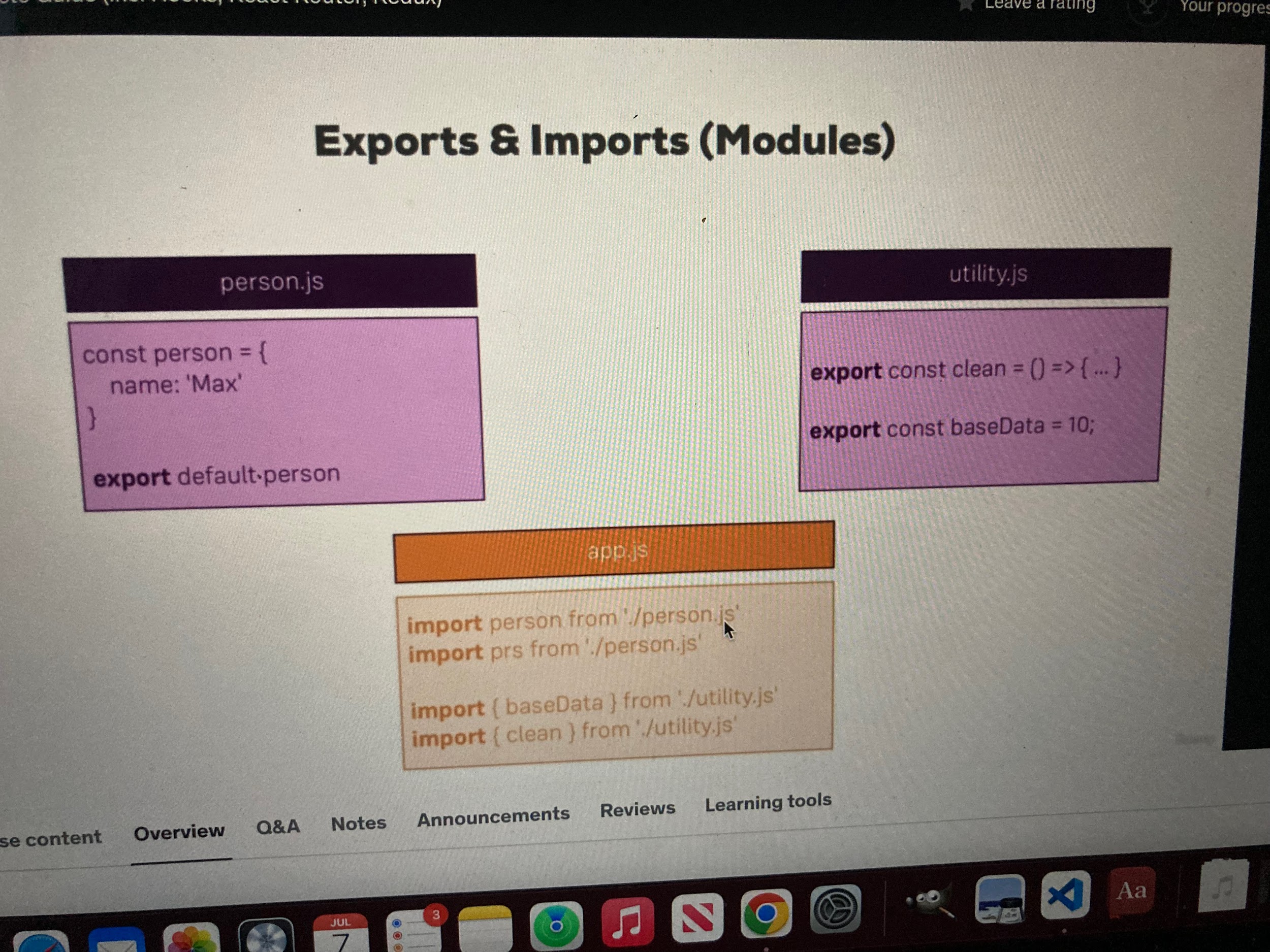
