1. Less是预处理器，是线下编译的，最终参照的还是css
2. Less编译成css是koala
3. Less中的注释
   1. 以//开头的注释，不会编译到css中
   2. /\*\*/的注释会被编译到css中
4. Less中的变量
   1. 属性值作为变量 @color : xxxx
      1. Background : @color;
   2. 属性名和选择器作为变量 @selector : xxx
      1. @{selector} {xxx:xxx}
5. Less中的延迟加载
   1. 等定义变量的位置全部加载完后才去利用变量
6. Less中的嵌套规则
   1. 普通嵌套 .wrap{.inner{}}
   2. &规则·
      1. &表示此时所在嵌套中的所有选择器
7. Less中的混合
   1. 普通混合，相当于嵌套
   2. 不带参数混合，与普通混合差不多
   3. 带参数混合 .hunhe(@a,@b){ color : @a ; background : @b}
      1. 有点像函数，可以给选择器的属性做成参数
   4. 带参数且带默认值的混合 .hunhe(@a：white ,@b : black){ color : @a ; background : @b}
   5. 命名参数 .hunhe(@a : black) （被调用时）
      1. 当形参与实参不匹配的时候，在调用时可以直接声明给哪个形参值
   6. 匹配模式
      1. .triangle(R，@a,@b) .triangle(L,@a,@b) 此时R,L不为参数，只是用来标识，调用处也需要写R or L
      2. .triangle(\_@,@a,@b) 当每次调用R or L 的时候，都会调用此混合,此混合的a,b形参名字可以不为a,b
   7. @Arguments
      1. Duogecanshu(@a,@b,@c){ background : @arguments}
      2. 调用时，可以直接duogecanshu(1px, solid,black) ，@arguments会自动匹配上
8. Less的运算
   1. 可以直接在属性的值进行运算
      1. 如： width : (100 + 100px)
9. Less的继承 :extend(.hunhe)
   1. 无法设置给选择器设置参数
   2. 性能比混合好，但是柔韧性比较低
10. Less的避免编译语法
    1. ~” ” 内的东西不会进行编译
    2. 如：~”calc(100+100px)”的时候，100+100px不会被less的运算法则进行编译。