* QC at 9:30am on Tuesday
  + Will do some review at the end of the day
  + Will also review on Tuesday
  + Will be over last 2 weeks
* Little bit of review
  + Transient is reset every time the page is refreshed
* Visualforce continued
  + Remember to keep view state small
  + Parameters
    - Passed through URLs in visualforce
    - <apex:param>
      * Has to be a direct child of <apex:actionFunction>, <apex:actionSupport>, <apex:commandLink>, <apex:outputText>, <apex:interview>
      * Allows to pass a parameter to its parent
  + Wrapper classes
    - Special type of apex class that is used to collect values in a single object
    - Similar to making a custom data type
    - Gives additional data to an object that you don’t want to be an actual field on the object
  + Wizards
    - Sets of pages that guide users through a process
    - Old version of screen flows
    - Use the same controller
    - Have command buttons that will move to between pages
  + Testing visualforce pages
    - Two ways
      * Preview the page
        + Also called viewing markup
        + Should view your visualforce page in several different browsers to make sure it looks how you want it to across all browsers
        + Want to ensure full functionality regardless of any parameters
      * Writing code
        + Writing test classes
        + Write tests using best practices for controllers and controller extensions
        + Do not need to write code for standard controllers
        + To test the page, you should set the page reference to the page you want to test
        + PageReference

pageRef = Page.nameOfVFPage

test.setCurrentPage(pageRef)

* + How to include in the UI
    - Navigate to the tabs in setup and add a visualforce tab
    - Attach the page to the tab
    - You can embed in an existing page via lightning app builder
      * You will need to enable use in the lightning environment on the visualforce page setting in setup
  + Dynamic visualforce
    - Allows for selective rendering of components
      * Also called dynamic rendering
    - <apex:dynamicComponent>
      * Has a required attribute componentValue
    - You pass the expression syntax formatted name of a custom controller method that will return a component
    - {!reference[expression]}
  + Custom labels
    - Not confined to just visualforce
    - These labels translate a base message into the language set in the user settings
    - You need to enable this in translation language settings
* SOSL
  + Indexing
    - The global search can’t search every field on every object
    - It can only search indexed fields that have allowed search
    - Reduces the computational expenses of searching
    - Searchable objects
      * To make an object searchable, you need to allow search in the object creation
    - We don’t have an allow index field checkbox
      * Fields are indexed when they are marked as an external ID or unique field
    - Most standard fields are auto indexed
    - Salesforce can add an index to a non-indexable field but you need to contact them
  + We are not querying the database, we are searching the database
    - This is why indexing is important
  + Salesforce object search language
  + Performs text queries against a subset of the search index
  + Returns a List<List<SObject>>
  + Example:
    - 
  + Clauses
    - FIND
      * First required one
      * Followed by query string in quotes
      * Case insensitive
      * This is what we want to look for
    - IN
      * Optional
      * Can take the values of ALL FIELDS, EMAIL FIELDS, NAME FIELDS, NAME FIELDS, or PHONE FIELDS
      * This is where we want to look for the query string
      * All fields does not include all fields
        + It will only actually search the text, email, phone or name fields that match our query
    - RETURNING
      * Last required clause
      * After it, we provide a list of all the objects and their fields that we want returned from our search
      * This does not determine what is searched through, only what is returned
      * You can return non-indexed fields with your objects
      * The list of objects being returned will be a comma separated list
      * After each object name, there is a ()
        + Within those you specify the fields you want to return and some other things
    - RETURNING (nesting)
      * Inside the () of the object you want to return, you can futher filter what you want returned
      * WHERE
      * ORDER BY
      * OFFSET
      * Functionally identical so SOQL
      * FORMAT()
        + Can wrap around a date, number, time or currency field to have the data displayed in the users locale
      * WITH
        + Data category

Supported by apex

Allows us to filter salesforce knowledge articles by their category

* + - * + DivisionFilter

Supported by apex

Used with divisions

* + - * + Network

Allows us to find or filter results from a specific experience cloud site (used to be community)

* + - * + Spell\_correction

Spell checks your search string

This can be used with apex

* + - * + Snippet

Can be used with apex

Allows you to see the surrounding context for your search string

Returns both the context and the match

Returns html and mark tags

* + - * + Highlight

Not apex supported

Must be executed with something like the rest API

Use marktags around the match of the search strings

Only works on certain standard objects

* + - * + Metadata

Not apex supported

Allows us to retrieve information about the fields that we return themselves

* + - * + PricebookID

Not supported by apex

Filtering products based on a price book id

More to come next week

* + - * + LIMIT

Is interesting in SOSL

You can use it in 2 different places

Inside the ()

Limiting the number of records returned of that specific object

Outside the ()

Limiting the total number of records returned

* + Wildcards
    - \*
      * Can include in your text string searches
      * Matches 0 or more characters
    - ?
      * Exactly 1
    - OR/AND
      * Allows for several strings to be searched at once
      * ‘Hello or world’
  + When to use what
    - SOSL is computationally expensive
    - More often than not, you will use SOQL
    - You should use SOSL if you don’t know what object you are looking for
* Reports and dashboards
  + Two types
    - Standard
      * Come out of the box
    - Custom
      * Only show up on custom objects if we check them
  + Adding a field will make it show up on that report
  + You can have new fields added to custom report types
  + Basically a report is a fancy spreadsheet
  + Custom report types allow you to create your own framework for a report
    - Can select up to 4 related objects
    - Use to open up across wider variety of relationships
  + Reports are stored in folders
  + We have different report formats
    - Tabular
      * Simplest way to look at data
    - Summary
      * Allow you to group by rows, view subtotals and make charts
    - Matrix
      * Allow you to group by rows and columns
    - Joined