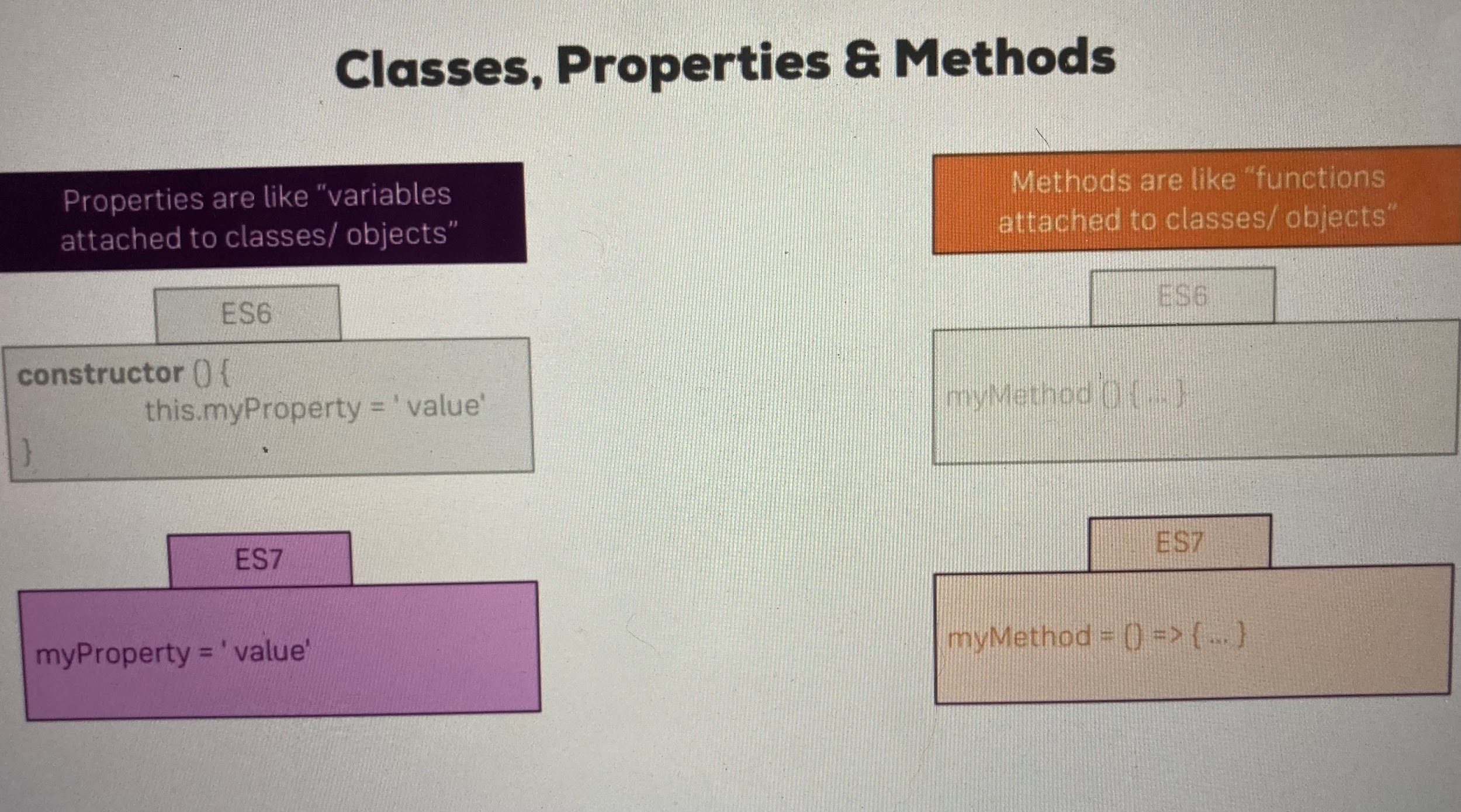
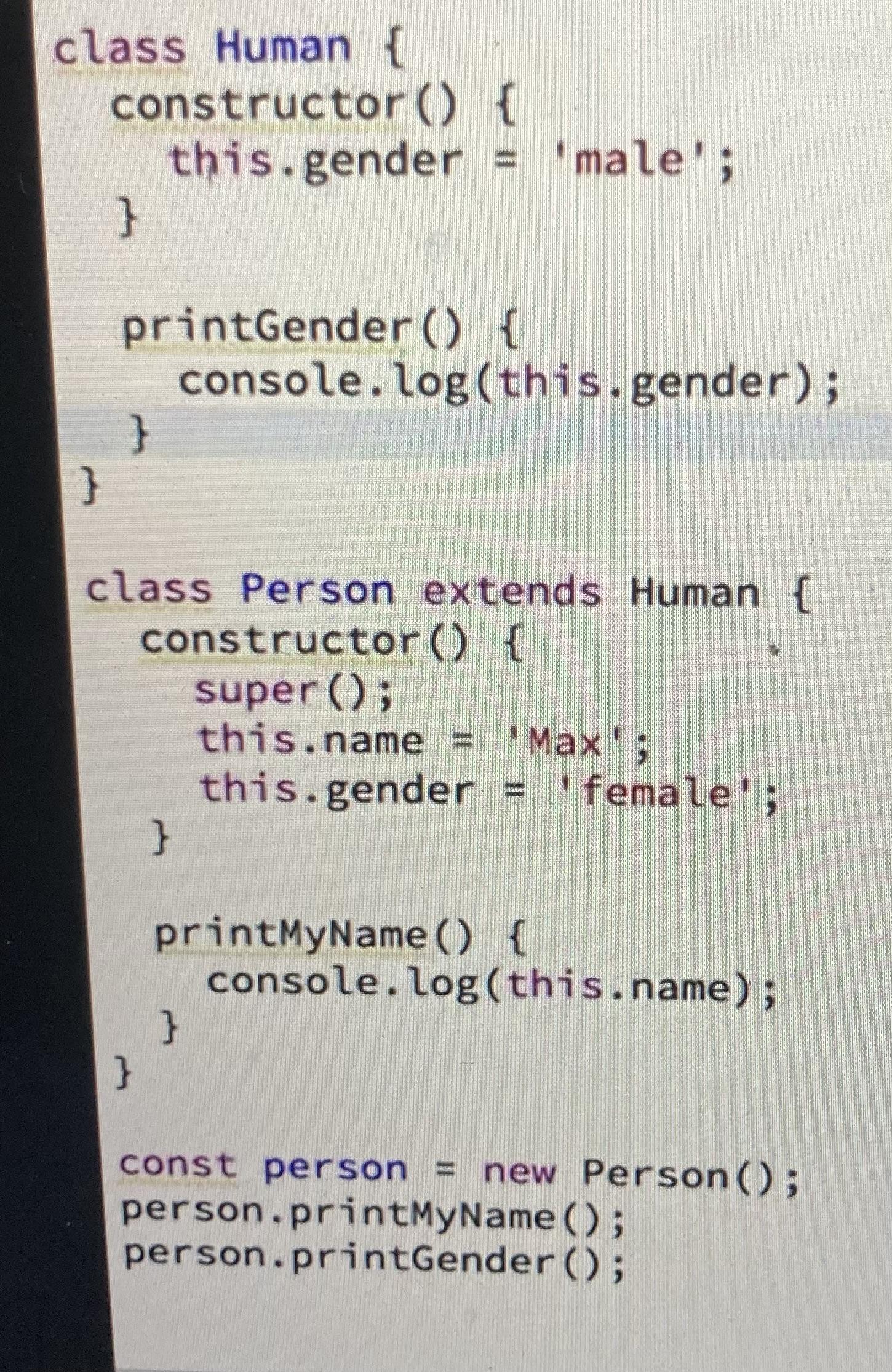
