- StyleResolver has a gigantic switch (StyleResolver.cpp)
    - Before going into the switch, look ar a map in DeprecaredStyleBuilder,
    - If the property is not in the switch, check SVGStyleResolver
    - This means style resolving is in 3 places!
- We also have the CSS Parser, which is another gigantic switch for all properties
- Computed style is a third gigantic switch with all properties
- Want to at least combine classes for resolving the style so that at least that part is in one place instead of 3 different classes
- The property resolving function is not used for resolving a shorthand
- The question is: is there a way to combine the switches?
    - no argument that having 3 ways to resolve styles is a bad thing
    - should we go back to a single giant switch, or to something else?
        - old giant switch also came with a bunch of macros
            - unsure if that's good or bad
        - Use template functions in switch to make the code easier to read
    - moving everything into one file/giant switch doesn't solve the problem
    - is it possible to make modules for certain features?
        - modularization can have performance problems
        - maybe there is a way to just modularize logically?
    - suggestion to index into an array based on CSS property index, array contains pointers to objects or functions
        - concerns on this being slower than a switch
    - tabled question about complexity addition to switch solution by allowing dynamically enabling/disabling CSS properties
    - suggestion to modularize by splitting into multiple header files
        - have a large switch in the cpp that calls inline functions from header files
        - one header file per feature? or per property?
        - already done for shapes
        - This was a bad experience at the beginning of the project with everything organized into files
            - any arbitrary grouping into features makes it harder to find the code
            - one advantage of a giant file is the same as an advantage of lots of small files
                - makes it easier to find things when you're not an expert with the layout
        - 400 CSS properties, if grouped by shorthand, would end up with ~300 header files
- question about what is the real problem?
    - answer: code is unreadable
- is it possible to generate the switch statement + supporting code?
- Right now, the code is a switch statement + a bunch of calls to setPropertyHandler that is pretty much a copy of the switch

- concerns about generation because it's another skill/programming language that's needed to work on/understand the code
- concerns about generation about accidentally editing generated code and getting confused why it didn't fix anything
- there might be a performance issue in addition to the code readability issue
- another question is about compiled code size, if this refactoring will make it better or not
- CSSPrimitiveValueMappings.h
    - all this code is here for modularity of the style system and the CSS DOM
    - this is a case where template has been used wisely/correctly
- one of the issues with StyleResolved is that it has some giant templates with insane numbers of arguments
- Move all code to StyleResolver into a giant switch, not using templates, since the templates are the hard things to read
    - but then maybe we'll have to use Macros instead of the templates
- can we move SVGCSSStyleSelector.cpp into the giant switch as well?
    - yes
    - this is old code that shows how StyleResolver used to be implemented
    - the HANDLE_INHERIT and HANDLE_INHERIT_AND_INITIAL macros are interesting
- Conclusion:
    - use a single switch statement, have the statement call functions, in a single cpp file
    - once we have it set up like that in a single cpp file, then we can optimize for more readability, performance, etc
    - make sure we *really* like the first function style/etc before writing a whole bunch of them
    - This switch statement and supporting code should be in a file of it's own, like "StyleBuilder"