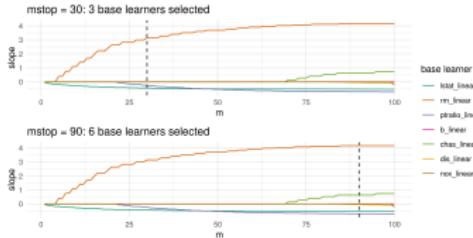


# Introduction to Debugging Lectures

## Demo Lecture

### Layout Macro Testing With Cannibalized Content



### Learning goals

- Test layout
- Fix stuff
- Go home

# DEMO SLIDES

This is a demo lecture chunk.

It is recommended to view these slides as PDF and LaTeX code side by side

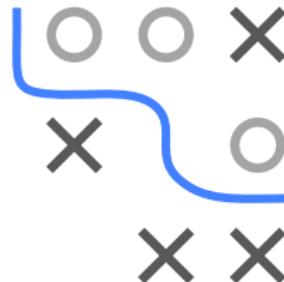
Refer to the `slds-lmu/lecture_debug` repository for the source and PDF:

- ► Compiled PDF
- ► LaTeX source



## \SPLITV(TCB)(TCB)

- \splitVCC creates two centered columns
- \splitVTT creates two top-aligned columns
- \splitVBB for bottom-aligned columns



**Left column** these two columns should both be top-aligned with their unequal content lengths

**Right column** Lorem ipsum dolor sit amet, consectetur adipiscing elit

- Example itemize content for centered columns
- Second itemize item

  Lorem ipsum dolor sit amet, consectetur adipiscing elit

## \SPLITV WITH UNEQUAL SIZES

You only need to specify the width of the first column:

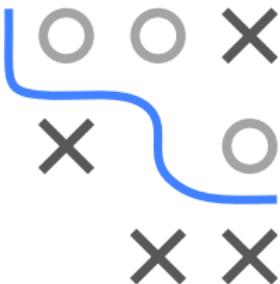
`\splitVTT[0.75]{left content}{right content}` creates a column with 75% width and the second column will fill the remaining space:

First column with 75% of the text width on the slide

Second column  
with rest

Tiny column

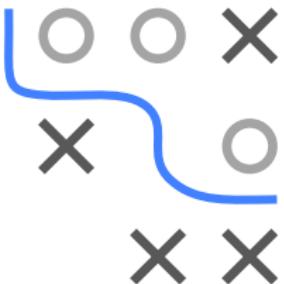
Second column with a lot of room for activities



## TWO COLUMNS, MINIMAL ADJUSTMENTS

Compact version if you do not want to use the full slide width:

\splitVCompact{0.2}{0.2} only takes up 20% of the slide width in each column for a total of 40% with a minimal horizontal spacer:



First column      Second column  
with 20%            with 20%

## 3 COLUMNS

Using `\splitVThree{Left}{Middle}{Right}` for 3 equally sized columns:

First column content  
is here

Second column con-  
tent is here as well

And also a third col-  
umn is here just in  
case



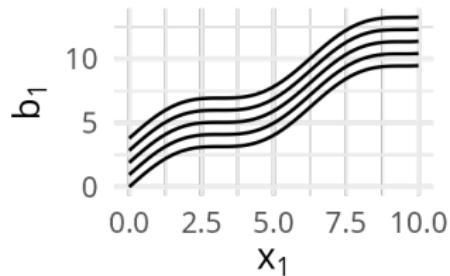
Then there is `\splitVThreeCustom` for three columns of arbitrary widths, where the width of the third column can also be inferred from the first two

first column with width	second column with 20% width	third column with remaining width 20% width
----------------------------	------------------------------------	--

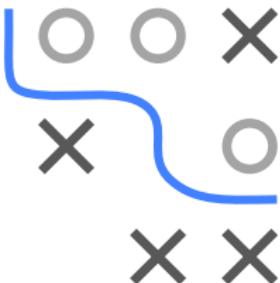
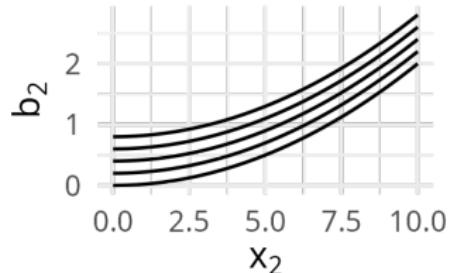
## TWOBYTWO 2×2 LAYOUT / QUADRANTS

