* Jest
  + JavaScript test runner
  + Javascript library for creating, running and structuring tests
  + Intended to support behavior driven development
  + Ships as an NPM (no package manager) package
  + You can install it on any JS project
  + One of the most popular test runners
  + What should be tested?
    - Essential business logic
  + With jest you can write unit tests and end-to-end tests
    - Unit testing is on an indivisible unit of code being tested in isolation
    - E-2-E testing (end to end) is where you simulate a user experience of a feature with everything integrated together
  + Typical test flow
    - Import the function to test
    - Give an input to the function
    - Define what to expect as the output
    - Check if the function produces the expected output
  + How to set up jest
    - Have Node.Js and npm installed
    - Salesforce CI command sfdx lightning:lwc:test:setup
      * Run the command in the top-level directory of each salesforce dx project
    - To create a test for a component: sfdx force:lightning:lwc:test:create -f force-app/main/default/lwc/helloLEC/helloLWC.js
    - To run the test, do npm run test:unit
* Debugging with LWC
  + The lightning component framework executes in one of two modes
    - Production mode
      * The default mode of the framework
      * It is optimized for performance
      * Framework code is optimized and minified to reduce the size of JS code
      * The JS code served to the browser is obfuscated
    - Debug mode
      * When this is enabled, the JS code isn’t minified and is easier to read and debug
      * It also adds more detailed output for some warnings and erros
  + DevTools
    - Chrome has devtools for debugging JS
    - Load the simplest page possible that can run your lightning component
* Open source LWC
  + LWC is open source
  + It allows developers to build enterprise-ready web components on any platform
  + The open source dev site is lwc.dev
  + Lightning Out can be used to interact with salesforce data
    - The difference between LWC on the platform vs open source LWC is that LWC is a managed version of the open source LWC
  + When working off the salesforce platform, you can download LWC, configure it your way, deploy your application on any hosting environment and choose when to upgrade, give you a lot of control over your functionality
  + LWC OSS and LWC on the platform have different release schedules
  + The open source LWC engine is identical to the LWC engine of the platform
    - The difference is the way the compiler is configured
  + Usage of experimental LWC APIs on the salesforce platform is restricted
  + When pushing an LWC module to the salesforce platform, the following lenting rules are applied
    - All rules from @salesforce/eslint-config-lwc/base are enforced
    - Dynamic importing is prohibited
    - Access to aura from a LWC module is prohibited
  + Components run with @lwc/synthetic-shadow enabled
    - Mode that allows support for IE11 which doesn’t implement the shadow DOM
    - Link and xlink:href attributes on the SVG <use> element are sanitized by locker to prevent potential malicious script injection
    - LWC modules are evaluated onside Lightning Locker
      * On top of restrictions applied to standard web platform APIs