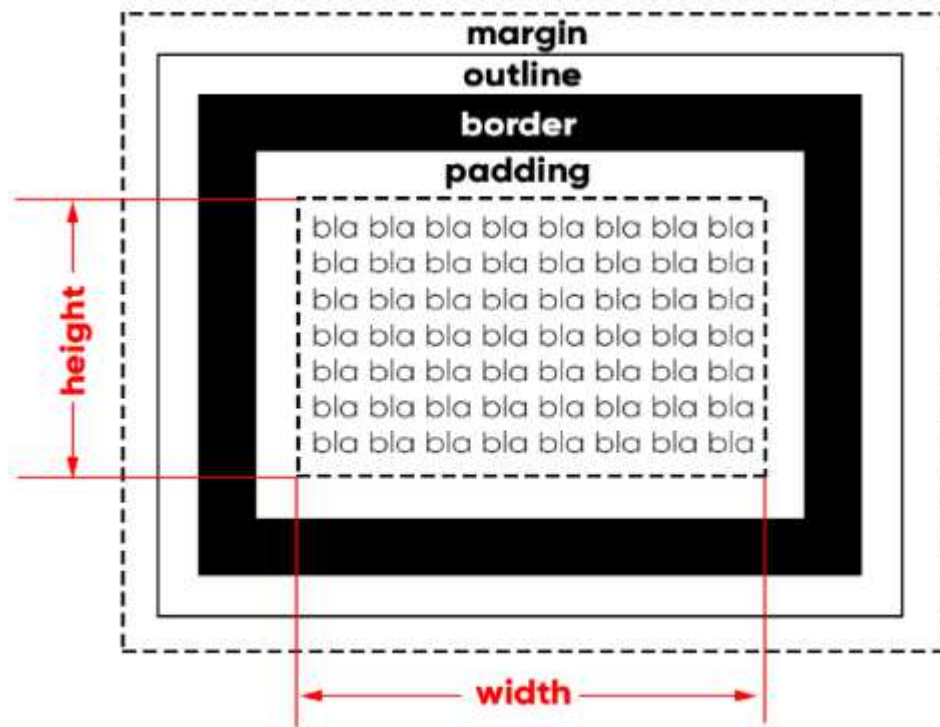


MODELO DE CAIXAS

De forma simples e objetiva, a grande maioria dos elementos HTML que temos no nosso site são como caixas (“box model”). Elas são *containers* quer armazenam conteúdos ou até mesmo outras caixas.

ANATOMIA DE UMA CAIXA



Tudo começa a partir do **conteúdo** (content), que representamos acima com o bla bla bla... Por padrão, toda caixa é composta apenas pelo conteúdo e não possui padding, nem border, nem outline e nem margin. Uma exceção curiosa é o elemento `<body>` que já vem com uma margin de 8px.

Todo conteúdo possui uma **largura** (width) e uma **altura** (height) e a esse conjunto de propriedades, damos o nome de box-size (tamanho da caixa). O tamanho da caixa não inclui as medidas de padding, border, outline e margin.

Depois do conteúdo e de seu tamanho, vamos nos focar na **borda** que fica em volta dele. Ela pode ter uma espessura, uma cor e um formato.

Entre a borda e o conteúdo - da borda para dentro - temos o **preenchimento** (padding) e da borda para fora, temos a **margem** (margin).

Entre a margem e a borda, podemos determinar o **contorno** (outline) que é muito pouco utilizado, mas existe. Ele é um traçado visual que podemos criar fora da borda e o cálculo da sua espessura faz parte da margem estabelecida.

Vamos criar um exemplo simples para exemplificar todos esses componentes, configurando as propriedades do modelo de caixa de um título `<h1>`. Acompanhe o trecho de código a partir das definições de estilo.

```
7  <style>
8    h1 {
9      width: 300px;
10     height: 50px;
11     background-color: lightgray;
12     border-width: 10px;
13     border-style: solid;
14     border-color: red;
15     padding: 20px;
16     outline-width: 30px;
17     outline-style: solid;
18     outline-color: blue;
19     margin: 50px;
20   }
21 </style>
22 </head>
23 <body>
24   <h1>Exemplo de Caixa</h1>
25 </body>
```



Olhando de perto, podemos analisar as medidas configuradas no código apresentado. As medidas de `height` e `width` (300x50) são medidas apenas pela parte pontilhada do conteúdo.

