|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | **Cues**  NoteGem Horizontal Line      what are three concepts in preprocessing to understand?      what does preprocessing do?    what is LESS?    how do you declare a variable in LESS?        what is a mixin?  why are they named that?        what's the rule for nesting and why?          what operators can you use in LESS?          what is escaping?                how do you comment in LESS?                what is scope?        how do you import in LESS?          what goes in a function?                          how do you loop in LESS?                  what are some pre-defined functions you can use in LESS?        what is a parametric mixin? | **Notes**  NoteGem Horizontal Line      Three concepts in preprocessing to understand     * **less** = language or syntax used to write code * **node** = compiler that converts language * **normal css** = output from compiler     preprocessing extends css's ability by adding abstractions and making it easier to use    **LESS** = language extension for CSS    variables start with @ (@variable-name: #002244;)     * .class {color: @variable-name;}     **mixins** derive name from ability to mix classes together     * example: .class{color: red; font-size: 14px;} * .example-mixin {.class();}     **nesting** = avoid going over four levels deep     * can nest media queries inside selectors * increases code readability * \*\*\*INSERT PICTURE FROM ONENOTE CURRICULUM * \*\*\*INSERT DIAGRAM FROM ONENOTE CURRICULUM     can use basic math operators or any number, color, or variable  operators convert units before adding     * leftmost stated unit type     **escaping** = use any arbitrary string as property or variable value     * ~"\_\_\_\_" or ~'\_\_\_\_\_'     namespaces/accessors:     * #bundle{.button{…}.tab{…}} = #header a {#bundle.button(); OR #bundle>.button}     /\* comment on less \*/    **maps**     * #colors(){primary: blue;secondary: green;} = .button{color: #colors[primary];}     **scope** = variables and mixins first looked for locally     * else inherited from parent     **import** = @import "library"; (library.less)     * @import "typo.css";     **functions**     * if returns one of two values - depending on condition * if(condition, value if true, value if false)     can store condition in variable     * @bg-light: boolean(condition); * div{if(@bg-light,black,white)};     escape ('string to escape') = applies URL-encoding to string or e('\_\_\_\_\_') escapes string without quotes    **loops**     * .recursion() when (@ctns {.recursion()} * this repeats until condition is met     extract returns value specified from list    each = bind evaluation of ruleset to each list member     * each(list, rule);     **others**: ceil() floor() percentage() round() sqrt() abs() min() max() pow()    **boolean** = isnumber isstring iscolor iskeyword isurl ispixel isem ispercentage isunit isruleset    **parametric mixins**     * .custom-layout(@justify, @align){display:flex; justify-content: @justify;align-items: @align;}     use: .custom-layout(flex-end, center);    most functions return a value | |
| |  | | --- | | **Summary** | |