

**ORGANIZANDO  
MELHOR O CSS**



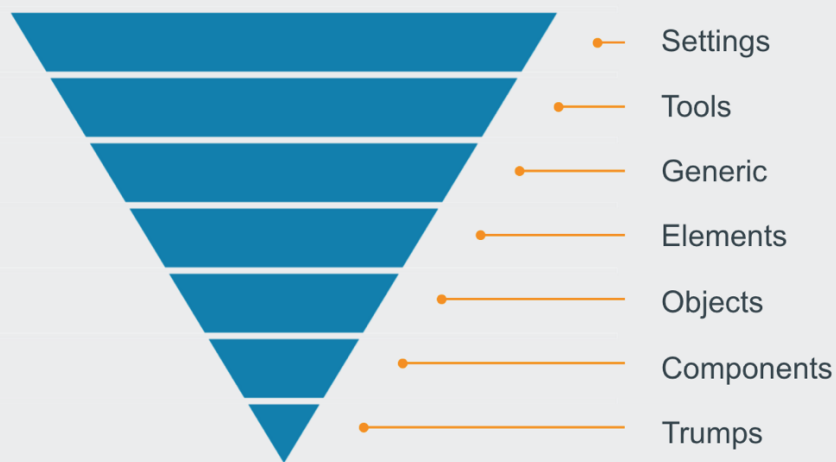
# O CSS

O CSS trabalha com a ideia de que, quanto mais específico for o seletor, maior a sua força. **O seletor com mais força sobrescreve o seletor mais fraco.**



# O ITCSS

O ITCSS (Inverted **T**riangle CSS) segue isso baseado na ideia do **Triângulo Invertido**. Quanto mais acima do triângulo você estiver, mais fraco será o seletor. Os seletores presentes em uma camada sempre sobrescrevem os seletores da camada de cima.

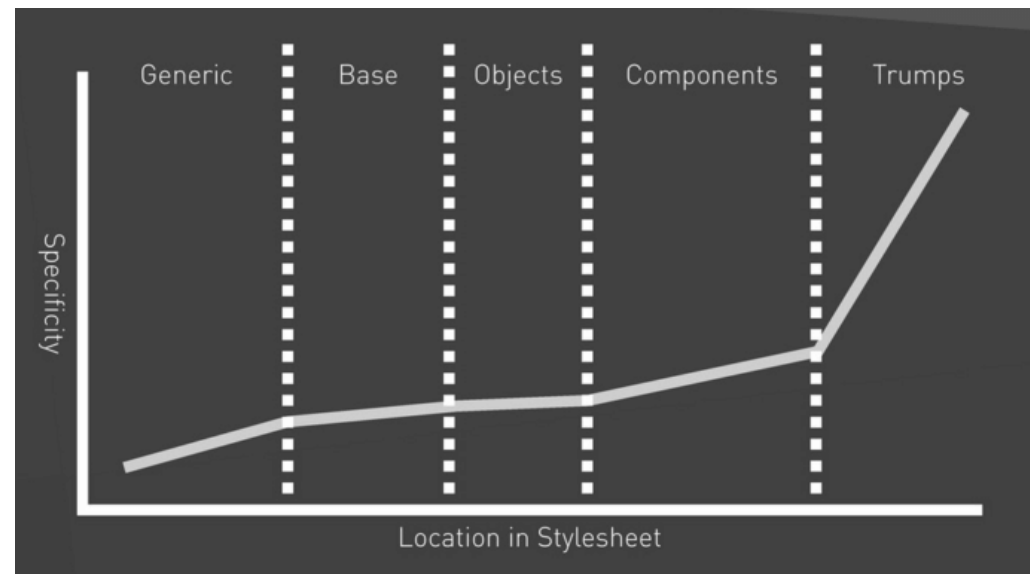
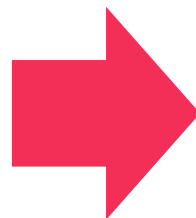
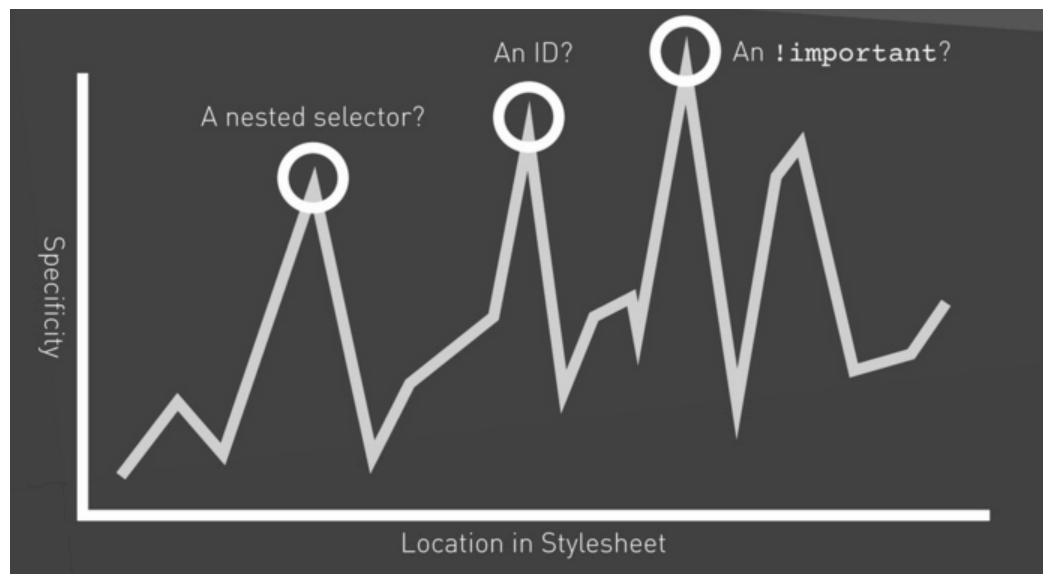




