - Later, [download](#) and play with my Listapalooza Demo project
- Ask questions as we go!



WHAT WE'LL COVER TODAY

1. CSS tricks for lists nested within OL and UL lists
2. Custom OL and UL list formats
3. Definition lists (aka description lists)
4. Paragraph classes that act like OLs
5. Auto-numbering table rows (two examples)
6. Cross-references to “step” LIs in OLs
7. Flare UI tricks for quickly crafting OL and UL lists
8. **Bonus feature:** List item snippets



MY DESIGN GOALS

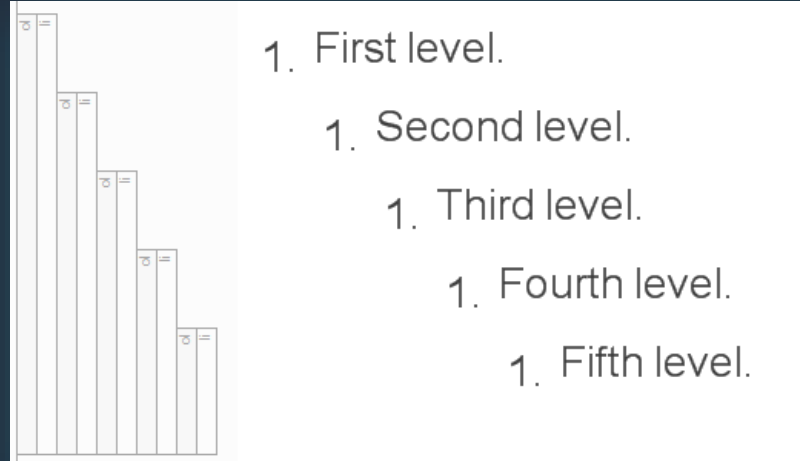
- Seek the **simplest solution** because that will be the easiest to maintain
- Seek the solution that requires **as little author intervention as possible**



CSS tricks for lists nested within OL and UL lists

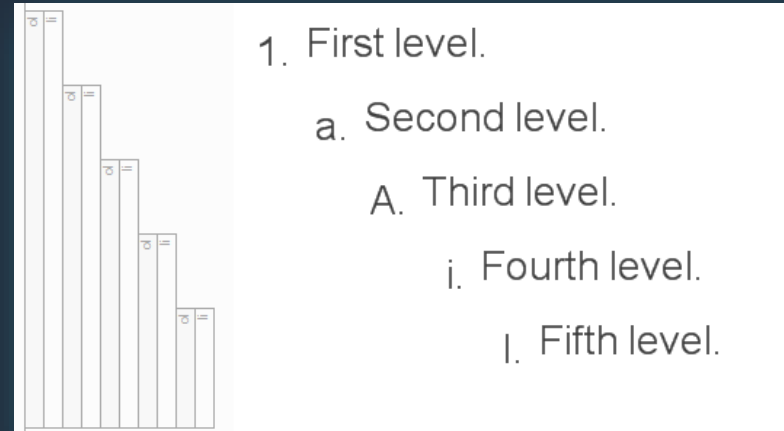
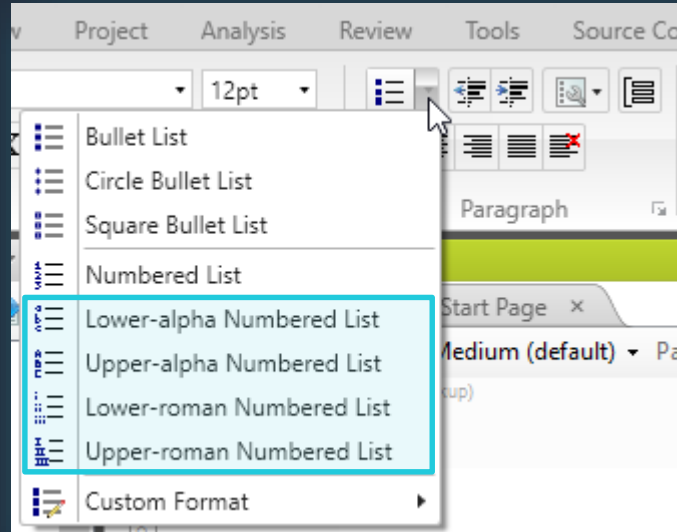
DEFAULT CSS FOR NESTED ORDERED LISTS

- All lists begin with “1”, regardless of level



DEFAULT CSS FOR NESTED ORDERED LISTS

- Could choose list formats on the Home ribbon to set the number/alpha formats for the nested lists



DEFAULT CSS FOR NESTED ORDERED LISTS

- But don't! That's inline formatting!

```
<ol>
  <li>First level.<ol style="list-style-type: lower-alpha;"><li>
Second level, with inline formatting.<ol style="list-style-type: upper-alpha;
"><li>Third level, with inline formatting.<ol style="list-style-type: lower-
roman;"><li>Fourth level, with inline formatting.<ol style="list-style-type:
upper-roman;"><li>Fifth level, with inline formatting.</li></ol></li></ol></
li></ol></li></ol></li>
  </ol>
```

CUSTOM CSS FOR NESTED ORDERED LISTS

- Achieved with **complex selectors**
- Sets the **list-style-type** attribute
- Requires no or very little author intervention when writing
 - None for OLs nested within a OL
 - A little for OLs nested within a UL

```
/* styles for controlling OL sublists within OLs
or ULs via complex selectors*/

ol ol,
ul ol
{
    list-style-type: lower-alpha;
}

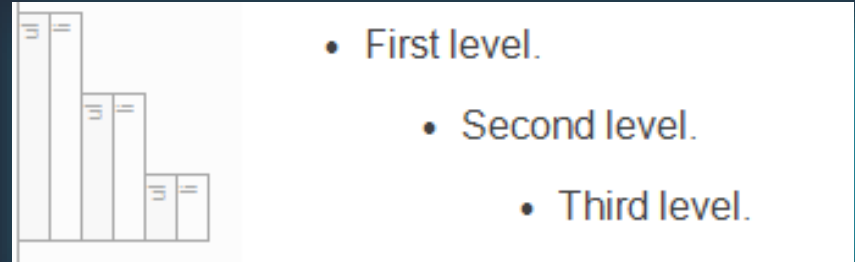
ol ol ol,
ul ol ol
{
    list-style-type: upper-alpha;
}

ol ol ol ol,
ul ol ol ol
{
    list-style-type: lower-roman;
}

ol ol ol ol ol,
ul ol ol ol ol
{
    list-style-type: upper-roman;
}
```

