

# Exercícios

1. [Introdução] Teste a configuração do seu ambiente:
  - a. Transforme o documento `teste.css` em um documento `.less`, defina uma variável `@cor` para as cores que se repetem e aninhe os três blocos. Utilize um compilador Less para gerar um documento CSS e teste-o para verificar se ainda funciona.
  - b. Altere o arquivo `teste.html` de forma que ele carregue o documento Less (em vez do CSS) para processamento durante a carga. Altere o documento Less (mude o valor da variável `@cor`) e veja se o HTML é alterado.
2. [Aninhamento] O arquivo `basics.css` contém diversos blocos de seletores contextuais e pseudo-classes. Utilize-o como base para criar um documento `.less` que reorganize os 17 blocos em um único bloco partindo de `body`, utilizando o aninhamento ao máximo (para contexto e pseudo-elementos).
3. [Variáveis] Remova a duplicação de dados da folha de estilos `variables.css`. Crie uma folha de estilos Less equivalente que utilize variáveis para:
  - a. todas as cores
  - b. famílias de fontes
  - c. partes de nomes de propriedades que se repetem (`border`, `color`);
  - d. parte da URL que se repete nas imagens
  - e. nome da propriedade `.sec1` (inclua `sec1` em uma propriedade)
4. [Extend] Utilize o documento `extend.less` que importa `basics.less` e usando apenas pseudo-elementos `:extend` (sem declarar novas propriedades), faça com que a página `extend.html`, que usa `extend.css` "herde" alguns estilos de `basics.less`. Estenda:
  - a. `.article` com `article`
  - b. `.article .contents` com `.contents`
  - c. `.article .contents .section` com `section`
  - d. `.article .contents .section h1` com `section .h1`
  - e. `.nav li` com `nav li`
  - f. `.nav li a:hover` com `nav li a:hover`.

Use aninhamento e faça adaptações se necessário. Verifique se a declaração usada no `extend` é uma correspondência *exata*.

5. [Extend] O documento `mixin-extend.less` utiliza um mixin `.gradient()` para construir um gradiente de cinco cores. O mixin está definido em `gradient-mixin.less` que é importado em `mixin-extend.less`. A página `mixin-extend.html` exibe um div branco, cujas dimensões e borda estão definidas em `mixin-extend.less`. Analise o código gerado no CSS e altere o documento `mixin-extend.less` de tal maneira que o gradiente apareça desenhado no div.
6. [Mixins] O documento `mixin-6.html` contém quatro tabelas idênticas que diferem apenas no id. O documento `mixin-6.less` aplica funções que variam a largura e o brilho de uma cor para cada tabela. Observe que há bastante duplicação de código. Crie um mixin para gerar o CSS para as tabelas que receba como argumentos pelo menos uma cor e uma largura. Chame o mixin para cada id e verifique se o CSS gerado é o mesmo.
7. [Mixins] Analise o documento `mixin-7.less` e identifique trechos duplicados ou trechos que possam ser embutidos em mixins. Crie os mixins indicados utilizando ou não os nomes sugeridos, e substitua os trechos duplicados ou longos por chamadas a esses mixins. Defina argumentos como indicados e use valores default se desejar.
8. [Operações e funções] Analise o documento `operations-8.less`. Veja o resultado em `operations-8.html`. Troque os valores fixos por valores obtidos a partir das duas variáveis declaradas no documento. Siga as instruções nos comentários para:
  - a. Determinar os valores de `@conversion` em duas células `td` através de conversão do valor de `@dimension` no seletor `#unit`.
  - b. Determinar os valores de `@result` em três células `td` através do uso de funções de arredondamento do valor `@number` no seletor `#round`.
  - c. Obter o valor de `@angulo` usando *apenas* o valor de `@number` (converta o número de radianos para graus); depois obter os valores `@result` através de funções trigonométricas sobre `@angulo` ou `@number`.
9. [Operações e funções] Abra o documento `operations-9.html`. Veja o HTML e o resultado no browser. O documento `operations-9.less` reposiciona os objetos usando valores fixos. Altere-o para que os valores

sejam calculados em função das três variáveis `@width` (largura de um quadrado menor), `@height` (altura) e `@gap` (margem). Use os comentários como guias.

10. *[Operações e funções]* O documento `operations-10.less` possui várias variáveis e valores fixos. Siga as instruções nos comentários para que os valores sejam derivados ou extraídos das variáveis e dados fornecidos:

- extraia a *menor* percentagem `@minp` da lista `@p`, a *maior* dimensão `maxh` da lista `@h` e a *última* cor `@lastc` da lista de cores `@c`.
- construa um string de query iniciado em `?`, separando pares `nome=valor` por `&` com caracteres especiais codificados no formato *url-encoding*. Use o exemplo fornecido como guia e quaisquer funções ou operadores do Less para obter o resultado. Depois que montar o *query-string*, inclua-o no final da `@url`.
- extraia a cor (formato `#rrggbb`) do *query-string* montado no exercício anterior. Será necessário converter o caractere `%23` em `#`, eliminar o início do string e converter o resultado em uma cor, que será usada em duas outras propriedades.