# ITCSS

A ideia do ITCSS é montar uma arquitetura que consiga mitigar esses pequenos problemas do CSS.

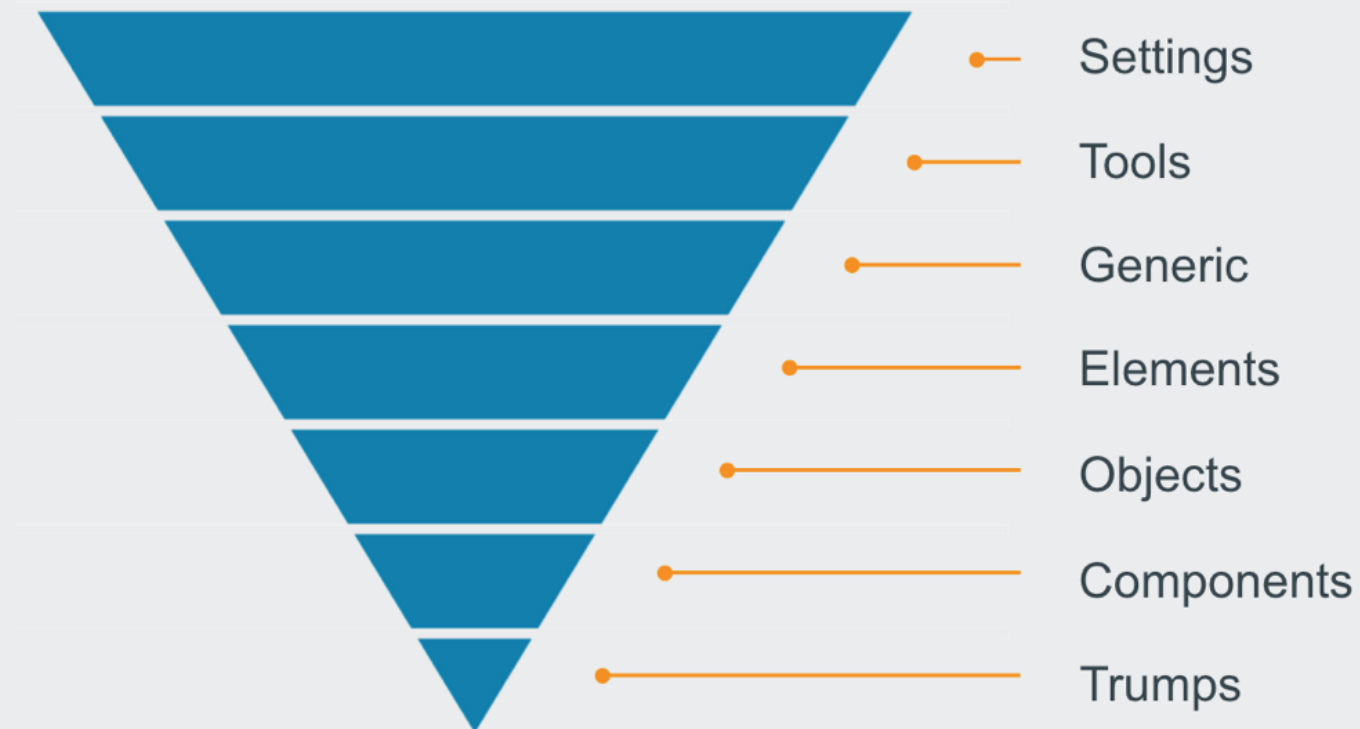
Que envolve basicamente visualizar todo o CSS em camadas, que montadas formam um triângulo de cabeça para baixo. Essa organização hierárquica ajuda a organizar o CSS da forma mais efetiva, diminuindo conflitos e sobrescritas.