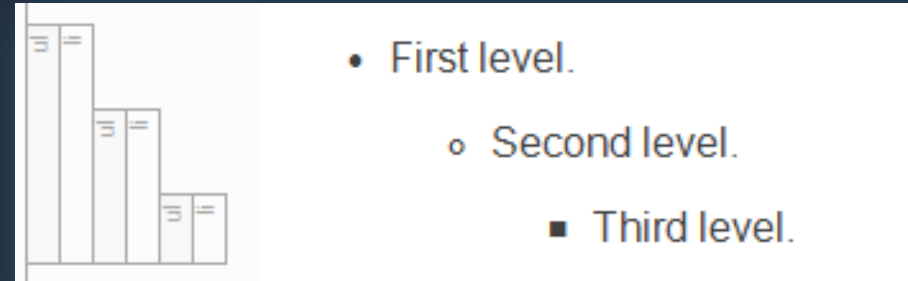
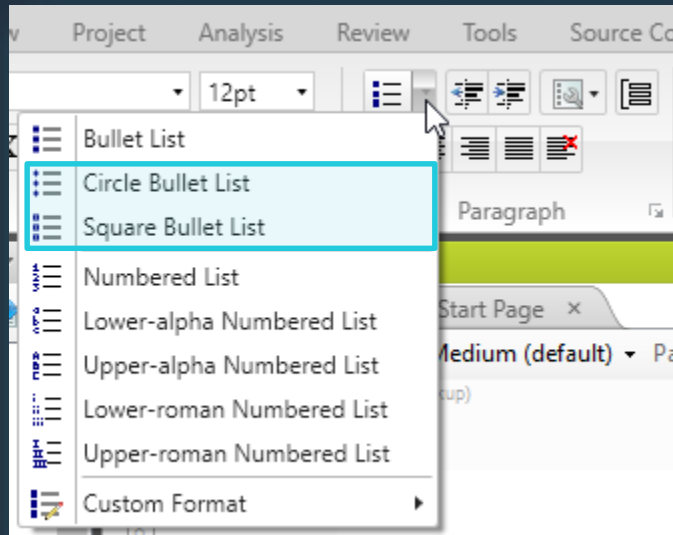
DEFAULT CSS FOR NESTED UNORDERED LISTS

- All lists begin with a solid bullet, regardless of level



DEFAULT CSS FOR NESTED UNORDERED LISTS

- Could choose list formats on the Home ribbon to set the style of bullet for the nested lists



DEFAULT CSS FOR NESTED UNORDERED LISTS

- But don't! That's inline formatting!

```
<ul>
  <li>First level.<ul style="list-style-type: circle;"><
li>Second level.<ul style="list-style-type: square;"><li>Third
level.</li></ul></li></ul></li>
</ul>
```

CUSTOM CSS FOR NESTED UNORDERED LISTS

- Achieved with **complex selectors**
- Sets the **list-style-type** attribute
- Requires no or very little author intervention when writing

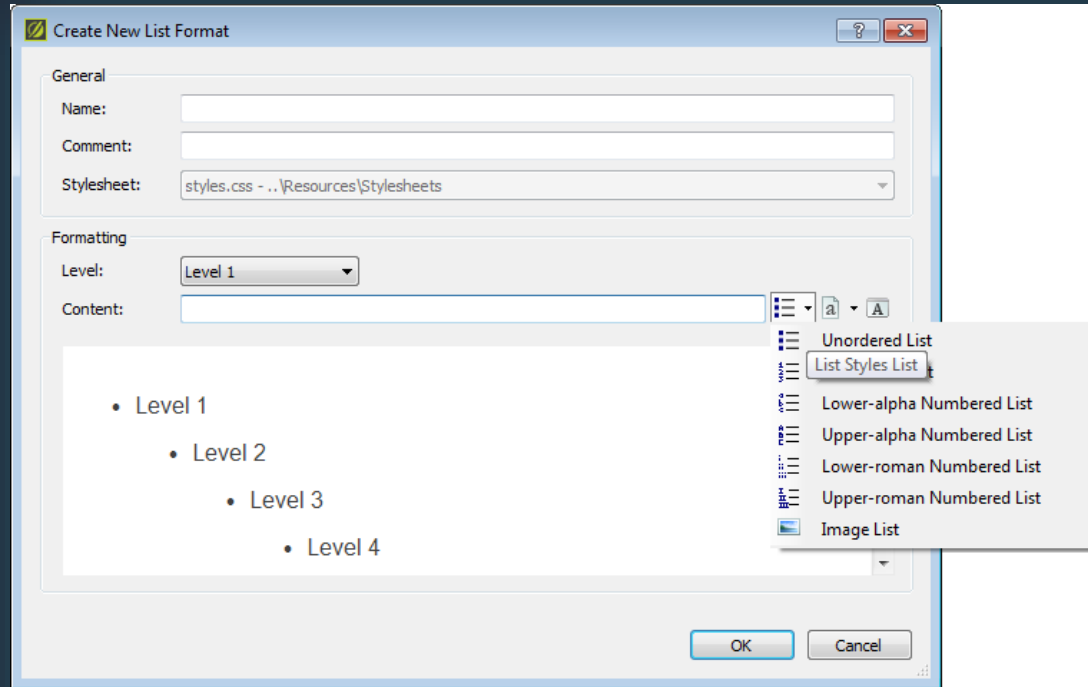
```
/* UL sublists within OLs or ULs */  
  
ol ul,  
ul ul  
{  
    list-style-type: circle;  
}  
  
ol ul ul,  
ul ul ul  
{  
    list-style-type: square;  
}
```



Custom OL and UL lists

CUSTOM LIST FORMATS

- Based on CSS3 marker elements and, per MadCap, not yet supported in HTML5 output
- Still a good way to jump-start crafting the CSS for a custom list



MIL-SPEC-STYLE ORDERED LISTS

- Used the Custom List dialog to generate CSS for the 5 levels
- Then played – a lot – in the stylesheet

```
/* Level 1 */
```

```
ol.MilSpecList
```

```
{
```

```
list-
```

```
list-
```

```
margi
```

```
text-
```

```
}
```

```
ol.MilSpe
```

```
{
```

```
list-
```

```
margi
```

```
margi
```

```
text-
```

```
}
```

```
ol.MilSpe
```

```
{
```

```
font-
```

```
font-
```

```
conte
```

```
}
```

```
/* Level 2 */
```

```
ol.MilSpecList > li > ol
```

```
{
```

```
/* Level 5 */
```

```
ol.MilSpecList > li > ol > li > ol > li > ol >
```

```
{
```

```
list-style-type: decimal;
```

```
list-style-image: none;
```

```
margin-left: 0px;
```

```
text-indent: 85px;
```

```
}
```

```
ol.MilSpecList > li > ol > li > ol > li > ol >
```

```
{
```

```
}
```

```
ol.MilSpecList > li > ol > li > ol > li > ol >
```

```
{
```