A `border` de 10px ficou em vermelho e o `outline` de 30px ficou em azul. O `padding` de 20px fica da borda para dentro e a `margin` de 50px fica da borda para fora.

Sendo assim, a medida total que essa caixa vai ocupar é de $50 + 10 + 20 + 300 + 20 + 10 + 50 = \mathbf{460\text{px de largura}}$ e $50 + 10 + 20 + 50 + 20 + 10 + 50 = \mathbf{210\text{px de altura}}$.

Dá pra simplificar?

As configurações de borda e contorno também possuem *shorthands* para simplificar o código anterior. A ordem para as duas configurações é sempre a mesma para as duas shorthands: largura (-width), estilo (-style) e cor (-color).

MODO COMPLETO	SHORTHAND
<pre>border-width: 10px; border-style: solid; border-color: red; outline-width: 30px; outline-style: solid; outline-color: blue;</pre>	<pre>border: 10px solid red; outline: 30px solid blue;</pre>

Preenchimento e margem personalizados

Todo elemento de caixa possui quatro valores para padding e quatro para margin, sempre nessa mesma ordem: superior (-top), direita (-right), inferior (-bottom), esquerda (-left). Quando colocamos um único valor de dimensão para o preenchimento ou margem, esse mesmo valor é aplicado simetricamente a todas as direções, mas também podemos fazer códigos como:

MODO COMPLETO	SHORTHAND
<pre>padding-top: 10px; padding-right: 15px; padding-bottom: 20px; padding-left: 25px; margin-top: 0px; margin-right: 10px; margin-bottom: 20px; margin-left: 30px;</pre>	<pre>padding: 10px 15px 20px 25px; margin: 0px 10px 20px 30px;</pre>

Margens no automático

Um recurso que também vai ser muito usado em nossos exercícios é a centralização de blocos. Para que isso seja feito, devemos pedir que o navegador calcule automaticamente as margens da esquerda e da direita para que o bloco seja colocado no meio do navegador, independente do tamanho da janela.

Para centralizar uma caixa, use a seguinte declaração no seu seletor:

```
margin: auto;
```



Caixa do tipo block-level

Um elemento dito *block-level* sempre vai se iniciar em uma nova linha e vai ocupar a largura total do elemento onde ele está contido. Se não estiver contido em nenhuma outra caixa, ele vai ocupar 100% da largura do <body>.

O elemento *block-level* mais conhecido é o <div> e suas variações semânticas modernas da HTML5, como <main>, <section>, <aside>, etc.

Na lista a seguir, coloquei alguns elementos HTML que são block-level:

<address>	<article>	<aside>	<blockquote>	<canvas>	<dd>
<div>	<dl>	<dt>	<fieldset>	<figcaption>	<figure>
<footer>	<form>	<h1> - <h6>	<header>	<hr>	
<main>	<nav>	<noscript>		<p>	<pre>
<section>	<table>	<tfoot>		<video>	

