# 常量定义

使用**@**标识常量名，后跟“:常量值”。常量可被用于选择器，公式，或其他常量中

@nice-blue:#5B83AD;//常量定义

@light-blue:@nice-blue + #111;//常量计算

#header{

    color: @light-blue;//常量使用

}

===输出CSS如下===

#header {

  color: #6c94be;

}

# 混合（Mixins）

## 基本使用

将已经定义好的A选择器写入B选择器中，B中就拥有了A的样式。

@nice-blue:#5B83AD;

#A{color: @nice-blue;}

#B{

#A() //mixin混入，这个()可写可不写

}

===输出CSS如下===

#A {color: #5B83AD;}

#B {

  color: #5B83AD;

}

## 选择性输出

括号()是可选的，如果希望less中定义的变量不输出到CSS中就必须加()

.button-style(){

    font-size: 11px;

    &:hover{

        border: 2px solid black;

    }

  }

button{

    .button-style();

}

.a{

    .button-style;

}

====输出CSS如下======

button {

  font-size: 11px;

}

button:hover {

  border: 2px solid black;

}

.a {

  font-size: 11px;

}

.a:hover {

  border: 2px solid black;

}

## 命名空间

需在更复杂的选择器中混合属性，则可以堆叠多个ID或类

.button-style(){

    .inner{

        background-color: red;

        font-size: 100px;

    };

    .outer{

        background-color:blue;

        font-size: 200px;

    };

    .hider{

        font-weight: 200;

    }

};

.f{

    .button-style > .inner();

};

.k{

    .button-style > .outer;//“>”前不允许加括号

};

.g{//混合button-style中多个属性，生成多个结果

    .button-style()

};

====输出CSS如下=====

.f {

  background-color: red;

  font-size: 100px;

}

.k {

  background-color: blue;

  font-size: 200px;

}

.g .inner {

  background-color: red;

  font-size: 100px;

}

.g .outer {

  background-color: blue;

  font-size: 200px;

}

.g .hider {

  font-weight: 200;

}

## 受保护的命名空间

@color:blue;

@fontstyle:100px;

//受保护的命名空间，条件成立就输出，条件不成立编译器报错

.abc when (@color = blue) and (@fontstyle = 100px) {

    .mixin() {

        color: @color;

        font-size: @fontstyle;

    }

}

p{

    .abc .mixin();

}

==输出CSS如下=======

p {

  color: blue;

  font-size: 100px;

}

## !important

被!important标记后，编译成CSS后也带有!important。

## 混入函数

//可为参数指定默认值

.mixin(@value1:blue,@font-size:200px){

    color: @value1;

    font-size: @font-size

}

.fon{//传入的参数可以写成key-value形式，也可以直接按顺序写，但不能跳着传参

    .mixin(@value1:red,100px);

};

.fon1{

    .mixin(green);

};

.fon2{

    .mixin();

};

==输出CSS如下====

.fon {

  color: red;

  font-size: 100px;

}

.fon1 {

  color: green;

  font-size: 200px;

}

.fon2 {

  color: blue;

  font-size: 200px;

}