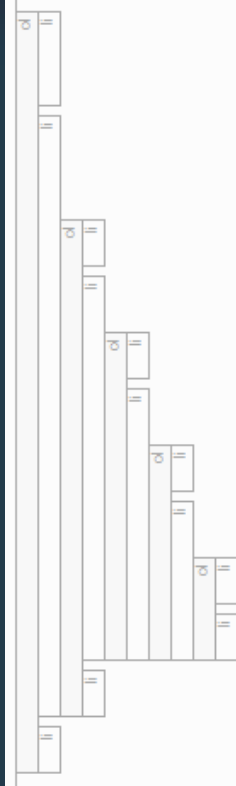
```
font-weight: bold;
```

```
font-style: normal;
```

```
content: '[' counter(MilSpecList) ' '];
```

MIL-SPEC-STYLE ORDERED LISTS

- In the XML Editor



a. This is the first step at the first level. This is the first step. This is the first step. This is the first step. This is the first step. This is the first step. This is the first step. This is the first step.

b. This is the second step. This is the second step. This is the second step. This is the second step. This is the second step. This is the second step. This is the second step.

1. This is a second-level list item.

2. This is a second-level list item.

(a) This is a third-level list item.

(b) This is a third-level list item.

(1) This is a fourth-level list item.

(2) This is a fourth-level list item.

[1] This is a fifth-level list item.

[2] is is a fifth-level list item.

3. Back to the second-level.

c. Back to the first-level.

MIL-SPEC-STYLE ORDERED LISTS

- In PDF output

What follows is a custom list that mimics "mil spec-like" ordered lists with unusual numbering and indentation.

a. This is the first step at the first level.
This is the first step. This is the first step.
This is the first step. This is the first step.
This is the first step. This is the first step.
This is the first step.

b. This is the second step. This is the second step. This is the second step. This is the second step. This is the second step. This is the second step. This is the second step.

1. This is a second-level list item.

2. This is a second-level list item.

(a) This is a third-level list item.

(b) This is a third-level list item.

(1) This is a fourth-level list item.

(2) This is a fourth-level list item.

[1] This is a fifth-level list item.

[2] is a fifth-level list item.

3. Back to the second-level.

c. Back to the first-level.

PARTING ADVICE ABOUT CUSTOM LISTS

- Use Flare's Custom List functionality to generate the CSS for all the levels you need
- Then experiment in the stylesheet directly and add more finesse
- Test, test, test

**We'll swing back to custom lists a little later...*



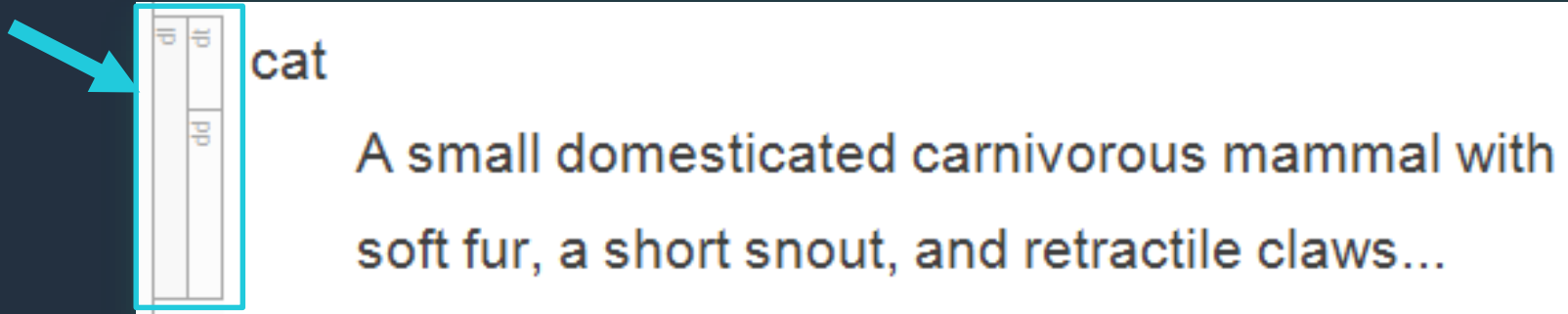
Definition lists

DEFINITION LISTS*

*aka description lists

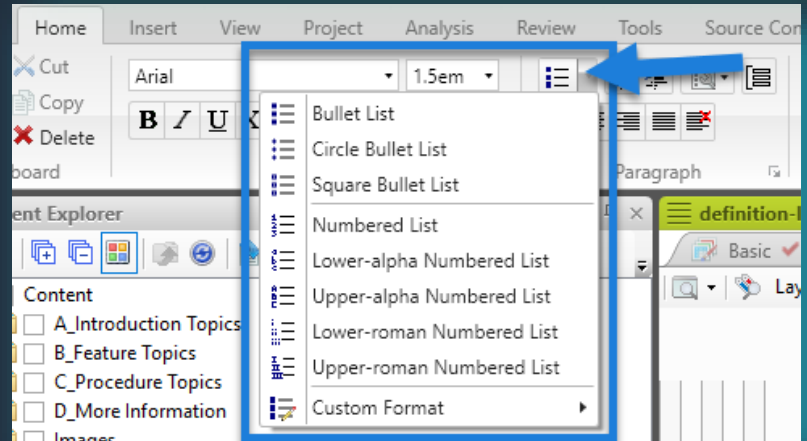
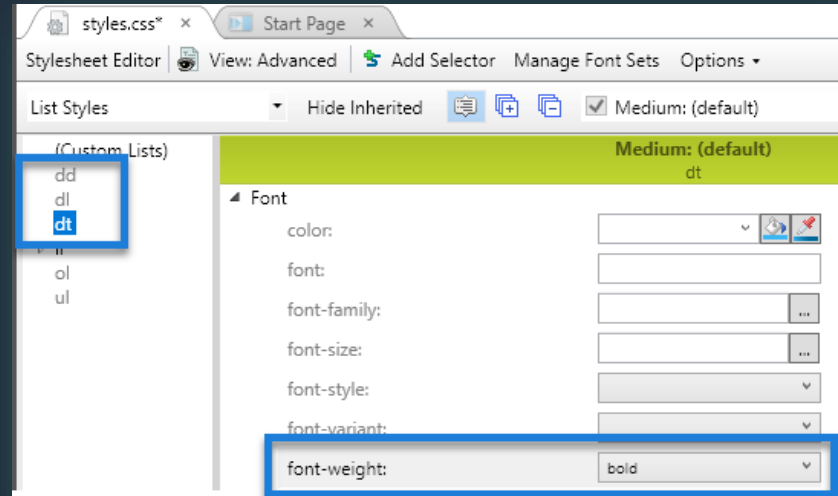
- An HTML structure
 - `<dl>` - the list
 - `<dt>` - the term
 - `<dd>` - the definition
 - Can contain other tags

```
<dl>  
  <dt>cat</dt>  
  <dd>A small domesticated  
carnivorous mammal with soft fur,  
a short snout, and retractile  
claws... </dd>  
</dl>
```



