* Flows
  + Not the same as workflows
  + The most powerful process automation tool that we have
  + Visual coding
    - Drag and drop
  + Still has limits but can be useful
  + Not as powerful as actual coding in Apex
  + Salesforce is looking to make a certification just for flow
  + Types – more than below
    - Screen flows
      * Essentially a setup wizard
      * Guides users through creating a contact or account
      * Can be added to a page
      * Can also be attached to a button
      * Can be embedded on an experience cloud site
      * Can embed pretty much anywhere
    - Auto-launch flow
      * Launched by something else
      * Invoked by another process
      * Can call a flow from a flow
        + Not a trigger, is specifically called by another process
    - Platform event trigger
      * Triggered by a platform event
      * Use case: read a message that came in and process it
    - Scheduled trigger flows
      * Get data from anywhere
      * Perform at a specific time at a specific interval
    - Record triggered flows
      * Information comes from a change in a trigger, much like Apex triggers
  + Create using flow builder
    - Need specific permissions to be able to create and manage a flow
  + Flows are versioned
    - Need to activate and deactivate and can return previous versions
  + To build
    - Go to flows in setup
    - New flow
    - Enter the type
    - All flows start with a “start”
  + Elements
    - Things your flows can do
    - Interaction
      * How it interacts with a user
    - Logic
      * Control statements
      * Data assignment
      * Data manipulation
    - Data
      * DML statements
      * Can also retrieve records
  + Always test in debug before activating
    - Be sure to save before you debug
  + To deploy
    - Save
    - Activate
    - To embed in a webpage
      * Go to lightning app builder
      * Edit the page you want
      * Scroll through custom components until you find flow
      * Drag and drop
      * Follow the UI
  + Flow versions
    - When you want to edit a flow, you have to create a new version
  + Autorun flows run in the background automatically
    - Can help in dynamic approval routing
    - Can run before or after a save
    - Help update a lot of fields
    - Can run on delete
    - Always runs before delete
* Review
* Save order of execution
  + If a tool was working, then stopped, your save order of execution likely got messed up
  + Series of events that occur during a save
    - Specifically new and updates
  + Order
    - Loads the original records from the database
    - Perform system validation if the save is initiated from UL
    - Execute *before* record-triggered flows then execute *before* apex triggers
    - Run most system validation steps again and custom validations
    - Execute duplicate rules
    - Save
    - Execute *After* Apex triggers
    - Assignment rules
    - Execute workflow rule, if the workflow updates a field
      * Execute b*efore update* Apex trigger
      * Execute most system validation but no custom validation
      * Save to database
      * Execute *after update* Apex triggers
      * Executing workflow can cause Apex triggers to fire, which can cause recursion which we do not want
      * This loop is called a spur
    - Execute process
      * Execute *before update* Apex triggers
      * Execute most system validation but no custom validation
      * Save to database
      * Execute *after* update Apex triggers
      * Execute workflow rules
      * Execute processes again if recursion is set in process builder
    - Execute after-save record-triggered flow
      * Execute *before update* Apex triggers
      * Execute most system validation but no custom validation
      * Save to database
      * Execute *after* update Apex triggers
      * Execute workflow rules
    - Perform calculation for roll-up summary field or cross-object workflow on ancestors, working from parent upward
    - Execute criteria-based evaluation
    - Commit to database
    - Execute post-commit logic
    - [If an update occurs on parent record in a roll-up summary, another whole loop is triggered]
  + All of this is part of the same transaction
    - This is why it is important to pay attention to what you are doing and the number of records you are modifying
  + Any fail will make all changes roll-back to last commit
  + Transaction
    - An interaction with the database
  + One person can use a lot of resources
    - That is why we have governor limits
  + Use declarative tools first, since they are later in the save order of execution and are supported for longer