# DIVISÕES DO ITCSS

Conforme você pode ver no gráfico acima, já temos as divisões, que são a base do ITCSS.





# 1.Settings

É onde declaramos as configurações do nosso projeto, como variáveis de medidas, cores, etc.

```
$color-ui: #BADA55;  
$spacing-unit: 10px;
```



## 2.Tools

É o lugar onde você vai guardar seus mixins e funções necessárias para a construção de seus layouts. Pode ser qualquer coisa, desde mixins de font-face, até mixins de animações, etc. Se não estiver utilizando pré-processadores como Sass ou Less, essa camada não precisa existir.

```
@mixin font-brand() {  
    font-family: "UI font", sans-serif;  
    font-weight: 400;  
}
```



## 3.Generic

Essa é a primeira camada que vai de fato aplicar CSS final e ela é destinada para as propriedades mais genéricas e com a menor especificidade possível. Em geral, é onde colocamos resets, box-sizing, etc.

```
* {  
  box-sizing: border-box;  
}
```





## 4.Base ou Elements

Base, também chamada de Elements. Essa camada serve para estilizarmos a parte mais básica dos elementos. Nada de IDs nem classes aqui, apenas seletores de elementos como os headings h1-h6, a, buttons, etc.

Normalmente utilizada para se fazer o CSS Reset.

```
ul {  
  list-style: square outside;  
}
```



## 5.Objects

Seguindo os princípios de OOCSS (CSS Orientado a Objetos), aqui é onde iremos ter nossos pequenos “objetos”, que nada mais são que pequenos pedaços da interface, em geral, padrões que se repetem por todo o site e que podem ter ou não uma camada visual por cima.

**Usamos apenas classes aqui.** É onde começamos a declarar a estrutura dos nossos elementos.

Aqui é onde fazemos os padrões de botões, listas, painéis, etc. Nesse momento, só é permitido também o uso de classes.

```
.base-list {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
  
  > .item {  
    padding: $spacing-unit;  
  }  
}
```



## 6.Components

**Aqui também só utilizamos classes.** É onde começamos a estilizar os nossos elementos de forma mais específica.

Enquanto nos objetos nós tentamos abstrair o máximo possível, para ter muitos objetos reutilizáveis e genéricos, aqui nós vamos ser específicos ao criar os componentes.

```
.products-list {  
  @extend font-brand();  
  border-top: 1px solid $color-ui;  
  
  > .item {  
    border-bottom: 1px solid $color-ui;  
  }  
}
```