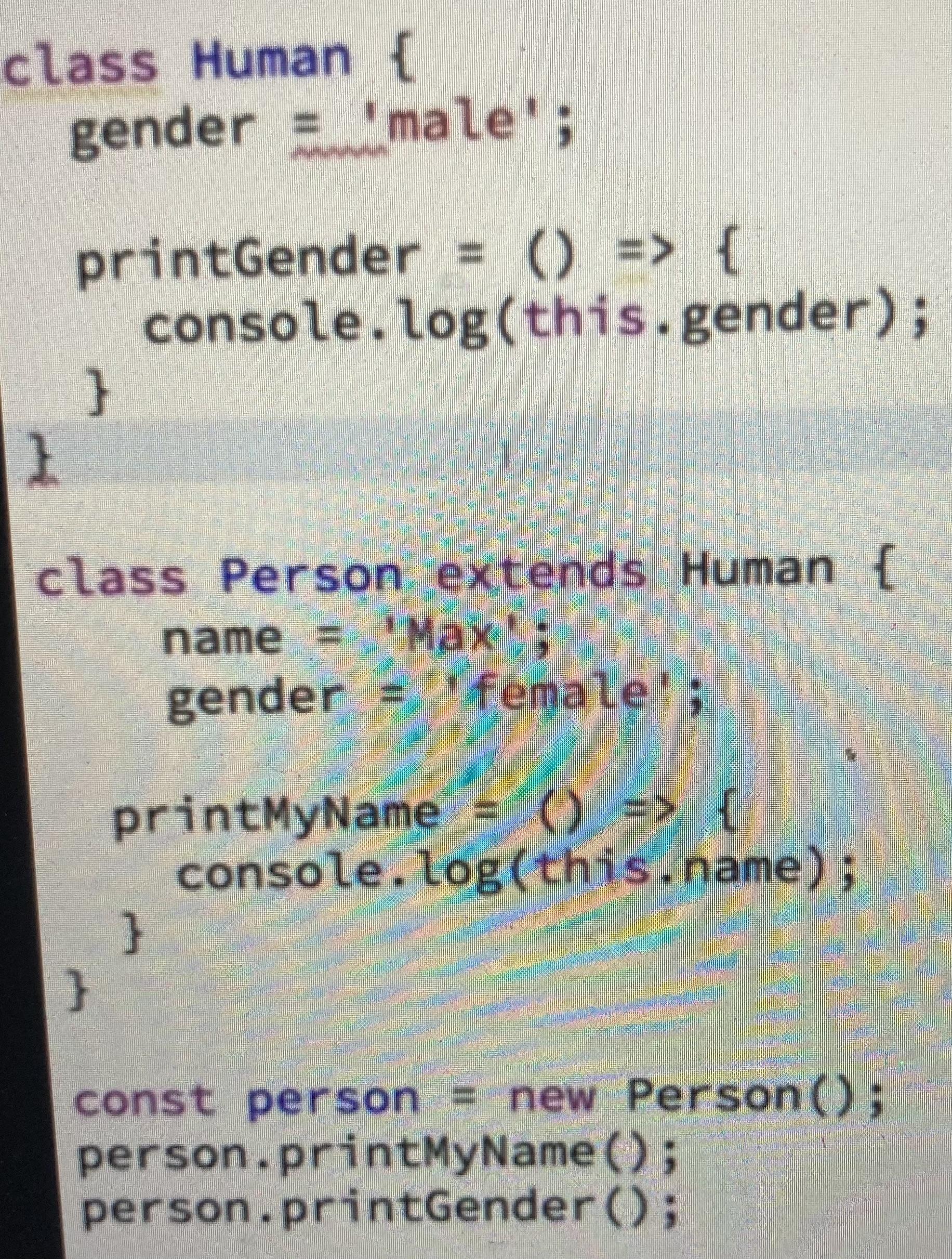
* What is react?
