

# CSS

## Introdução Cascade Style Sheets

# O que faz o meu website mudar de aspecto?

# CSS - Cascade Style Sheet

Style Sheet Language

Usada para formatar a “apresentação” do nosso documento.

# CSS - Cascade Style Sheet

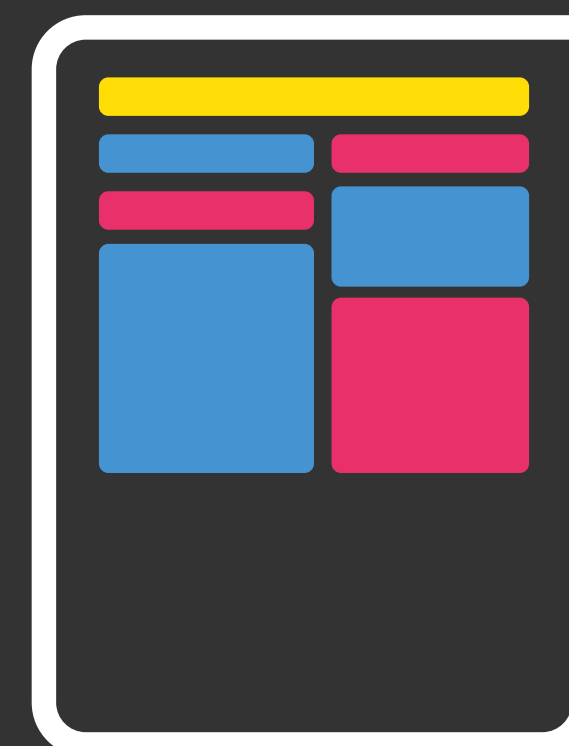
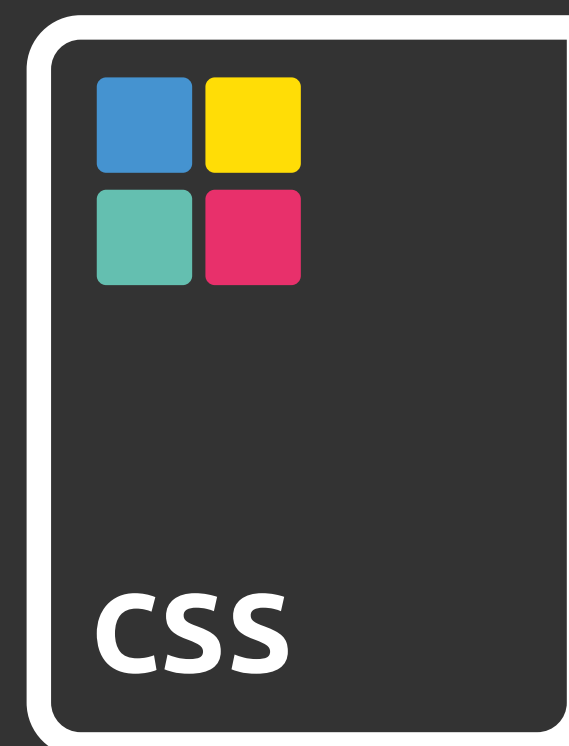
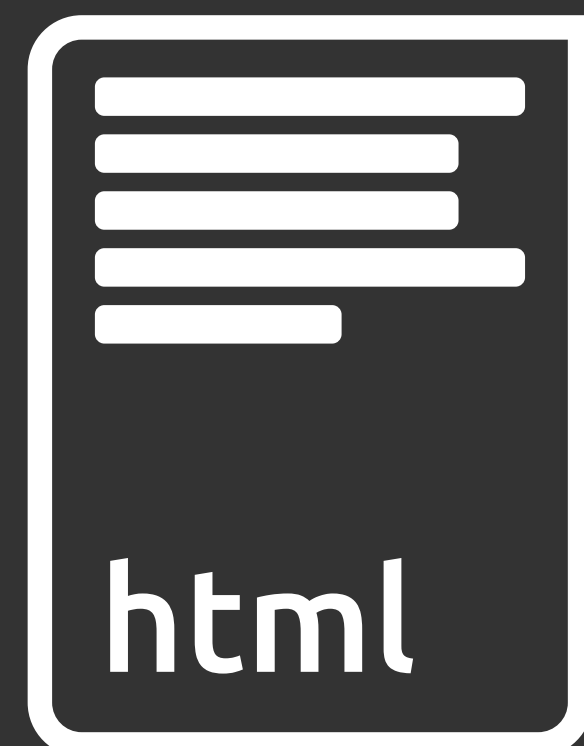
Descreve como os elementos devem / podem ser renderizados na página.