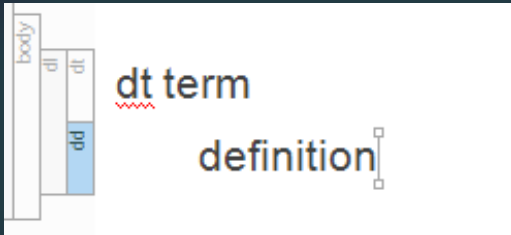
HOW TO ACHIEVE IN FLARE

- Can customize the CSS in the Stylesheet Editor
- Can't insert DLs with Flare's GUI interface
- Must code by hand (or use trick on the next slide)

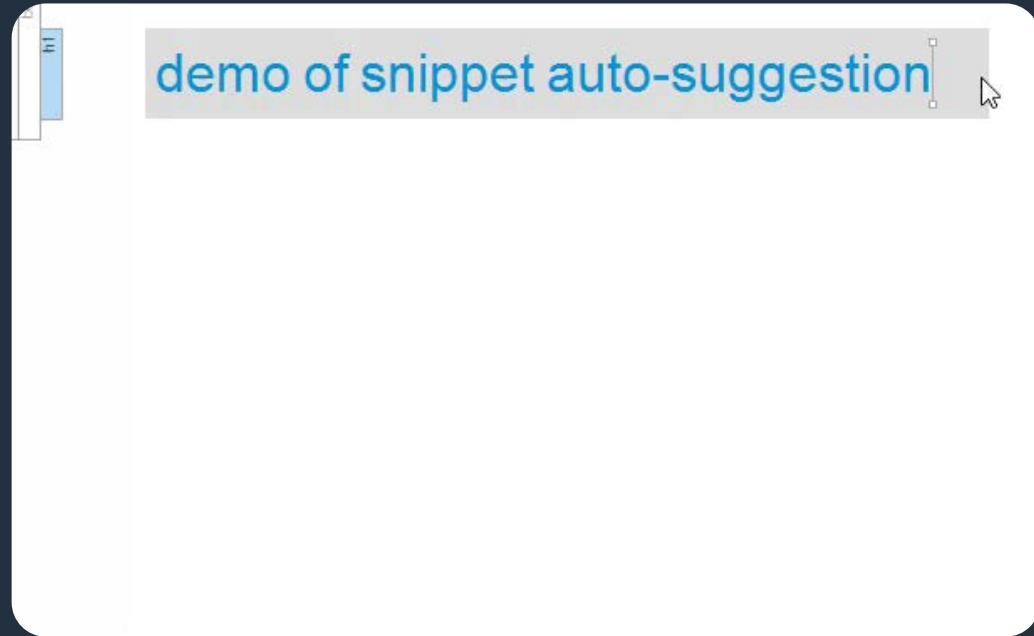


TIPS FOR QUICKLY INSERTING DEFINITION LISTS

- In a snippet, code a DL by hand, then save with placeholder content



- Turn on snippet auto-suggestions
- Insert, convert to text, and edit



COMMON USES FOR DEFINITION LISTS

- Glossaries of terms
 - An alternative to Flare's glossary functionality
- API documentation
 - List of programming parameters / arguments

Individual Bit Instructions

It is often useful to consider a 32-bit or 64-bit register as 32 or 64 individual bits and to perform instructions simultaneously on each of the bits of two sources.

Syntax

Description

Examples

Description

The b1 form is used with control (c) register sources. It can only be used with the instructions and, or, xor, and not.

and

Performs the bitwise AND operation on the two sources *src0* and *src1* and places the result in the destination *dest*. The and instruction can be applied to 1-, 32-, and 64-bit values.

or

Performs the bitwise OR operation on the two sources *src0* and *src1* and places the result in the destination *dest*. The or

TABLES VS DEFINITION LISTS

Parameter	Definition
NaN	Not A Number. Only supported for floating point source operand types. Returns true if either floating-point source operand is a NaN.
<u>num</u>	Numeric. Only supported for floating point source operand types. Returns true if both floating-point source operands are numeric values (not a NaN).
<u>eq</u> , <u>ne</u> , <u>lt</u> , <u>le</u> , <u>gt</u> , <u>ge</u>	Ordered comparisons. These correspond to <i>equal</i> , <i>not equal</i> , <i>less than</i> , <i>less than or equal</i> , <i>greater than</i> and <i>greater than or equal</i> respectively. All support both integer and floating point source operand types. Additionally, <u>eq</u> and <u>ne</u> support the <u>b1</u> bit source operand type. For floating-point source operands, if either is a NaN, then the result is false. Otherwise, returns the corresponding comparison performed on the source operands.
<u>equ</u> , <u>neu</u> , <u>ltu</u> , <u>leu</u> , <u>gtu</u> , <u>geu</u>	Unordered comparisons. There are unordered forms of all the ordered comparisons. For example, <u>leu</u> is the unordered form of <u>le</u> . Only supported for floating point source operand types. If either operand is a NaN, then the result is true. Otherwise, returns the same result as the corresponding ordered comparison.

NaN

Not A Number. Only supported for floating point source operand types. Returns true if either floating-point source operand is a NaN.

num

Numeric. Only supported for floating point source operand types. Returns true if both floating-point source operands are numeric values (not a NaN).

eq, ne, lt, le, gt, ge

Ordered comparisons. These correspond to *equal*, *not equal*, *less than*, *less than or equal*, *greater than* and *greater than or equal* respectively. All support both integer and floating point source operand types. Additionally, eq and ne support the b1 bit source operand type. For floating-point source operands, if either is a NaN, then the result is false. Otherwise, returns the corresponding comparison performed on the source operands.

equ, neu, ltu, leu, gtu, geu

Unordered comparisons. There are unordered forms of all the ordered comparisons. For example, leu is the unordered form of le. Only supported for floating point source operand types. If either operand is a NaN, then the result is true. Otherwise, returns the same result as the corresponding ordered comparison.