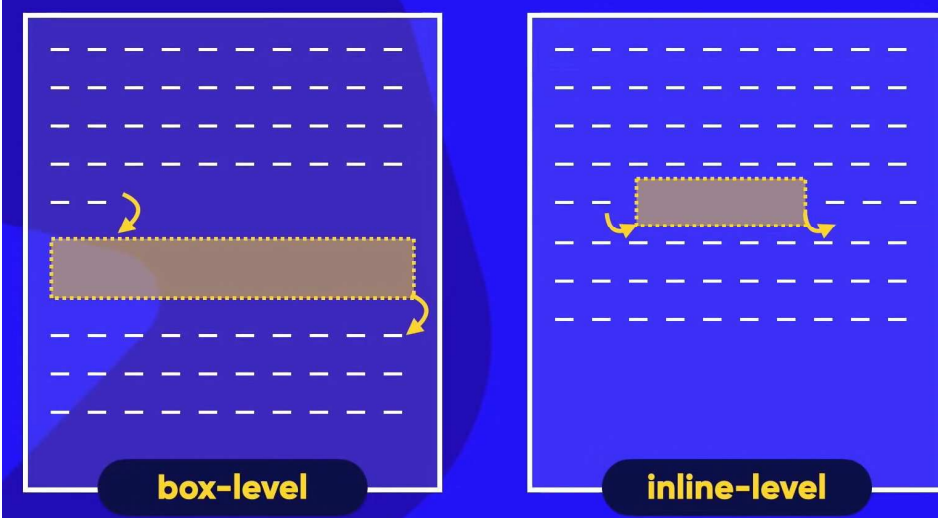
Caixa do tipo inline-level

Um elemento do tipo *inline-level* não vai começar em uma nova linha, e sim no ponto exato onde foram definidos. E a largura dele vai ocupar apenas o tamanho relativo ao seu conteúdo.

Abaixo, listei alguns elementos *inline-level* usados pela HTML:

<a>	<abbr>	<acronym>		<bdo>	
<button>	<cite>	<code>	<dfn>		<i>
	<input>	<kbd>	<label>	<map>	<object>
<output>	<q>	<samp>	<script>	<select>	<small>
		<sub>	<textarea>	<tt>	<var>

Tipos de Caixa



Através das linhas de códigos é possível transformar um elemento inline em box e vice versa. Usando código ***display: block*** para box e ***display: inline*** para inline.

GROUPING TAGS e SEMANTIC TAGS

HEADER: Cria áreas relativas à cabeçalhos. Pode ser o cabeçalho principal de um site ou até mesmo o cabeçalho de um artigo ou seção. Normalmente inclui títulos h1 – h6 e subtítulos. Podem também conter menus de navegação.

NAV: Define uma área que possui os links de navegação pela estrutura de páginas que vão compor o website. Um `<nav>` pode estar dentro de um `<header>`.

MAIN: É um agrupador usado para delimitar o conteúdo principal do nosso site. Normalmente concentra as seções, artigos e conteúdos periféricos.

SECTION: Cria seções para sua página. Ela pode conter o conteúdo diretamente no seu corpo ou dividir os conteúdos em artigos com conteúdos específicos. Segundo a documentação W3C, “uma seção é um agrupamento temático de conteúdos, tipicamente com um cabeçalho”.

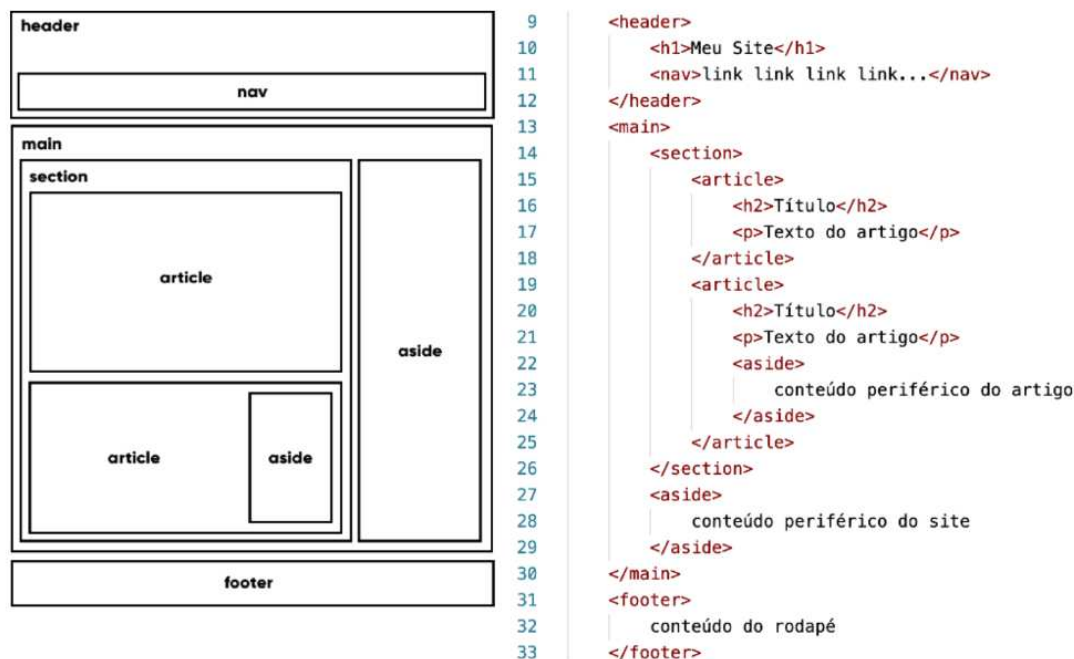
ARTICLE: Um artigo é um elemento que vai conter um conteúdo que pode ser lido de forma independente e dizem respeito a um mesmo assunto. Podemos usar um `<article>` para delimitar um post de blog ou fórum, uma notícia, etc.