Use `\twobytwo{top left}{top right}{bottom left}{bottom right}` for horizontally aligned quadrants

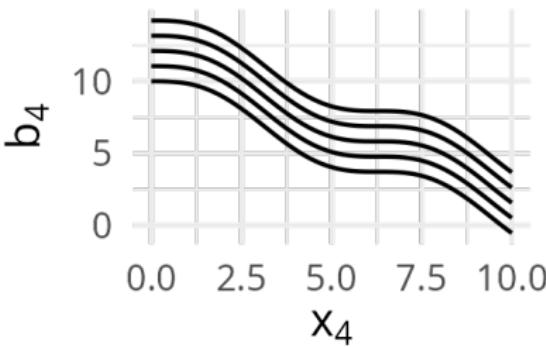
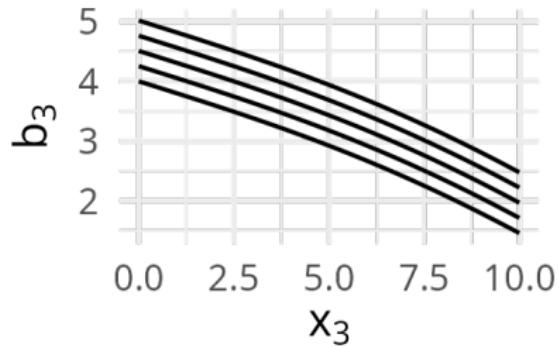
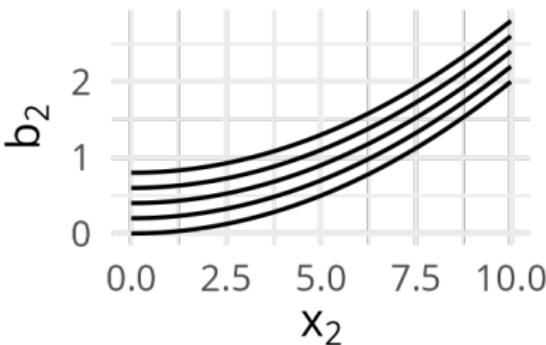
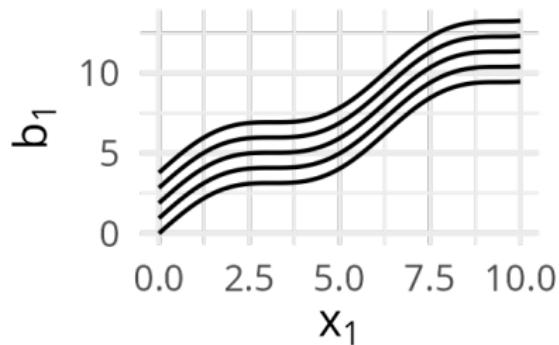
- Example top left...
- ...quadrant content
- Next to a figure



- Bottom left...
- ...quadrant content

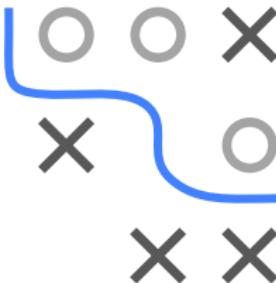


## **2×2 LAYOUT: ALL IMAGES**



# ITEMIZE WRAPPERS

Presets from `itemize` with different vertical spacings (`itemsep`). The default value for `itemsep` is apparently 2pt. Use `\the \itemsep` to find out the current value.



- Default `itemize`

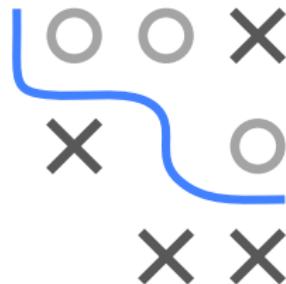
- `itemsep` is unmodified
- Another thing
- Words
- A fourth thing to show
- Just filling space here
- Hello there

- Uses environment

- `itemizeS`
- `itemsep` is: -2.0pt
- Another thing
- Words
- A fourth thing to show
- Just filling space here
- Hello there

# ITEMIZEM AND ITEMIZEL

- Default itemize with itemizeM
  - itemsep is unmodified
  - Another thing
  - Words
  - A fourth thing to show
  - Just filling space here
  - Hello there
- 
- Uses environment itemizeL
  - itemsep is: 15.0pt
  - Another thing
  - Words
  - A fourth thing to show
  - Just filling space here
  - Hello there



# ITEMIZEF FOR VERTICAL FILLING

- Uses environment `itemizeF`
- `itemsep` is: 10.07397pt plus 1.0fill
- Automatically uses all vertical space
- Depends on containing environment though!