  + JS client side library for building user interfaces
  + Ex. netflix classic ex. Of react interface
    - Mobile apps/desktop apps are very ‘reactive’ you don’t have to wait for new pages etc..
  + Websites used to be clunky as you only had the ‘request’ ‘response’ cycle where you were sent a new html page by the server with every click. Javascript allows you to break that cycle and tweak and request new info without loading a new page from the server
  + . Runs logic in the browser,,, able to manipulate in the DOM… to manipulate what the user sees without loading a new page or visible request to the server
  + React is a library that is a building block to help with making complex, modern reactive user interfaces for the web
  + **Why is react necessary?**
    - Javascript is an imperative approach– you have to describe every action after every action to ensure it works properly… can reach its limits.. Also a lot of simple repetitive tasks like creating elements for every part of the html doc etc..
    - React all about splitting application into small building blocks/components where they all have clear task to make it manageable. Does the work of combining your code to bring things together
      * React library defines elements by default
    - Declarative component based approach
  + **Build SPA(single page applications)**
    - With react we often build single page applications
    - Sometimes you might add react widgets to html pages etc..
      * Server only sends one html page then react takes over and controls the UI
    - Ex. netflix… if you go from main page to liked list… it appears like you went to a new page, but there was never a server request for it.. You just used JS thru react to manipulate what is seen.
      * Smooth UI/experience
    - Other JS libraries,
      * Certain features not included in React and you may need to install third party libraries
        + Lean and focused component based UI library… some features like routing added via community packages
        + 3rd party packages often necessary to add when using react
        + Main focus of react is the components and how u use them… so certain things are added via 3rd party
    - Angular is a react alternative.. JS library
      * Has more features, also very popular.. Embraces typescript…
      * Requires few 3rd party addons as it is larger and has more features built in, but can be excessive for smaller projects.
    - Vue.JS
      * Another alternative.. Kindof mixture between angular and react. More features than react but not as jam packed as angular
* What to expect (general overview)
  + 3 main sections
    - 1. basics/foundations of react
      * Core features that you have to know for everything you do
      * Components/ using/building/ building UIs
      * Work with events/data(clicks etc.), props and state
      * Styling apps/components
      * React hooks
    - 2. More advanced
      * Building for production
      * Popular concepts that are commonly needed
      * Side effects “refs” and more react hooks
      * React’s Context API & redux(one of the most important 3rd party libraries for react)
      * User input… forms… requests, store/fetch data in database, build “custom hooks”
      * Routing, deployment, give the illusion of multiple pages, deploy app, NextJS(another framework for building on react, makes building even easier)
    - 3. Summaries and refreshers
      * Javascript refresher
      * React JS summary at end of course
* 2 different ways to do course
  + 1st. Standard
    - Go thru course step by step
* Javascript review
  + import/export
    - Modular code
    - Within javascript files we can import from other files
    - Export default-variable



* If you use default– that just means that if you import something from that file it will be
* the default export
* Properties vs methods
  + properties are like variables attached to classes/objects
    - Can define in constructor or use easier syntax by defining with =
  + methods are like functions attached to classes/objects
    - Method syntax nowadays uses arrow functions so this can be used
* 
* Old way
* 
* New way (next gen js)
  + This syntax makes it unnecessary to add constructor or this as it’s already understood by default.. Also super isn’t necessary anymore
* 
* Primitive vs reference types
  + Numbers strings booleans are primitive types..
  + If you copy the info of a variable to another variable it creates a copy of the original but the original is untouched so they are completely separate despite having same info
  + Objects and arrays are reference types
    - If you copy info from one array to another it is referencing the original
    - If you create an object then another object that references that and change info in the first object.. It alos updates the second object