- iOS Design
  - KEY TAKEAWAYS
    - use two column design for tablet in portrait
    - list view is the iOS standard approach for browsing data
      - using slider in “Card” view is an anti-pattern
    - Using pure functions operate at the machine level (e.g., `GetField ( sort selection )`
      - fastest way possible to get data
  - List views
    - three fields per row in portrait
    - show more fields in landscape (six per row) because of available width
      - can potentially use “Hide if” and anchoring to put fields on same layout (don’t need to switch layouts)
    - Sorting Methods
      - Best Practice
        - One interaction point (“Sort” button)
          - use dropdown list
        - One sub summary part
        - One script
  - Form view
    - two column design (example was in portrait)
    - top area - what do you want to stand out?
      - e.g., photo, first & last name, key fields, etc.
    - combine buttons together (one control area)
      - also at top of screen
    - below the header, put the fields into column1  or column2
    - use dropdown & popup lists
    - alternate input methods
      - use keyboard type (e.g., for numbers, email addresses, etc.)
      - siri dictation
    - use APIs for things like address (e.g., Google Maps API)
      - examples are in starter solutions
    - Approaches for large layouts
      - use popovers (e.g., Gradual Disclosure)
        - popovers to show fields
        - popovers to show other buttons
      - build layouts with READ-ONLY fields by default, because they’re more compact
        - for each data group, can have a popover that shows editing fields (which take more real estate)
      - Things that are popovers on the iPad should become layouts in the iPhone
  - Portals
    - don’t work well with iOS on scrolling layouts
      - particular problem: nested scroll view
        - e.g., scrolling down a layout when suddenly encounter a portal; the portal starts scrolling, not the layout
      - instead, use List View
        - go to a list view based on the related table, rather than a portal showing the related records
      - Example: Invoices, Event Management, and Projects starter solutions
    - can be used in a Master-Detail or similar approach, where the portal is the primary object on the screen
      - most likely an iPad in layout view, not iPhone or iPad portrait
      - examples in iPhone
        - Yelp and LinkedIn for iPhone
  - Searching
    - ...?
  - Navigation & commands
    - put actions on the right
  - First design of the iPad, then for the iPhone
    - basically get the iPhone view for free
    - build/prioritize for portrait
      - if you design for landscape, you won’t get iPhone portrait for free
  - Prioritize your fields — don’t try to show everything
    - split it into multiple layouts
  - Q & A
    - What are the pros & cons of using merge fields?
      - If you use related fields, merge fields can have issues with long names
    - Base table scrolling (e.g., going from invoice to invoice with swipe)
      - can use Get Gesture
      - can use buttons
        - go back to list view, and trigger a script to go to the next related record
    - Getting fullscreen views
      - gestures in FMG 14
        - three fingers up hides the chrome
        - three fingers down shows the chrome
    - What is the advantage of merge fields over regular fields?
      - Text reflows better — fields always take up a certain amount of space, even if they’re empty. Merge fields will auto-collapse if data is not present.
      - Merge fields are more compact.