# ITEMIZE SPACING AND FONT SIZE CONTROL

## itemizeS with small text

- Small font + compact spacing
- Use for detailed lists
- When every line matters

## itemizeL with large text:

- Large font + wide spacing
- Makes content stand out
- Ideal for takeaways

## itemizeM with default text:

- Default sizing and spacing works best for most content
- Provides good readability and balanced slide appearance

## itemizeM with footnotesize text:

- Default sizing and spacing works best for most content
- Provides good readability and balanced slide appearance



# REFERENCE BUTTONS

► Author 2025

- Use \furtherreading with citation key:
  - Example: ► Author 2025
  - Basically same as old citelink / citebutton
- Use \sourceref with URL:
  - Direct URL: ► Click for source
  - URL with www: ► Click for source
- Or use \sourceref with citation key:
  - Example: ► Click for source
  - Another example: ► Click for source
  - Uses the url field in bib file for link:



The buttons have different background colors:

- Light blue background for further reading
- Light orange/peach background for source references

# IMAGE TAKING THE FULL SLIDE WIDTH



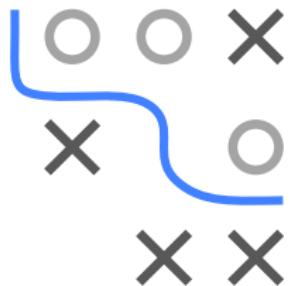
# IMAGE USING HALF THE AVAILABLE WIDTH



# CENTERED IMAGE USING HALF THE AVAILABLE WIDTH

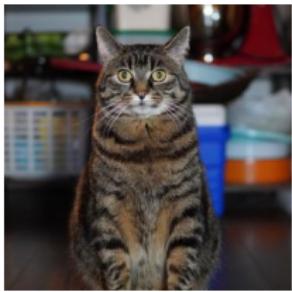


# FULL WIDTH CENTERED IMAGE WITH CITEKEY ATTRIBUTION

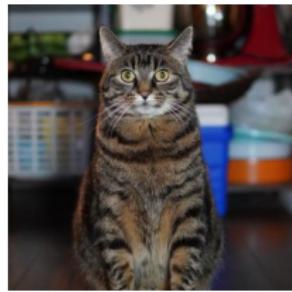


▶ Click for source

# IMAGES ATTRIBUTION CAN BE URL OR CITEKEY



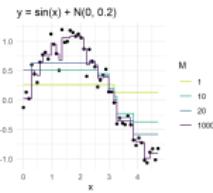
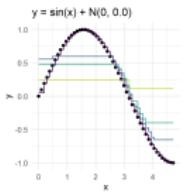
▶ Click for source



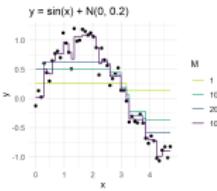
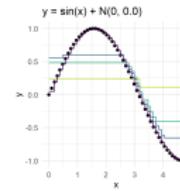
▶ Click for source



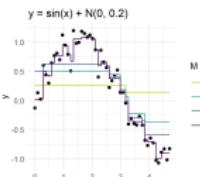
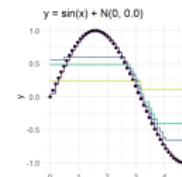
# IMAGES WITH ALIGNMENT AND ATTRIBUTION



▶ Click for source



▶ Click for source

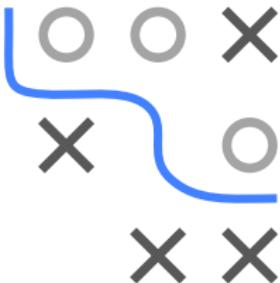


▶ Click for source

# IMAGEFIXED: ABSOLUTE POSITIONING

Use `\imageFixed{x}{y}[width][attribution]{file}` for absolute positioning:

- x and y are absolute coordinates from top-left
- Coordinates can be lengths like `2cm` or relative like `0.1\paperwidth`
- Width is optional (default 1.0 relative to `\textwidth`)
- Attribution works the same as other image macros
- Useful for overlaying images at precise positions