LEARN MORE ABOUT DEFINITION LISTS

- MDN web docs
 - <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/dl>
- w3schools.org
 - https://www.w3schools.com/tags/tag_dl.asp



Paragraph classes that act like
OLs



PARAGRAPH CLASSES THAT ACT LIKE OLS

- Disadvantage
 - Author has to select a lot of styles while writing
- Advantage
 - Have total control over the numbering scheme
 - Can point xrefs to bookmarked auto-numbering paragraphs

PARAGRAPH CLASSES THAT ACT LIKE OLS

```
p.Step,  
p.StepReset  
{  
    mc-auto-number-position: outside-head;  
    mc-auto-number-offset: 62px;  
    margin-left: 62px;  
    mc-auto-number-format: 'Step {n+}. '  
}
```


```
p.StepReset  
{  
    mc-auto-number-format: 'Step {n=1}. '  
}
```

```
p.Substep,  
p.SubstepReset  
{  
    mc-auto-number-position: outside-head;  
    mc-auto-number-offset: 25px;  
    margin-left: 85px;  
    mc-auto-number-format: '{a+}. '  
}
```

```
p.SubstepReset  
{  
    mc-auto-number-format: '{a=1}. '  
}  
  
.ElementWithinStep  
{  
    margin-left: 62px;  
}  
  
.ElementWithinSubstep  
{  
    margin-left: 85px;  
}
```

PARAGRAPH CLASSES THAT ACT LIKE OLS

Step 1. (p.StepReset)

Step 2.  bookmark (p.Step).
(p.ElementWithinStep)

Step 3. (p.Step)

a. (p.SubstepReset)

b. (p.Substep) See [Step 2](#).
(p.ElementWithinSubstep)

Step 1. (p.StepReset)

Step 2. (p.Step).
(p.ElementWithinStep)

Step 3. (p.Step)

a. (p.SubstepReset)

b. (p.Substep) See [Step 2](#).
(p.ElementWithinSubstep)

SHOUT OUT TO...

- Dee Vincent-Day's guest Madblog posts
- Oodles of advice for custom auto-numbering p classes
 - [Auto-numbering in MadCap Flare: Part 1 – How to Create Legal-Style Numbering](#)
 - [Auto-numbering in MadCap Flare: Part 2 – How to Create Custom Numbering Sequences](#)



Tables with auto-numbered rows

Two examples

EXAMPLE 1: AUTO-NUMBERING TABLE ROWS

- Ingredients
 - Custom p classes with auto-numbering
 - Table stylesheet
 - “Starter” snippet to insert, convert, and edit

Step	Action	Result
1.	Here is an action	And here is the result.
2.	And here is the next thing to do.	And the result for this step.
3.	And on...	
4.	And on...	
5.	And on...	
6.		

Step		Action	Result
1.			
2.	a.	This is the first sub-step to step 2.	And then this too will happen.
	b.	Make this one the second sub-step.	
	c.	Do this other thing, too.	
	d.	Do this other thing next.	And then this too will happen.
3.	a.	Do this something or other.	This will happen.
	b.	Do this something or other.	This will happen.
	c.	Do this something or other.	This will happen.
4.			
5.			

EXAMPLE 1: AUTO-NUMBERING TABLE ROWS

- CSS
 - p classes with auto-numbers

```
p.Table1_StepNumber /* For having table rows be auto-numbered as
steps */
{
    mc-auto-number-format: 'N:{n+}.';
}

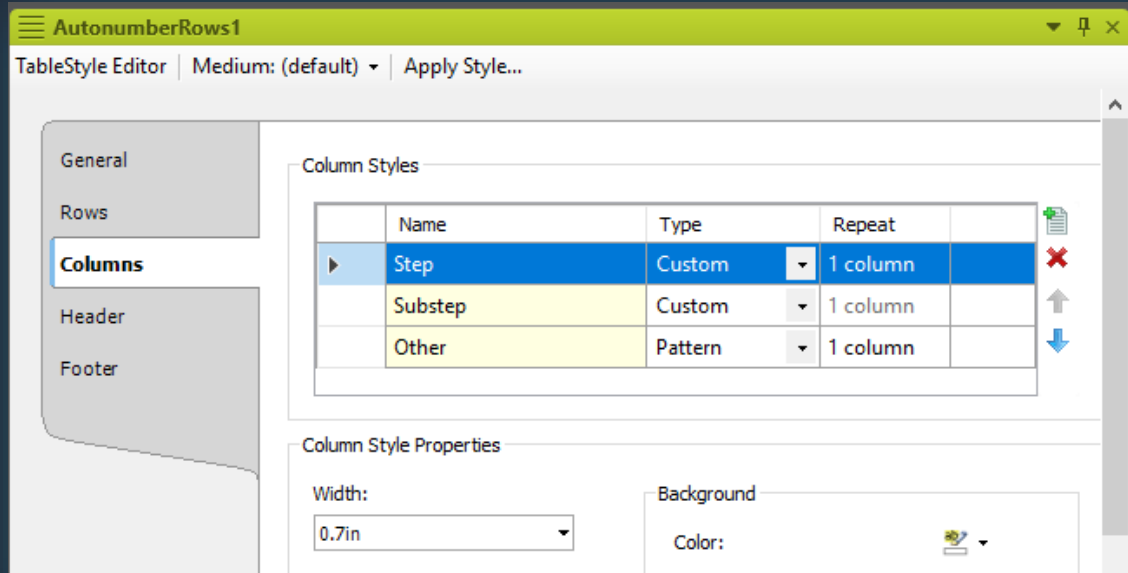
p.Table1_StepNumberReset /* For resetting a table row to step 1 */
{
    mc-auto-number-format: 'N:{n=1}.';
}

p.Table1_SubstepLetter /* For having table rows be auto-numbered as
substeps */
{
    mc-auto-number-format: 'L:{a+}.';
}

p.Table1_SubstepLetterReset /* For resetting a table row to substep a
*/
{
    mc-auto-number-format: 'L:{a=1}.';
}
```

EXAMPLE 1: AUTO-NUMBERING TABLE ROWS

- Table stylesheet
- Primarily controls borders, background colors, and widths
- Columns for steps, substeps, others



EXAMPLE 1: AUTO-NUMBERING TABLE ROWS

- First-time setup
 - Insert the table, apply the table stylesheet, start populating with content
 - For major steps
 - Set the first column to Step column
 - Set first row's first column's cell content style to `p.Table1_StepNumberReset`
 - Set all other rows' first columns' cell content style to `p.Table1_StepNumber`
 - For minor steps (substeps)
 - Likewise, select Substep columns and p classes for substeps

EXAMPLE 1: AUTO-NUMBERING TABLE ROWS

- Tips
 - Create CSS for cross-referencing to step numbers, substep letters
 - Save an already formatted table a “starter snippet”

table		
	tbody	thead
		tr
		tr
		tr
		tr

Step	Action	Result
1.	replace	replace
2.	replace	replace
3.	replace	replace

EXAMPLE 2: AUTO-NUMBERING TABLE ROWS

- Similar idea, just different numbering scheme
- Table stylesheet, CSS in the demo project

Step	Action	Result
1.0	Here is an action	And here is the result.
2.0	And here is the next thing to do.	And there result for it.
2.1	And on...	
3.0	And on...	
3.1	And on...	
3.2	And on...	