# Como elaborar um ficheiro .css

Descreve como os elementos devem / podem ser renderizados na página.

Primeiro de tudo, ficheiros .css,  
não trabalham sozinhos.

Precisam de conteúdo para estilizar.

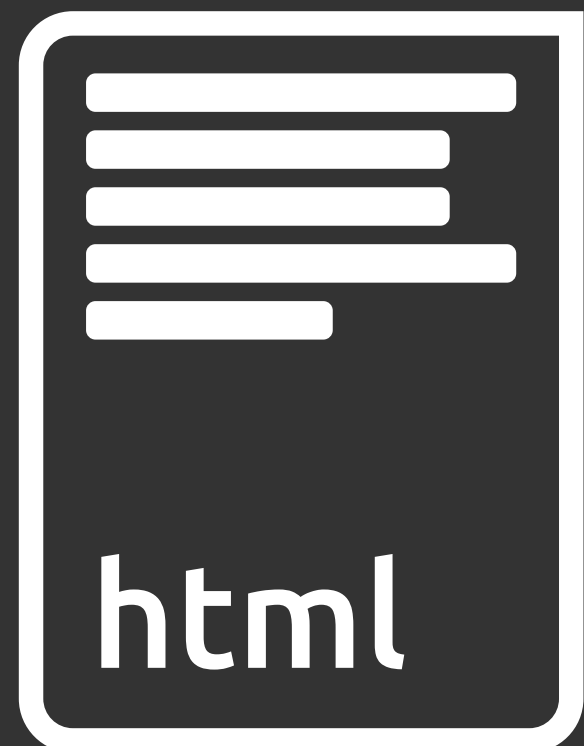


# Como elaborar um ficheiro .css

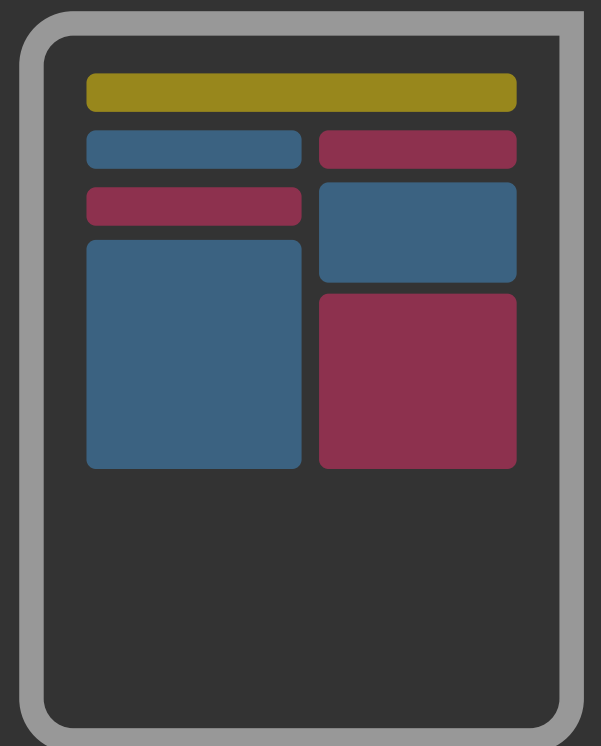
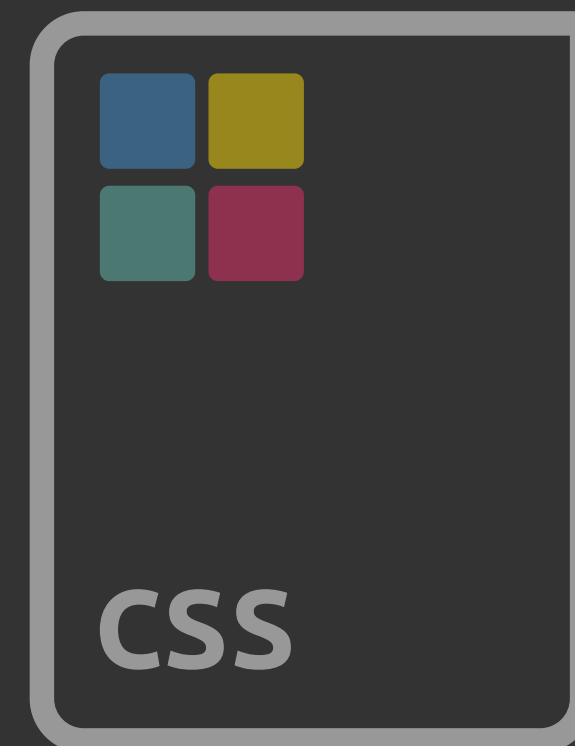