# IMAGEFIXED EXAMPLES

Top-left corner:

Center-right using paper dimensions:

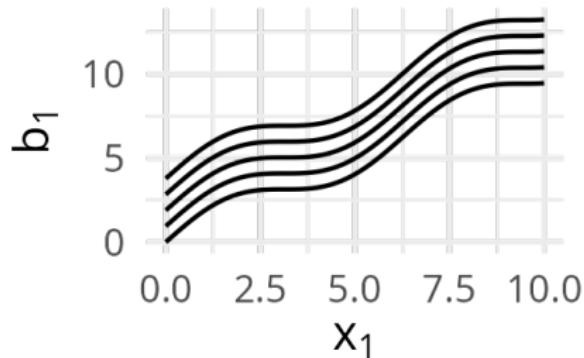
Bottom-left with attribution:



▶ Click for source

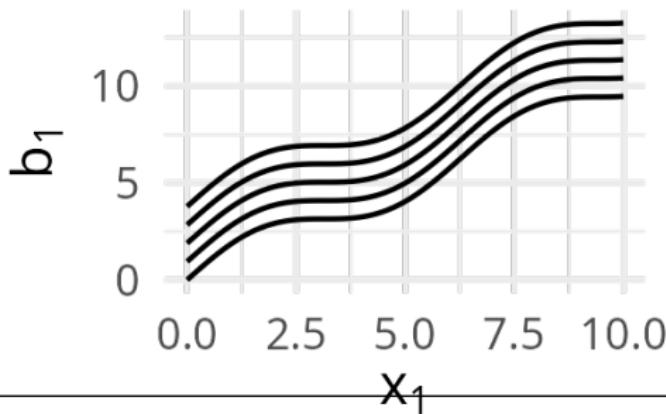
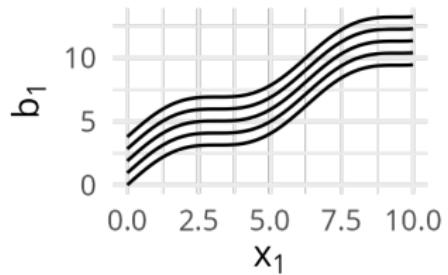
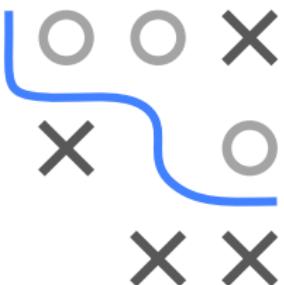
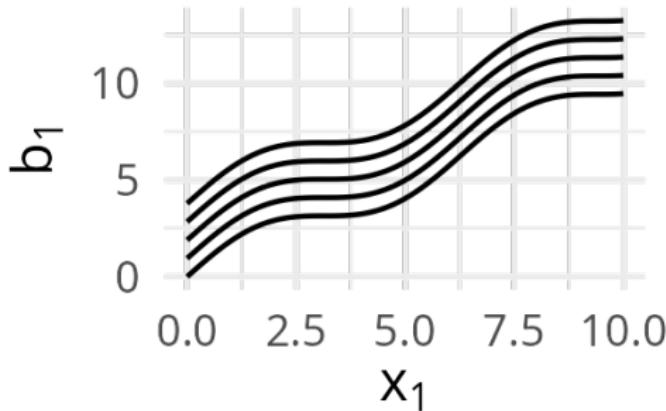
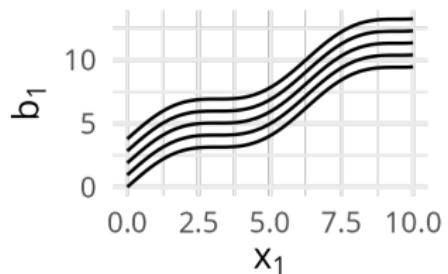
# IMAGES WITHIN ITEMIZE

- Foo bar
- Plubber

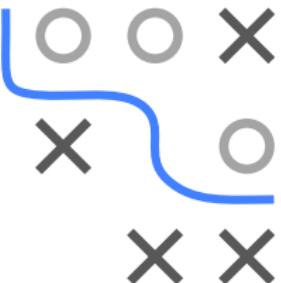


- jfowiehfgnsdlkjnfg

# IMAGES WITHIN SPLITV



# IMAGES WITHIN SPLITVCC



▶ Click for source



# FRAMEI ENVIRONMENT

- This is a frame consisting only of an itemize environment
- Nothing else here, just itemize.
- Default center alignment