Cross-references to “step” LIs in OLs

CSS TRICKS FOR XREFS TO LISTS ITEMS

- Can't point a cross-reference to a standard numbered list item
- Craft a custom OL list with `mc-auto-number-format` in lieu of default numbering*
- Craft custom xref format

**I haven't crafted CSS for Ps within LIs or for nested lists*

```
ol.StepNum
{
}

ol.StepNum > li
{
    list-style-type: none;
    mc-auto-number-format: 'X:{n+}. ';
    mc-auto-number-position: outside-head;
    mc-auto-number-offset: 40px;
    mc-auto-number-class: StepNum;
}

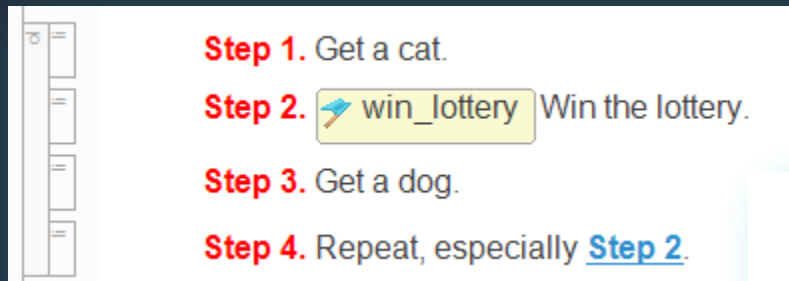
ol.StepNum > li:first-child
{
    mc-auto-number-format: 'X:{n=1}. ';
}

span.StepNum
{
    color: red;
    font-weight: bold;
}

MadCap|xref.Step
{
    mc-format: 'step {paranumonly}';
}
```

CRAFTING ONE OF THESE LISTS

- Select the custom list and enter your content*
- For any step you will point an xref to, add a bookmark
- Insert the xref



**I haven't crafted CSS for Ps within LIs or for nested lists*

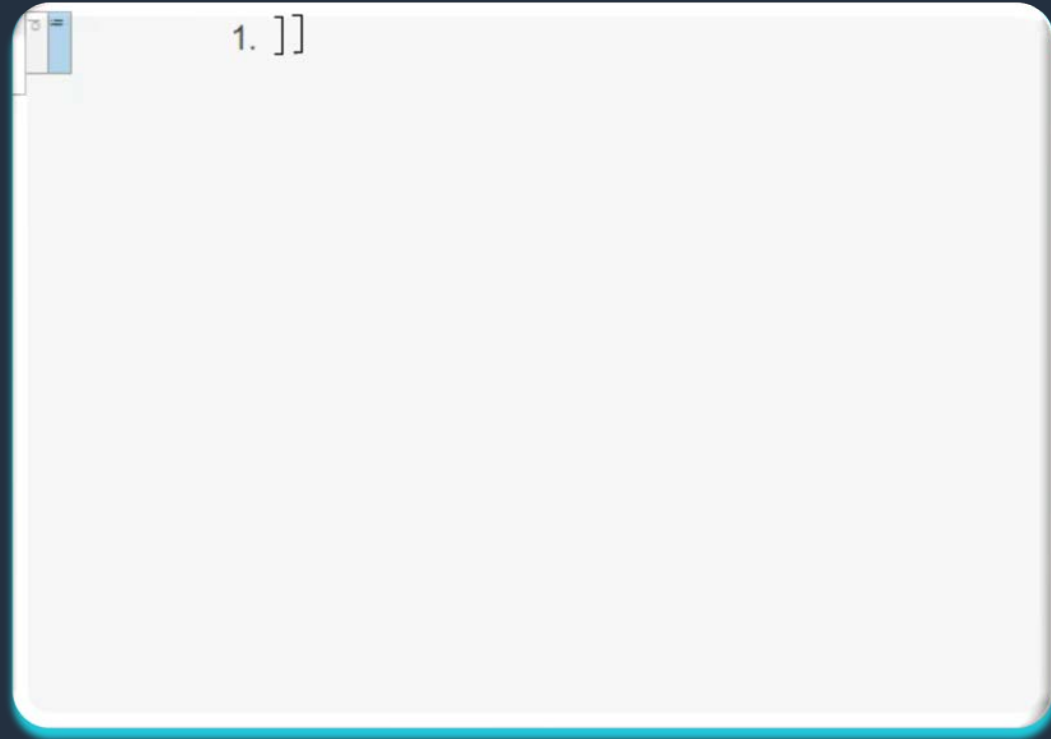
- Step 1.** Get a cat.
- Step 2.** Win the lottery.
- Step 3.** Get a dog.
- Step 4.** Repeat, especially [Step 2](#).