Descreve como os elementos devem / podem ser renderizados na página.

Primeiro de tudo, ficheiros .css,  
não trabalham sozinhos.

Precisam de conteúdo para estilizar.



Precisamos de construir o  
nosso ficheiro html para dividir  
lógica e geograficamente o nosso  
conteúdo.

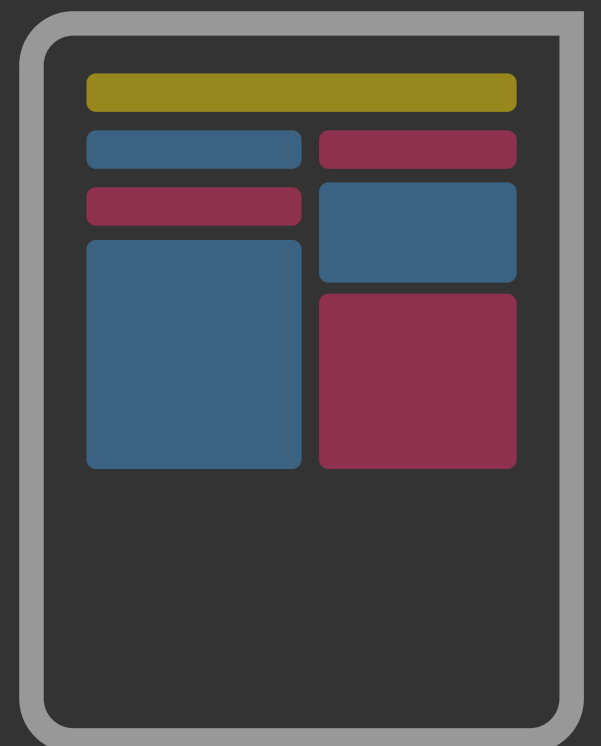
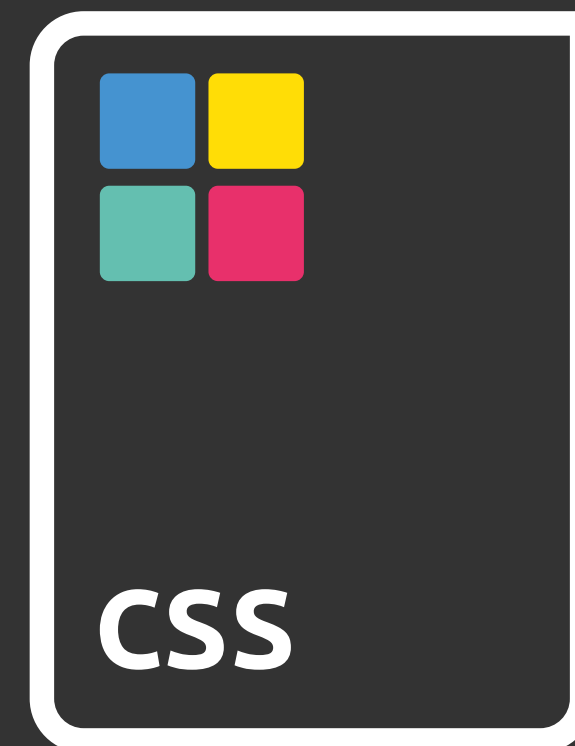


