

Principais propriedades do CSS

Agora que entendemos a ideia geral do CSS podemos aprender as principais e mais usadas propriedades dele, no momento vamos ver as principais mais simples e ao longo do curso veremos outras mais avançadas que te auxiliarão nas suas aplicações WEB.

Propriedade color

Como você deve imaginar a propriedade color altera a cor da fonte de um determinado elemento do HTML, seu uso é bastante simples, imagine que no HTML temos o seguinte:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <div>
    <p>Texto com cor diferenciada</p>
  </div>
</body>
</html>
```

Agora vamos selecionar o elemento <p> e alterar sua cor:

```
<style>
  p {
    color: blue;
  }
</style>
<body>
  <div>
    <p>Texto com cor diferenciada</p>
  </div>
</body>
```

Não há segredo nenhum envolvido, basta selecionar um elemento das formas que aprendemos e alterar a cor para qual quisermos, inclusive em breve teremos um assunto apenas sobre cores do CSS, mas por enquanto basta usar o nome das cores em inglês.

Propriedade background-color

Assim como o color, essa propriedade já deixa bem claro o que faz, altera a cor de fundo de um elemento, e como tudo no CSS → basta selecionar o elemento (ou elementos) desejado(s) e usar a propriedade:

```
<style>
  div {
    background-color: red;
  }
</style>
<body>
  <div>
    <p>Texto com cor diferenciada</p>
  </div>
</body>
```

Nesse caso acima a nossa div terá a cor de fundo alterada para vermelho.

Propriedade width

Essa propriedade define a largura de um elemento, em breve veremos mais sobre medidas do CSS por enquanto utilize medidas em pixels, junto da width temos a max-width que define a largura máxima de um elemento e a min-width que define a largura mínima de um elemento.

```
div {
  width: 300px;
}

// ou

div {
  max-width: 300px;
}

// ou

div {
  min-width: 300px;
}
```

Propriedade height

Essa é bem parecida com a width, porém ao invés de definir largura define a altura, e também podemos utilizar max-height e min height:

```
div {
  height: 300px;
}

// ou

div {
  max-height: 300px;
}

// ou

div {
  min-height: 300px;
}
```

Propriedade margin

A propriedade margin define um espaçamento externo ao elemento, no caso se você quiser separar os elementos evitando que fiquem colados, uma boa é usar o margin:

```
div {
  margin: 32px;
}
```

Nesse caso acima o elemento receberá 32px de margem por volta dele todo, porém podemos definir um valor para cada lado do elemento, podemos usar margin-top, margin-right, margin-bottom e margin-left:

```
div {
  /* Cima: */
  margin-top: 32px;
  /* Direita: */
  margin-right: 12px;
  /* Baixo: */
  margin-bottom: 10px;
  /* Esquerda: */
  margin-left: 20px;
}
```

Ou ainda temos outra opção de fazer o mesmo que fizemos acima de uma maneira mais simples:

```
div {
  margin: 32px 12px 10px 20px;
}
```

O margin (e outras propriedades que veremos em breve) segue uma regra onde ele define os valores na ordem: top → right → bottom e left por último, e ainda podemos fazer da seguinte forma:

```
div {  
  margin: 32px 16px;  
}
```

Nesse caso o elemento receberá 32px na parte de cima e de baixo e 16px nas laterais.

Propriedade padding

É bem parecida com a margin, a única coisa que muda é que ao invés de espaçamento externo é espaçamento interno, então o espaço fica do conteúdo até a borda do elemento:

```
div {  
  padding: 32px;  
}
```

As regras são as mesmas do margin.

Propriedade border

Com essa propriedade podemos inserir bordas nos elementos selecionados, podemos definir espessura, cor, formato, podemos também retirar bordas de elementos que possuem bordas nativamente, enfim, várias possibilidades:

```
button {  
  border: 2px solid red;  
}
```

Acima definimos que a espessura da borda do botão será de 2px, lisa/contínua (solid) e na cor vermelha, se quiséssemos retirar a borda que o botão possui nativamente bastaria fazer o seguinte:

```
button {  
  border: none;  
}
```

Isso vale para as outras propriedades também, caso queira por exemplo retirar toda margin de um elemento basta usar:

```
div {  
  margin: none;  
  // ou  
  margin: 0;  
}
```

Propriedade border-radius

Essa propriedade faz o arredondamento das bordas de um elemento, podemos passar valores em pixels, e ainda se quisermos por exemplo arredondar uma imagem podemos usar o valor 50%:

```
div {  
  border-radius: 8px;  
}  
  
img {  
  border-radius: 50%;  
}
```

Propriedade font-size

Essa propriedade define o tamanho da fonte de um texto, por padrão os textos possuem um tamanho que o navegador define, porém podemos alterar para o tamanho que quisermos:

```
p {  
  font-size: 32px;  
}
```

```
}
```

Propriedade font-weight

Essa propriedade altera o "peso" da fonte, podemos deixar as letras mais finas ou mais grossas como o negrito, por exemplo:

```
a {  
  font-weight: bold;  
}
```

"bold" deixa como o negrito, "normal" mantém uma fonte normal de fato, podemos também trabalhar com números:

```
p {  
  font-weight: 300;  
  ou  
  font-weight: 900;  
}
```

Quanto maior o número mais "pesada" a fonte fica.

Propriedade background-image

Essa insere uma imagem ao fundo de um elemento, ao fazermos isso devemos nos atentar em como essa imagem irá preencher o fundo do elemento:

```
div {  
  background-image: url('caminho da imagem');  
  background-repeat: no-repeat;  
  background-size: cover;  
  /* ou */  
  background-size: contain;  
  /* ou */  
  background-position: center center;  
}
```

Passamos url ('caminho da imagem') para a imagem ser inserida ao fundo do elemento, background-repeat: no-repeat para que a imagem não se repita ao fundo do elemento, background-size temos algumas possibilidades, cover dá um zoom na imagem para que a imagem preencha todo o elemento, contain deixa a imagem do tamanho que ela é dentro do elemento, podemos ainda definir tamanhos fixos para o background-image:

```
div {  
  background-size: 300px 100px;  
}
```

Onde o primeiro valor é a largura (width) e o segundo valor é a altura(height).

Propriedade text-align

Como você deve imaginar, essa propriedade faz o alinhamento de textos, podemos centralizar, justificar, enfim, algumas possibilidades:

```
p {  
  text-align: center;  
  /* ou */  
  text-align: justify;  
  /* ou */  
  text-align: left;  
  /* ou */  
  text-align: right;  
}
```

Visibility

"A propriedade CSS de visibilidade pode mostrar ou ocultar um elemento sem afetar o layout de um documento (ou seja, o espaço é criado para elementos independentemente de serem visíveis ou não). A propriedade também pode ocultar linhas ou colunas em um <table>" - [MDN Web Docs](#).

Podemos esconder um elemento de forma bem simples:

```
p {  
  visibility: hidden;  
}
```

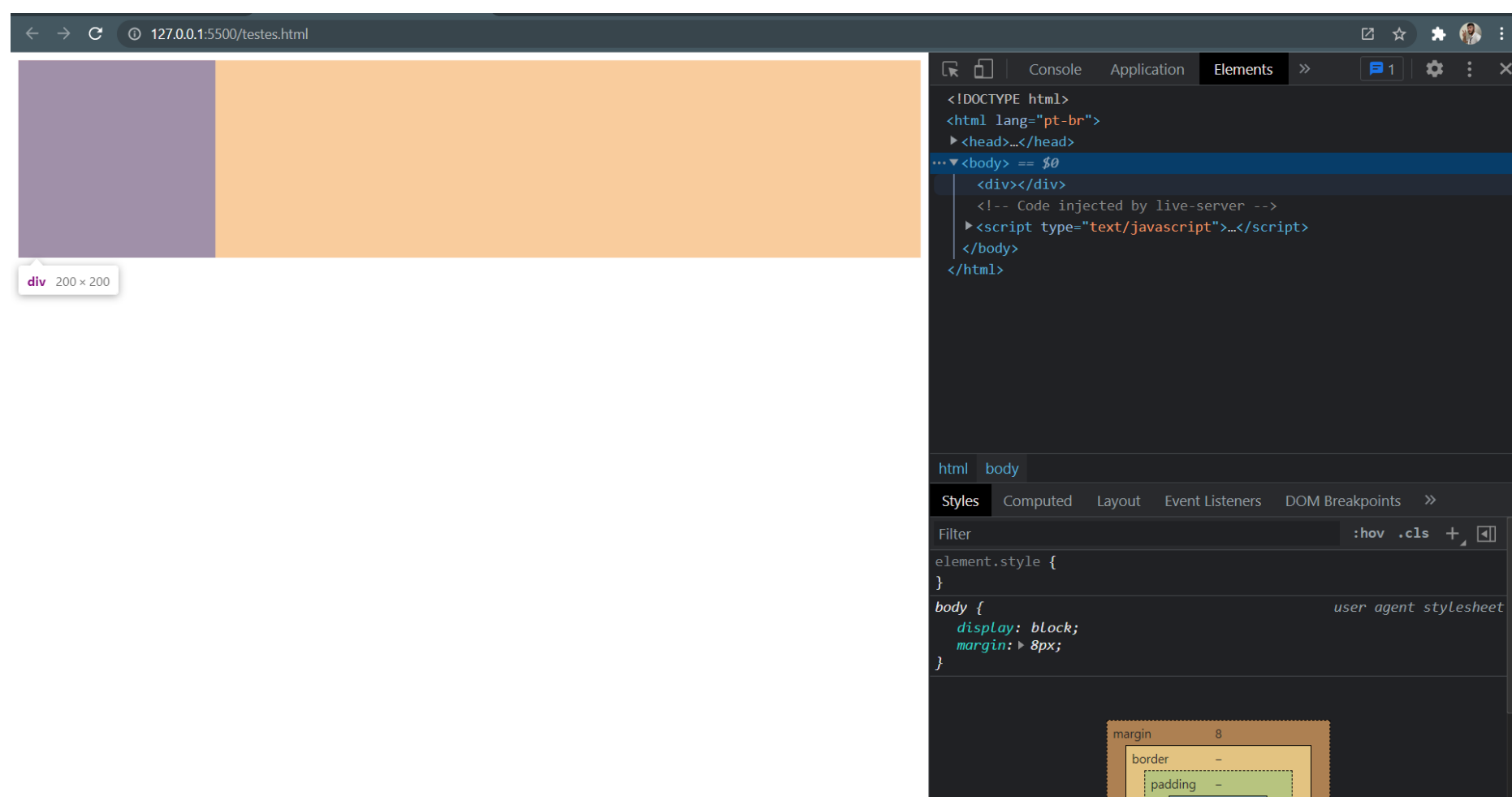
Ou então fazer com que ele seja visível de novo em tela (caso ele esteja escondido antes):