ASIDE: Delimita um conteúdo periférico e complementar ao conteúdo principal de um artigo ou seção. Normalmente um conteúdo `<aside>` está posicionado ao lado de um determinado texto ou até mesmo no meio dele, exatamente como fizemos no bloco de texto apresentado anteriormente, falando sobre “MULTIPLOS NIVEIS”.



MÚLTIPLOS NÍVEIS: A sua criatividade e planejamento vai definir a estrutura do seu site. Sendo assim, é possível ter um ou mais `<article>` dentro de uma `<section>` ou até mesmo criar `<section>` dentro de um `<article>`. Não existem limitações quanto a isso.

FOOTER: Cria um rodapé para o site inteiro, seção ou artigo. É um conteúdo que não faz parte diretamente do conteúdo nem é um conteúdo periférico, mas possui informações sobre autoria do conteúdo, links adicionais, mapa do site, documentos relacionados.



Analise o diagrama do lado esquerdo e o código do lado direito da imagem acima. Veja a hierarquia entre os elementos e quais deles estão dentro um do outro.

SOMBRAS NAS CAIXAS

`box-shadow: 3px 5px 4px ■black;`



Veja que uma sombra bem forte já pode ser percebida, assim que adicionamos a propriedade `box-shadow` e seus quatro valores. A ordem é sempre essa:

1. **Deslocamento horizontal** (*h-offset*): quanto a sombra vai andar para o lado direito (valores negativos causam deslocamento para a esquerda)
2. **Deslocamento vertical** (*v-offset*): quanto a sombra vai andar para baixo (valores negativos causam deslocamento para cima)
3. **Embaçamento** (*blur*): quanto a sombra vai se espalhar pelo fundo
4. **Cor** (*color*): cor da sombra. É possível usar transparência.



MUITO CUIDADO! Não exagere no uso de sombras, pois elas podem tornar o seu efeito visual muito pesado. Evite também usar sombras coloridas. Olhe ao seu redor e perceba que as sombras são sempre pretas. Use cores `rgba()` para obter uma transparência que cause efeitos mais suaves.

BORDAS DECORADAS

As bordas das caixas não precisam ser sempre retangulares e podem ter alguns detalhes especiais.

VÉRTICES ARREDONDADOS

```
border-radius: 10px 20px 30px 40px;
```

Usando sombras

Vamos aprender a usar sombras com esse exemplo simples.

Esse elemento segue o mesmo padrão das margins. Isso significa que ela aceita shorthands. Podemos escrever um único valor, que assim seria imposto aos quatro cantos, ou dois que seria impostos horizontalmente.

```
border-radius: 10px 30px;
```

Usando sombras

Vamos aprender a usar sombras com esse exemplo simples.