Flare UI tricks for quickly writing OL and UL lists

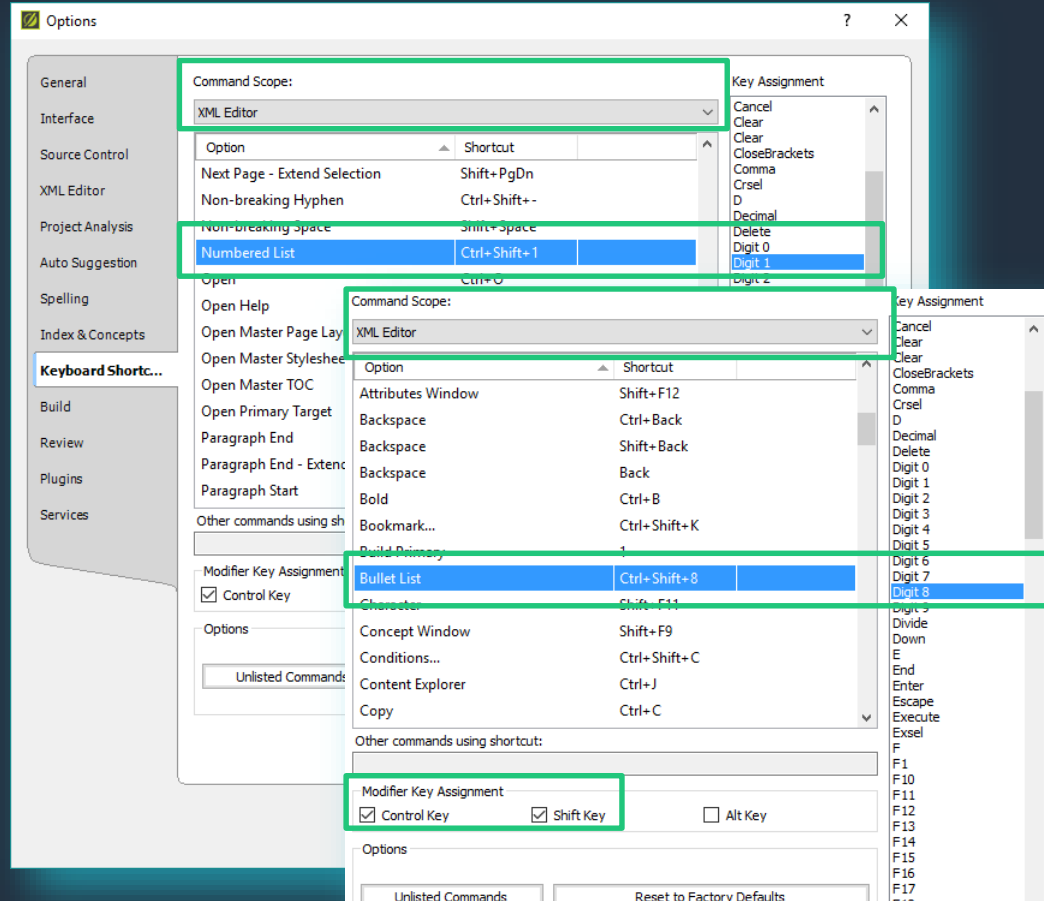
BUILT-IN KEYBOARD SHORTCUTS

- Created next block = **Enter**
- Indent a list item = **Tab**
- Nest a paragraph within a list item = **Ctrl + ;**
- Outdent (move left) a list item or nested paragraph = **Shift + Tab**



CUSTOM KEYBOARD SHORTCUTS

- Custom keyboard shortcuts for the XML Editor
 - **Ctrl + Shift + 1** (“Digit 1”) for a Numbered List (OL)
 - **Ctrl + Shift + 8** (“Digit 8”) for a Bullet List (UL)

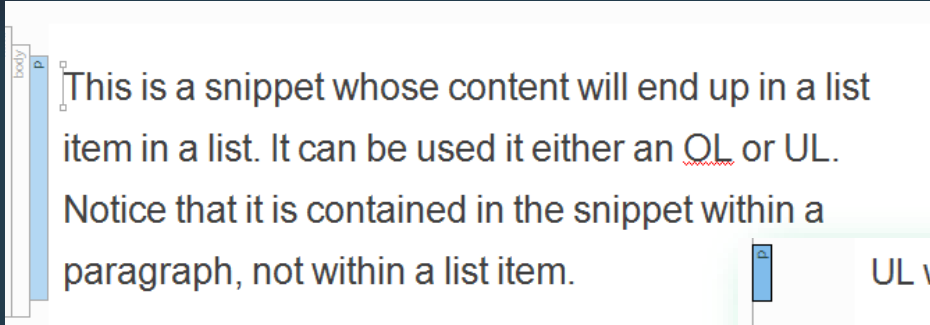




List item snippets

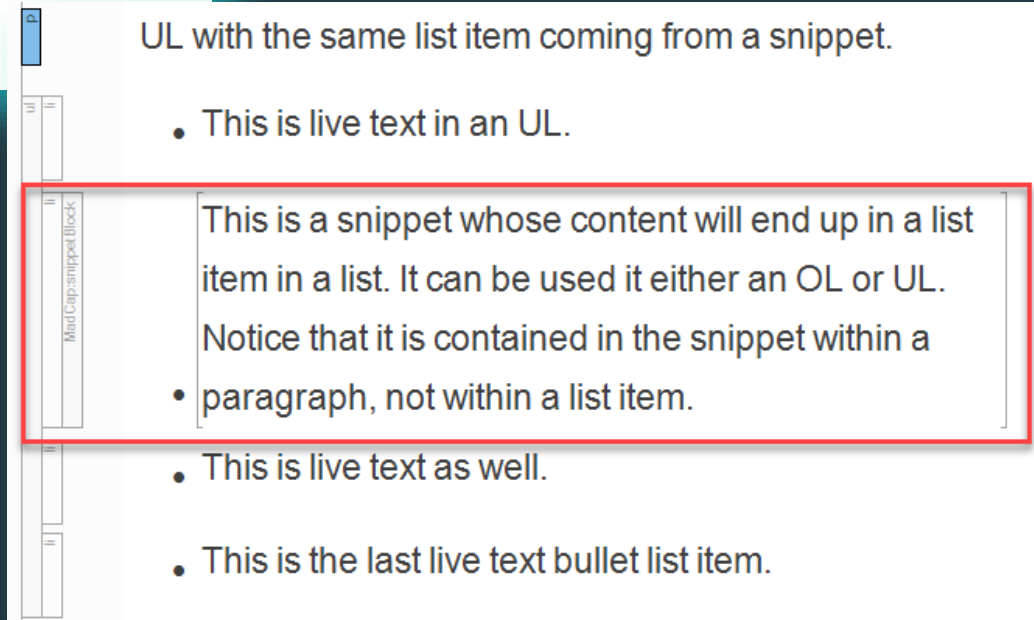
LIST ITEM SNIPPETS – SINGLE STEP

- Content resides in a `<p>`



This is a snippet whose content will end up in a list item in a list. It can be used it either an OL or UL. Notice that it is contained in the snippet within a paragraph, not within a list item.

The diagram shows a vertical sidebar on the left with a blue bar labeled 'body' and a small 'p' icon. The main text area contains the paragraph above.



UL with the same list item coming from a snippet.

- This is live text in an UL.

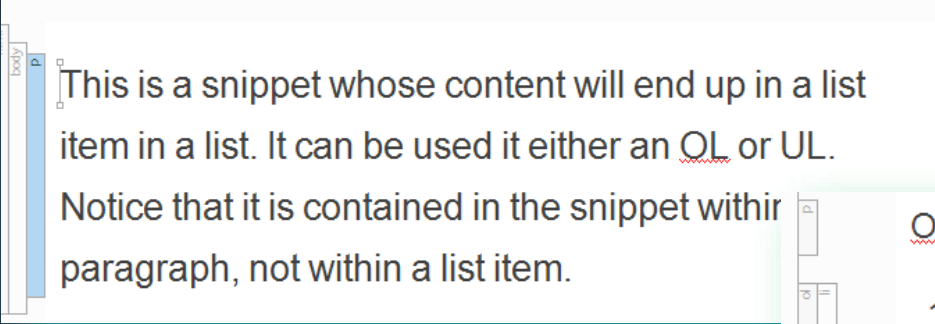
This is a snippet whose content will end up in a list item in a list. It can be used it either an OL or UL. Notice that it is contained in the snippet within a

- paragraph, not within a list item.