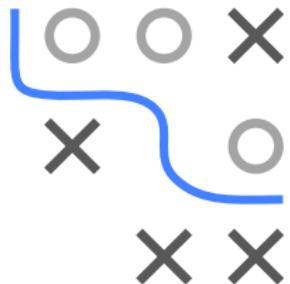


- Look, an image



# FRAMEI ENVIRONMENT

- This is a frame consisting only of an itemize environment
- nested items
- Look, an image



# FRAMEI ENVIRONMENT

- This is a frame consisting only of an itemize environment



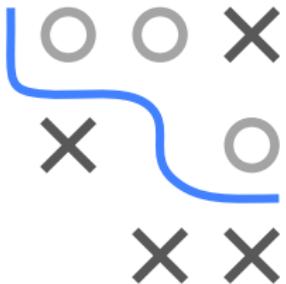
- Look, an image

- nested items



# CUSTOMIZED FRAMEI ENVIRONMENT

- The framei environment supports key-value arguments
- Font size: `fs=small` (or `footnotesize`, `large`, etc.)
- Separation: `sep=S` (or `M`, `L`, `F`)
- This example uses `framei[fs=small, sep=S]`
- Perfect for overview slides or detailed lists
- Makes it easy to fit more content on a single slide
- Please don't overdo it



## ANOTHER FRAMEI EXAMPLE

- Larger text with wider spacing
- Passed directly to the `framei` environment
- Using `[fs=large,sep=L]`
- For emphasis and impact



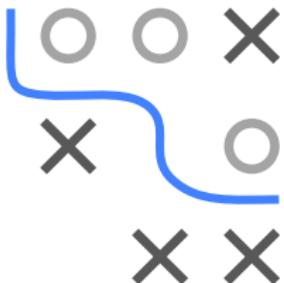
# A FILLING FRAMEI EXAMPLE

- Larger text with widest spacing
- Passed directly to the `framei` environment
- Using `[fs=large,sep=F]`
- For the glory of hypnotoad

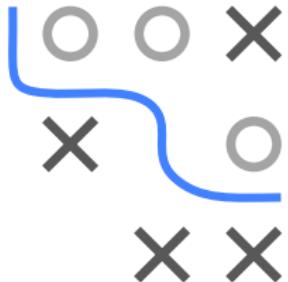


## FRAMEI WITH TOP ALIGNMENT

- The align=top option aligns content to the top of the frame
- Useful for consistency when not filling the entire frame
- Pairs well with fs and sep options
- Example: framei[align=top]



# FRAMEI WITH BOTTOM ALIGNMENT



- Bottom alignment with `align=bottom`
- Positions content at the bottom of the frame
- Can be combined with other options

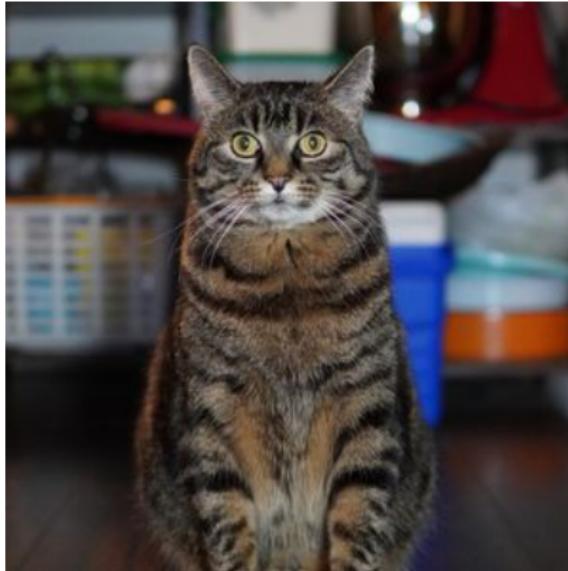
# FRAMEI: COMBINED OPTIONS

- This demonstrates combining multiple options
- Using [fs=small,sep=S,align=top]
- Small font size for more content
- Compact spacing (S)
- Top-aligned positioning
- Perfect for comprehensive slides