11. *[Loops e guards]* Utilize o documento `conditional.less` e escreva um mixin condicional para gerar colunas de acordo com a largura máxima da página e de cada coluna. *Multiplique* o número de colunas pela largura de cada coluna e some os espaços entre as colunas. Se este valor for *menor* que a largura total, gere as propriedades CSS para construir o número de colunas indicado. Se o valor for *maior*, calcule o maior número de colunas que pode ser inserido. Veja exemplos nos comentários.

12. *[Loops e guards]* Crie um mixin recursivo `.loop` para gerar o seguinte CSS quando chamado da forma `.loop(5)`:

```
.item-100 {
  top: 50px;
}
.item-200 {
  top: 100px;
}
.item-300 {
  top: 150px;
}
.item-400 {
  top: 200px;
}
.item-500 {
  top: 250px;
}
```

O mixin deve ser chamado com ou sem argumentos. Os nomes dos seletores e o valor de `top` devem ser calculados em cada repetição com base no valor do contador. Se o `loop` for chamado sem argumentos, ele

{less}

Error! No text of specified style in document.Error!

deve executar pelo menos uma vez.

13. [Loops e guards] O documento `loops-13.less` contém uma variável com uma lista de cores `@cores: red green blue yellow violet;`. Escreva um mixin recursivo que receba como argumento a quantidade de itens da variável `@cores`, e a cada iteração extraia a cor correspondente. Use `&:nth-of-type` para que o mixin possa ser chamado dentro de um seletor, da forma:

```
div {
  .loop3(length(@cores));
}
```

O resultado deve ser o CSS abaixo:

```
div:nth-of-type(1) {
  background-color: #ff0000;
}
div:nth-of-type(2) {
  background-color: #008000;
}
div:nth-of-type(3) {
  background-color: #0000ff;
}
...
```

Verifique o resultado no CSS e na página `loops-13.html`.

14. [Loops e guards] Abra o documento `loops-14-mixins.less`. Este documento possui:
- um mixin `.animate(@nome; duracao)` que gera propriedades de animação para vários browsers recebendo o nome do key-frame.
  - um mixin `.palette(@x; @y; @cor1; @cor3)` que gera cores para cada um dos divs da matriz de cores.
  - uma variável `@frames` contendo uma lista de percentagens (para animação).
  - uma variável `@turns` contendo uma lista de ângulos (para cores).

Este documento é importado pelo documento `loops-14.less` que define o tamanho de cada `div`, um bloco que chama os dois mixins em diferentes contextos e blocos de keyframes para Chrome e Firefox. Os keyframes chamam um mixin `.make-frames` que precisa ser implementado. O mixin `.make-frames` precisa extrair os dados de `@frames` e `@turns` para gerar a seguinte sequência:

```
0% {
  -webkit-transform: rotate(0deg);
}
25% {
```

{less}

Error! No text of specified style in document.Error!

```
-webkit-transform: rotate(90deg);
}
50% {
  -webkit-transform: rotate(180deg);
}
75% {
  -webkit-transform: rotate(270deg);
}
100% {
  -webkit-transform: rotate(360deg);
}
```

Implemente o mixin recursivo `.make-frames` dentro de `loops-14-mixins.less`. Quando terminar, descomente as linhas `//.make-frames`; em `loops-14.less` e veja o resultado no CSS e na página HTML.

15. [Cores] Adapte o loop do exercício 13 para que a cor gerada seja criada através da função `spin`. O mixin deve receber como argumentos opcionais: um número (`@contador`), uma cor inicial (`@cor`) e um incremento de ângulo (`@angulo`) que deve ter como valor default `360deg / @contador`. O loop deve a cada iteração gerar uma cor incrementando o ângulo. Por exemplo, se o valor máximo de `@contador` for 4, os ângulos gerados deverão ser `0deg`, `90deg`, `180deg`, `270deg`. Isto pode ser obtido somando  $360deg/4 = 90deg$  a cada repetição. As cores devem ser criadas com `spin(@cor, @angulo * (@contador - 1))`. Se o mixin for chamado da seguinte forma:

```
.div {
  .colorir(3);
}
```

(supondo uma `@cor` default de `#ff0000`), o resultado deve ser:

```
div:nth-of-type(1) {
  background: #ff0000;
}
div:nth-of-type(2) {
  background: #00ff00;
}
div:nth-of-type(3) {
  background: #0000ff;
}
```

Teste com diferentes valores.

16. [Cores] No documento `colors-16.less` existe um mixin chamado `.palette` que gera uma paleta de cores usando a função `difference`. Experimente este mixin com diferentes funções como `multiply`, `softlight`, `mix`, etc. e também com cores diferentes (mais ou menos saturadas) e veja o resultado na página `colors-16.html`.

{less}

**Error! No text of specified style in document.Error!**

17. *[+Less]* Abra o arquivo `more-17.html` no browser. O texto das cores nem sempre está legível pois a cor de fundo é clara demais. A cor está especificada no mixin em `more-17.less`. Altere a variável `@text-color` de tal maneira que ela sempre tenha uma cor que faça contraste com o fundo.
18. *[+Less]* Experimente com algumas bibliotecas de mixins importando os seus arquivos `.less` e aplicando em alguns elementos. Utilize o arquivo `more-18.less` e a página `more-18.html` ou outros arquivos usados no curso.
19. *[+Less]* Use o método `.toUpperCase()` do JavaScript para fazer com que as cores exibidas na página `more-19.html` apareçam em caixa-alta.