- This is live text as well.
- This is the last live text bullet list item.

The diagram shows a vertical sidebar on the left with a blue bar labeled 'p' and a 'MiniCap: snippet Block' label. The main text area contains the list item and the paragraph snippet, which is highlighted with a red border.

LIST ITEM SNIPPETS – SINGLE STEP, AS A <P>




This is a snippet whose content will end up in a list item in a list. It can be used it either an OL or UL.

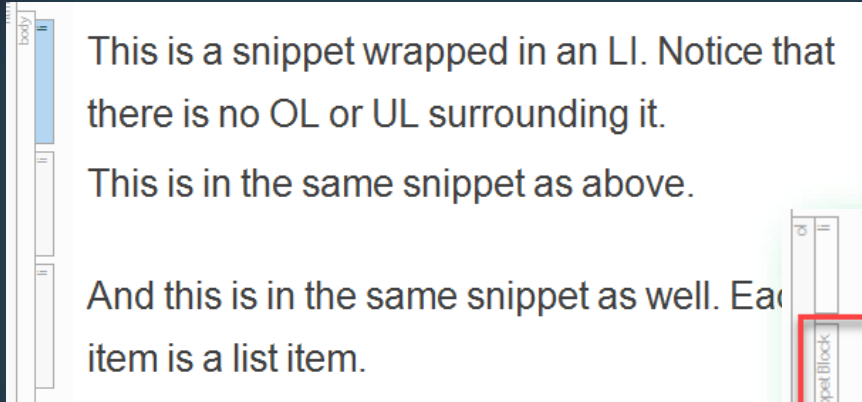
Notice that it is contained in the snippet withir paragraph, not within a list item.

- Insert the snippet as a snippet block or snippet text

OL with one list item coming from a snippet.

- 
1. This is live text in an OL.
 2. This is live text too.
 3. [This is a snippet whose content will end up in a list item in a list. It can be used it either an OL or UL. Notice that it is contained in the snippet within a paragraph, not within a list item.]
This is live text under the snippet list item.
 4. This is the last live text numbered list item.

LIST ITEM SNIPPETS – MULTIPLE BITS OF CONTENT



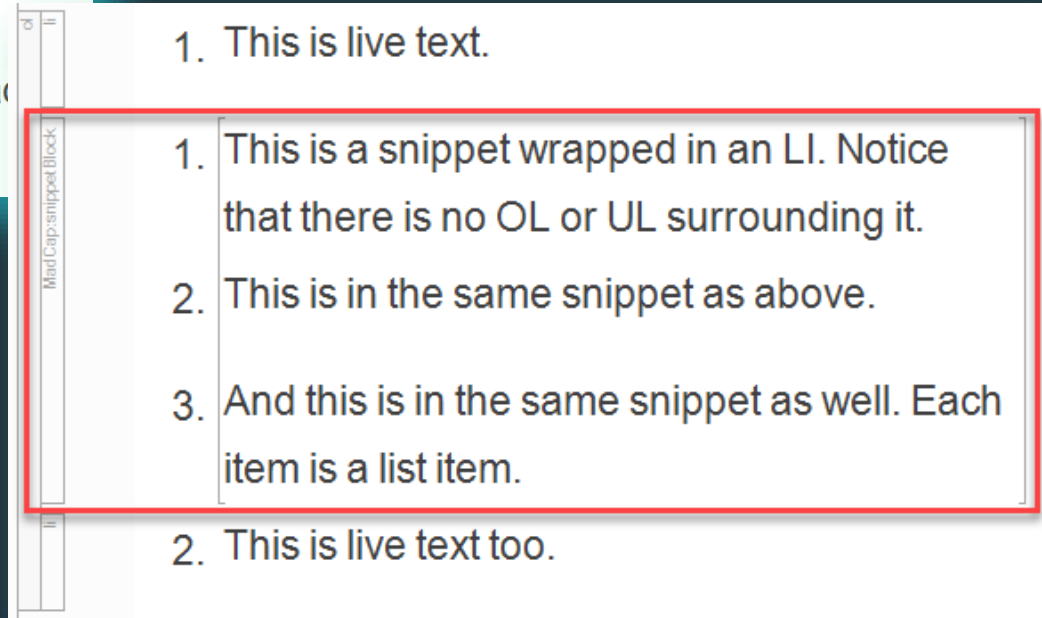
This is a snippet wrapped in an LI. Notice that there is no OL or UL surrounding it.

This is in the same snippet as above.

And this is in the same snippet as well. Each item is a list item.

The screenshot shows a document editor with a sidebar on the left. The sidebar has a 'body' section highlighted in blue. The main content area shows three paragraphs of text. The first paragraph is 'This is a snippet wrapped in an LI. Notice that there is no OL or UL surrounding it.' The second paragraph is 'This is in the same snippet as above.' The third paragraph is 'And this is in the same snippet as well. Each item is a list item.'

- Content resides in ``s, outside on a list



1. This is live text.

1. This is a snippet wrapped in an LI. Notice that there is no OL or UL surrounding it.

2. This is in the same snippet as above.

3. And this is in the same snippet as well. Each item is a list item.

2. This is live text too.

The screenshot shows a document editor with a sidebar on the left. The sidebar has a 'MadCap/snipet Block' section highlighted in blue. The main content area shows a list of items. The first item is '1. This is live text.' The second item is '1. This is a snippet wrapped in an LI. Notice that there is no OL or UL surrounding it.' The third item is '2. This is in the same snippet as above.' The fourth item is '3. And this is in the same snippet as well. Each item is a list item.' The fifth item is '2. This is live text too.'

LIST ITEM SNIPPETS – MULTIPLE BITS OF CONTENT

1. This is live text.
2. This is a snippet wrapped in an LI. Notice that there is no OL or UL surrounding it.
3. This is in the same snippet as above.
4. And this is in the same snippet as well. Each item is a list item.
5. This is live text too.



Final questions?

DEMO PROJECT



ListapaloozaDemo.flprjzip

Hmm... looks like this file doesn't have a
preview we can show you.

[Download](#)

Catch up with me in the Flare Slack community,
the MadCap user forums, and
the Flare group on LinkedIn.

beck-communications.com
nbeck@beck-communications.com

MW