# Como elaborar um ficheiro .css

Descreve como os elementos devem / podem ser renderizados na página.

Primeiro de tudo, ficheiros .css,  
não trabalham sozinhos.

Precisam de conteúdo para estilizar.



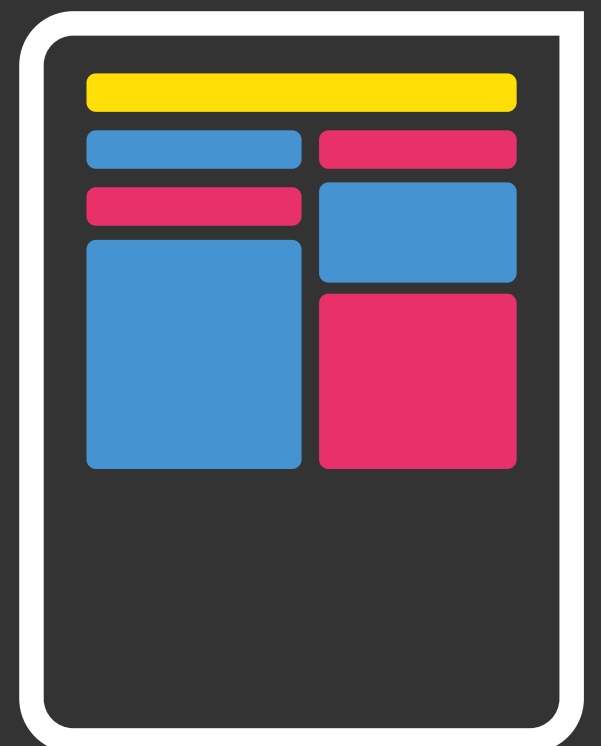
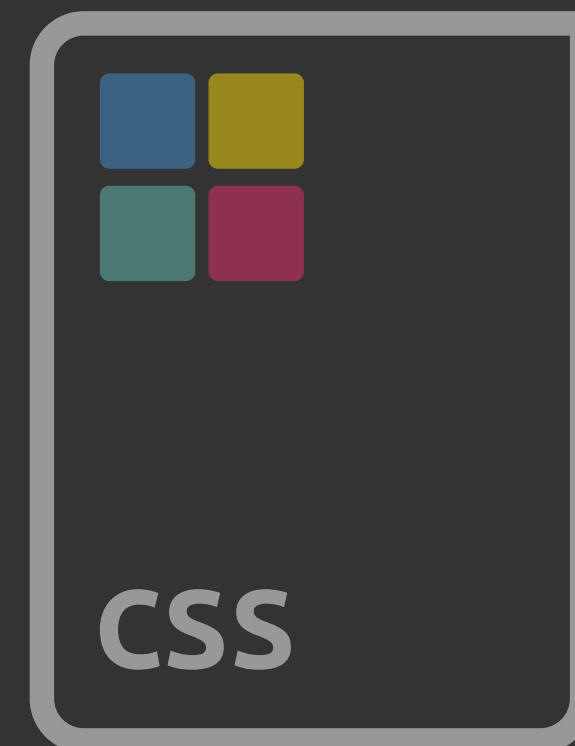
Ligamos o ficheiro .css ao ficheiro .html.  
Isto pode ser através de 3 métodos: CSS  
Externo CSS Interno CSS Inline

# Como elaborar um ficheiro .css

Descreve como os elementos devem / podem ser renderizados na página.

