

Introdução

Para obter mais informações sobre o conteúdo, recomendo aqui as fontes de pesquisa utilizadas.

- **GitHub**

Acesse o conteúdo do repositório público, lá o Guanabara disponibiliza todo material usado em aula.



- **Módulo 02 - Curso em Vídeo**

Assista a playlist completa do curso de HTML5 e CSS3.



- **W3schools**

W3Schools é um site educacional voltado ao aprendizado de tecnologias web. Seu conteúdo inclui tutoriais e referências relacionadas a HTML, CSS e Java Script, que irão ser estudados ao longo do curso.



- **W3C**

É a organização responsável pelas normas técnicas da World Wide Web. As tags obsoletas e outras informações são encontradas no item 15.



- **MDN Web Docs**

MDN Web Docs é uma fonte de documentação para desenvolvedores, mantida com o apoio de uma comunidade de desenvolvedores e escritores técnicos e hospedando muitos documentos sobre uma grande variedade de assuntos como: HTML5, JavaScript, CSS, Web APIs, Node.js, WebExtensions e MathML.



Ferramentas

- **VSCode**

O Visual Studio Code é um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS.



- **Gimp**

GIMP (GNU Image Manipulation Program) é um programa de código aberto voltado principalmente para criação e edição de imagens, e em menor escala também para desenho vetorial.



- **HandBrake**

HandBrake é um programa multiplataforma e multitarefa em código aberto (licenciado em GPL) de conversão de arquivos de vídeo e DVD e Blu-ray para MPEG-4, disponível para Mac OS X, Linux e Windows.



- **Adobe Color**

Crie paletas de cores com a roda de cores ou imagem e navegue por milhares de combinações de cores na comunidade do Adobe Color.



- **Google Fonts**

Google Fonts é uma biblioteca com mais de 800 fontes livres licenciadas, um diretório web interativo para navegar na biblioteca, e APIs para usar convenientemente as fontes através de CSS e Android.

A blue rectangular button with the text "Google Fonts" in white.

- **DevTools**

Chrome DevTools é um conjunto de ferramentas de desenvolvedor da web integradas diretamente no navegador Google Chrome.



- **Mockflow**

É a ferramenta líder para projetar planos de interface do usuário para sites e aplicativos.



Extensões

- **Web Developer**

Web Developer é uma extensão para navegadores da web baseados em Mozilla que adiciona ferramentas de edição e depuração para desenvolvedores da web.



- **ColorZilla**

O ColorZilla permite obter uma leitura de cores em qualquer ponto do navegador, ajustando rapidamente essa cor e colando-a em outro programa, como o Photoshop.



- **Fonts Ninja**

O Fonts Ninja permite capturar fontes utilizadas em sites e também oferece os detalhes como cor, tamanho, entre outros.



- **Window Resizer**

O Window Resizer redimensiona a janela do navegador para emular várias resoluções.



HTML em Cores

```
<body> /*Menu*/
  <main>
    <article>
      <p>Seletores /*6
Cores em CSS 7
Gradiente em CSS 8
Fontes em CSS 12
Fontes Externas 16
Alinhamento de Textos 18
Seletores Personalizados 19
Modelo de Caixas 23
Tipos de Caixa 26
Grouping Tags 28
Sombras nas Caixas 29
Caixas com Vértices 30
Variáveis em CSS 32
Responsividade para Sites 33
Configurando Textos e Linhas 35
Personalizando as Listas 36
Vídeo Responsivo</p> 37*/
    </article>
    <aside>
      <p>Escreva mais Rápido /*7
Identificando Fontes 18
Envelopamento Cascata 19
margin:auto 26
Editor de Sombras 30
Seletor * CSS</p> 33*/
    </aside>
  </main>
</body>
```

Seletores

Diagram illustrating the structure of a CSS rule:

```
h1 {color:blue; font-size:12px;}
```

The diagram labels the components of the rule:

- Seleção** (Selection): `h1`
- Declaração** (Declaration): `{color:blue; font-size:12px;}`
- Propriedade** (Property): `color` and `font-size`
- Valor** (Value): `blue` and `12px`

O seletor aponta para o elemento HTML que você deseja estilizar.

O bloco de declaração contém uma ou mais declarações separadas por ponto e vírgula.