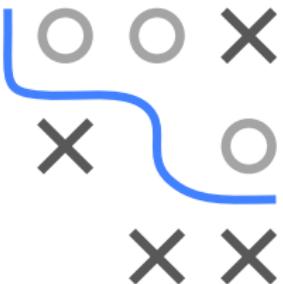
# INTEGRATION WITH FRAMEI ENVIRONMENT

- The framei environment creates an itemize list
- We can nest splitV inside list items
- Using splitVTT within a list item:



▶ Click for source

- Back to the main list
- With proper spacing between elements



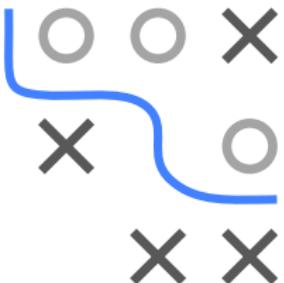
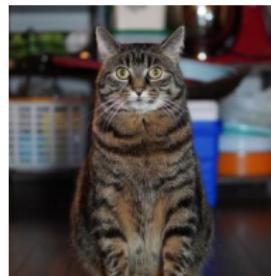
# FRAMEI WITH NESTED ITEMIZE AND FONT SIZE CONTROL

- Item A
  - itemize inherits footnotesize
- Global item
  - itemizeM overrides with tiny fontsize
- Another global item
  - itemizeM inherits footnotesize
- Item B

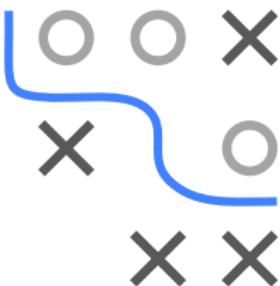


# SPLITV INSIDE REGULAR ITEMIZE

- First regular bullet point
  - Some more text in an item
  - This appears beside the image
  - No need for a nested itemize environment
- 
- Assume data is separable
  - We can continue the list afterward
  - Another example with top alignment:
- 
- Top-aligned content
  - In a normal itemize environment
  - Final regular bullet point



# FRAMEV LIKE FRAME BUT WITH FONT SIZE PARAMETER



This entire frame uses the small font size.

- this regular itemize list uses the small font size
- you can nest things in here as needed
- this itemizeM list overrides the font size
- if you use itemizeM etc. they just inherit the font size as well

# FRAMEV WITH NESTED REGULAR ITEMIZES



- Item A
  - Subitem 1
- Item B

# FRAMEV WITH TOP ALIGNMENT

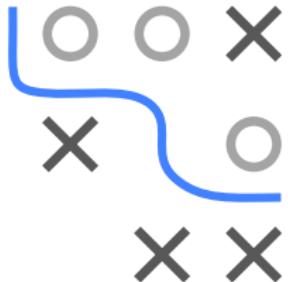
This framev uses [align=top] to position content at the top of the frame.

Unlike framei, framev does not automatically wrap content in an itemize environment.

- You can add itemize lists manually
- Top alignment applies to the entire frame content
- Combine with font size: [fs=small, align=top]



# FRAMEV WITH BOTTOM ALIGNMENT



Content positioned at the bottom using [align=bottom].  
This is useful for emphasis or visual variety in presentations.

# FRAMEV: SMALL FONT + TOP ALIGNMENT

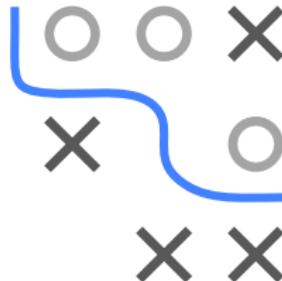
This demonstrates combining font size and alignment options:

[fs=small, align=top]

- Small font size makes room for more content
- Top alignment keeps content consistently positioned
- framev provides flexibility without automatic itemize wrapper

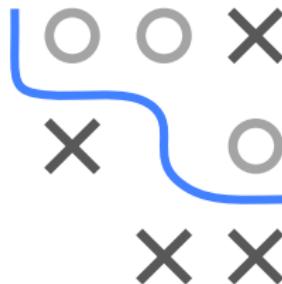
You can also use shorthand: [align=t] instead of [align=top]

Similarly: align=c for center, align=b for bottom



# VERBATIM CONTENT

- Verbatim does not work with framei or framev
- But you can mix itemizeL etc and verbatim
- The important thing is to use "normal" frame environment with  
fragile,c



```
#include<stdio.h>

int main() {
    printf("Hello World\n");
    return 0;
}
```