## 7.Trumps

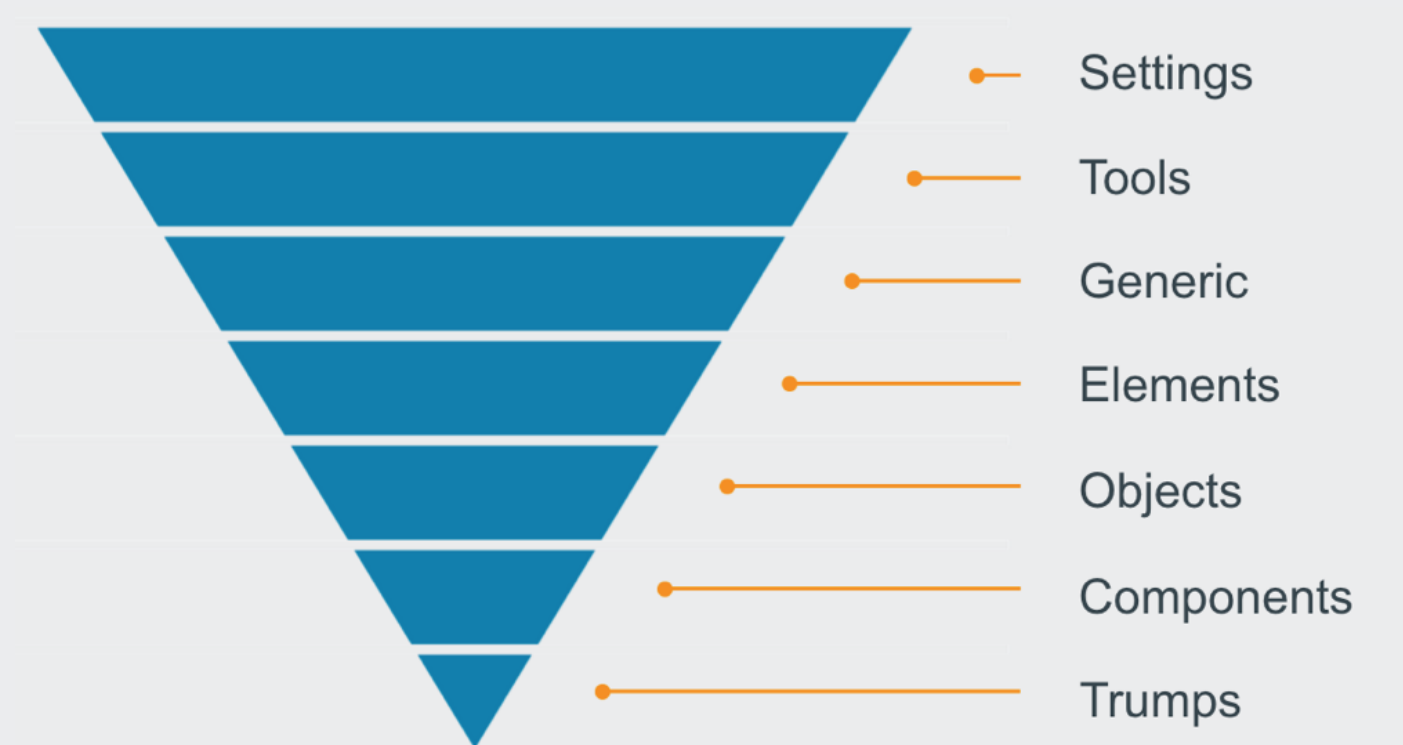
É a camada com maior especificidade. Aqui criamos pequenas classes com !important para sobrescrever certas regras.

```
.hidden {  
    display: none !important;  
}
```



# ITCSS

**Assim, temos a pirâmide invertida completa.**





# Organização dos arquivos

Independentemente se você utiliza ou não um pré-processador, é bom que no final todos os arquivos sejam minificados e unidos num único arquivo.

Para garantir a organização e a ordem de carregamento, fica a seguinte estrutura.

As pastas ficam numeradas apenas para manter a ordem quando formos ver no nosso editor ou IDE.

Dentro de cada pasta contém um arquivo principal, com o mesmo nome da pasta. Ele serve para carregar os demais arquivos.

```
|__ 1_settings
|   |__ _example_file.scss
|   |__ _1_settings.scss
|__ 2_tools
|   |__ _example_file.scss
|   |__ _2_tools.scss
|__ 3_generic
|   |__ _example_file.scss
|   |__ _example_file.scss
|   |__ _3_generic.scss
|__ 4_base
|   |__ _example_file.scss
|   |__ _example_file.scss
|   |__ _4_base.scss
|__ 5_objects
|   |__ _example_file.scss
|   |__ _example_file.scss
|   |__ _5_objects.scss
|__ 6_components
|   |__ _example_file.scss
|   |__ _example_file.scss
|   |__ _6_components.scss
|__ 7_trumps
|   |__ _example_file.scss
|   |__ _example_file.scss
|   |__ _example_file.scss
|   |__ _example_file.scss
|   |__ _7_trumps.scss
|__ styles.scss
```



# Organização dos arquivos

Na raiz há o arquivo `styles.scss`, que carrega cada um dos arquivos principais das pastas na ordem que as camadas devem seguir. Esse arquivo nunca será tocado.

```
@import "1_settings/1_settings";  
@import "2_tools/2_tools";  
@import "3_generic/3_generic";  
@import "4_base/4_base";  
@import "5_objects/5_objects";  
@import "6_components/6_components";  
@import "7_trumps/7_trumps";
```



# Organização dos arquivos

..
1_settings
2_tools
3_generic
4_base
5_objects
6_components
7_trumps
styles.scss