Primeiro de tudo, ficheiros .css,  
não trabalham sozinhos.

Precisam de conteúdo para estilizar.



O resultado será,  
o nosso conteúdo HTML renderizado  
consoante as regras determinadas no css



# Sem e Com CSS

I'm a button

# Sem e Com CSS

I'm a button

# Regras de CSS

## HTML Markup

```
<button>I'm a button</button>
```

## CSS Applied

```
button {  
  background-color: blue;  
  padding: 8px 16px;  
  border: 0;  
  color: white;  
  border-radius: 8px;  
  font-size: 16px;  
}
```

I'm a button

# Regras de CSS - Sintaxe

```
button {  
  background-color: blue;  
  padding: 8px 16px;  
  border: 0;  
  color: white;  
  border-radius: 8px;  
  font-size: 16px;  
}
```

# Regras de CSS - Sintaxe

```
selector {  
    property: value; /*declaration*/  
    property: value;  
    property: value;  
    property: value;  
    property: value;  
    property: value;  
}
```

# Regras de CSS - Sintaxe

```
/*declaration*/  
  
/* I'm a comment.  
I will be ignored */
```

# Princípios fundamentais de CSS

Cascata

Herança

Especificidade

# Princípios fundamentais de CSS

Cascata

Herança

Especificidade

Herança define o que acontece quando não existe valor de alguma propriedade num elemento.

Props. CSS podem ser categorizadas em 2 tipos:

Propriedades herdadas, que por default são computadas com o valor herdado pelo pai.

Propriedades não-herdadas, por default serão initial value.



# Princípios fundamentais de CSS

Cascata

Herança

Especificidade

A especificidade é o meio pelo qual os browsers decidem quais valores das propriedades CSS são os mais relevantes para um elemento e, portanto, serão aplicados.

A especificidade é baseada nas regras de correspondência que são compostas por diferentes tipos de seletores CSS.

# Princípios fundamentais de CSS

Cascata

1ª Regra

**div**

background: blue;



Herança

2ª Regra

**div**

background: orange;



Especificidade

# Princípios fundamentais de CSS

Cascata

Herança

Especificidade

**.pai**  
background: blue;  
color: white;

**.filho**  
background: orange;



# Princípios fundamentais de CSS

Cascata

Herança

Especificidade

# Selectores de CSS

**element**

Por Tags

**#id**

Por ids

**.class**

Por classe

**::before ou ::after**

Por pseudo selectores

**[type="text"]**

Por Atributos

**:hover ou :active**

Por pseudo classes

**etc**

entre vários outros

# Selectores de CSS

**element**

Por Tags

**#id**

Por ids

**\***

Selector universal

**.class**

Por classe

**::before ou ::after**

Por pseudo selectores

**+, >, ~**

Combinadores

**[type="text"]**

Por Atributos

**:hover ou :active**

Por pseudo classes

**etc**

entre vários outros

# Selectores de CSS

**element**

Por Tags

**#id**

Por ids

**\***

Selector universal

**!important**

O All Mighty destruídor de qualquer selector de css. Quando usado, nada mais importa.

**.class**

Por classe

**::before ou ::after**

Por pseudo selectores

**+, >, ~**

Combinadores

**[type="text"]**

Por Atributos

**:hover ou :active**

Por pseudo classes

**etc**

entre vários outros

# Resumindo

Especificidade é a unidade de medida do “quão específico” o selector é.

Alguns valores de propriedades serão herdadas pelo filho. Outros não.

Quando diferentes declarações tentam configurar um valor para o mesmo elemento / propriedade, este conflito tem que ser de alguma maneira resolvido.