```
visibility: visible;
```

Display

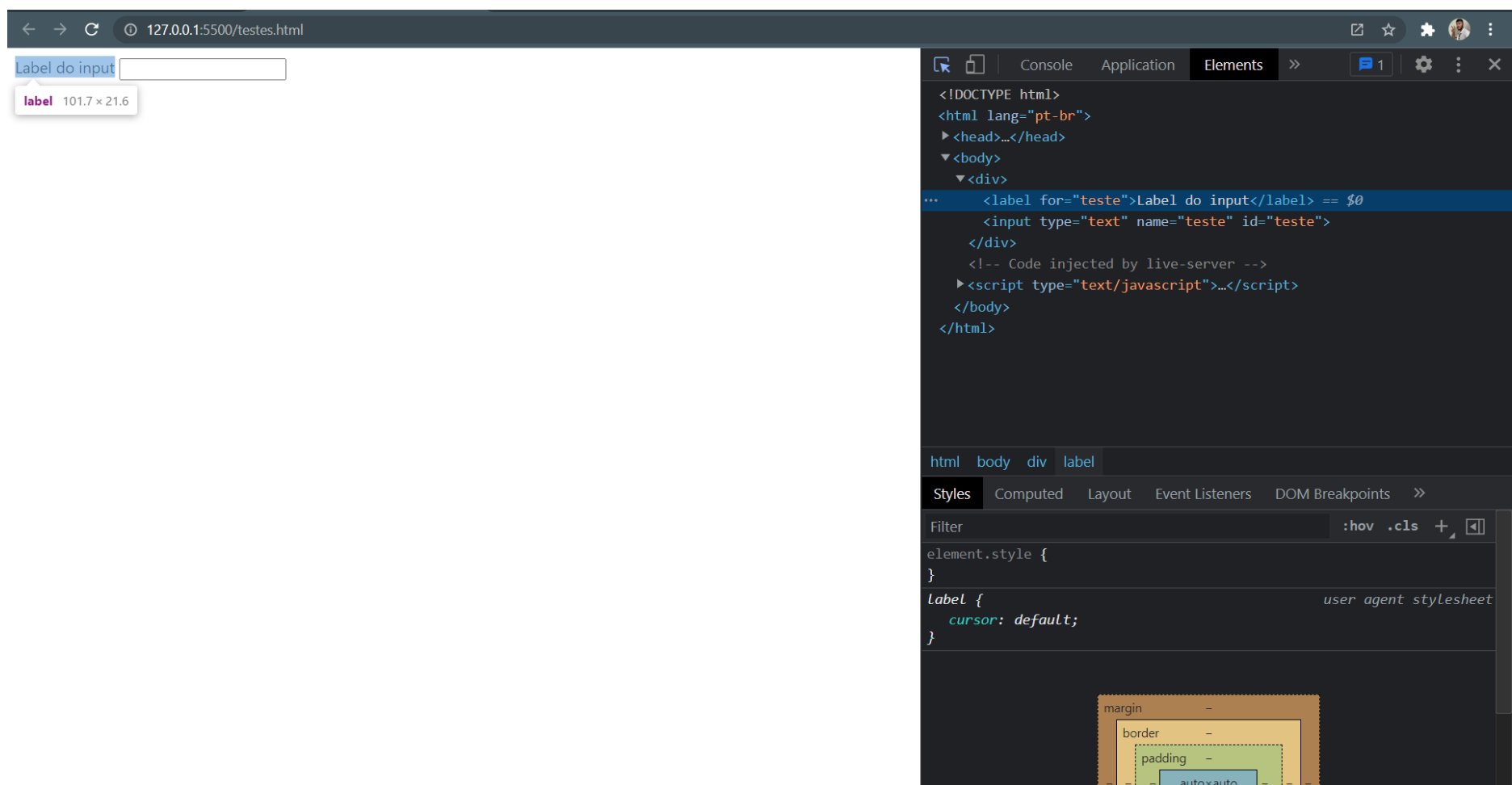
Essa é uma propriedade bem importante que pode te salvar muito se você a entender, ela é um pouco confusa de entender no início, e até desenvolvedores front-end mais experiente podem ter dúvidas a respeito dela, porém vamos aqui resumir o que é mais importante sobre ela.

Um elemento <div> que é largamente utilizado na WEB possui por padrão do navegador o **display block**, isso quer dizer que o elemento <div> ocupa um bloco todo na largura da tela. Abrindo o Dev Tools e inspecionando a div isso fica mais claro:



Veja que ela tem um tamanho e o resto em laranja é uma margin, e aí vai uma dica: se você definir a margin com o valor **auto** a <div> fica centralizada na horizontal, e caso queira ela centralizada na vertical também é preciso definir a altura do elemento pai dela com 100vh.

Agora temos aqui outro elemento que nativamente possui o display inline, que é o caso do <input> e do <label> por exemplo, entre outros, veja como ele é colocado em tela:



Tanto o `<label>` quanto o `<input>` ao lado são do tipo inline, por isso ficam um na frente do outro, no caso da `<div>` ficam uma abaixo da outra.

Com o `display` você também pode esconder ou mostrar elementos do HTML, para isso basta definir um elemento com propriedade `display` com o valor `none` ou `initial`.

Ao longo do curso vamos aprendendo novas propriedades.