Cada declaração inclui um nome de propriedade CSS e um valor, separados por dois pontos.¹

Várias declarações CSS são separadas por ponto-e-vírgula e os blocos de declaração são cercados por chaves.

1 CSS Properties

ESCREVA MAIS RÁPIDO

O VSCode corrige automaticamente o que você está escrevendo. Não se preocupe com espaços ou traços, apenas dê Enter.

```
<style>
  body {
    text-align:
  }
</style>
```

- text-align
- text-align-last
- text-anchor

Cores em CSS

Exploramos quatro maneiras diferentes de definir uma cor em CSS.²

CSS	Descrição	Exemplo
Nome	Cor	Blue
HEX	Hexadecimal	#rrggbb
RGB	Red, Green, Blue	(0, 0, 255)
HSL	Hue, Saturation, Lightness	(0, 100%, 50%)

HEX

RR (vermelho), GG (verde) e BB (azul) são inteiros hexadecimais entre 00 e FF especificando a intensidade da cor.

RGB

Cada parâmetro (vermelho, verde e azul) define a intensidade da cor como um número inteiro entre 0 e 255.

HSL

- H O matiz é um grau na roda de cores de 0 a 360. 0 é vermelho, 120 é verde, 240 é azul.
- S A saturação é um valor percentual; 0% significa um tom de cinza e 100% é a cor total.
- L A leveza também é uma porcentagem; 0% é preto, 100% é branco.

² [CSS Colors](#)

O atributo **style**, dentre outros estilos, define uma cor para o elemento.

```
<body>
  <h2 style= "background-color: blue; color: white;">Cores</h2>

  <h2 style= "background-color: #0000ff; color: #ffffff;">HEX</h2>

  <h2 style= "background-color: rgb(0, 0, 255); color: rgb(255, 255, 255);">RGB</h2>

  <h2 style= "background-color: hsl(240, 100%, 50%); color: hsl(0, 0%, 100%);">HSL</h2>
</body>
```

Gradiente em CSS

Os gradientes CSS permitem que você exiba transições suaves entre duas ou mais cores especificadas.³

Aplicamos folhas de estilo no documento HTML com a tag **<style>**. E surgiu a propriedade: **background-image** – Imagem de fundo, que nesse caso servirá para aplicar o gradiente.



As propriedades descrevem as características de um objeto.

³ CSS Gradients

Gradiente Linear

```
background-image: linear-gradient(direção, cor1, cor2, ...);
```

Para criar um gradiente linear, devemos definir pelo menos duas cores.

Também, definiremos um ponto inicial e uma direção (ou um ângulo) junto com o efeito de gradiente.

▲ Top to bottom



*De cima para baixo
(padrão)*

```
<style>
  body {
    background-image: linear-
      gradient(blue, yellow);
  }
</style>
```

▲ Left to right



*Da esquerda para
direita*

```
<style>
  body {
    background-image: linear-gradient(to
      right, blue, yellow);
  }
</style>
```

▲ Diagonal



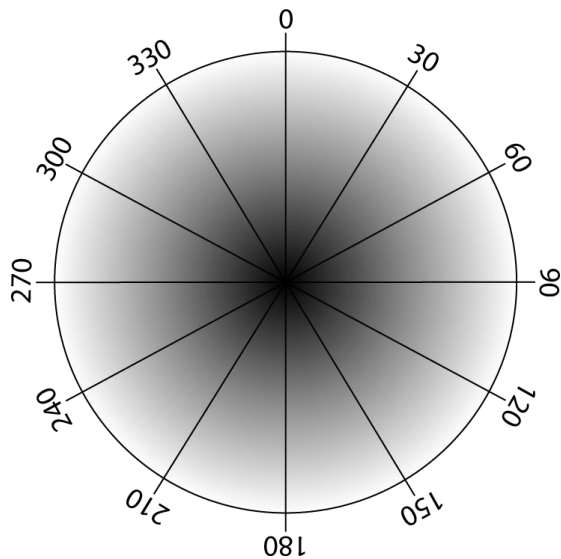
*Superior esquerdo
para inferior direito*

```
<style>
  body {
    background-image: linear-gradient(to
      bottom right, blue, yellow);
  }
</style>
```

▲ Ângulos



```
<style>
  body {
    background-image: linear-
      gradient(180deg, blue, yellow);
  }
</style>
```



0degree	Para cima
90degree	Para a direita
180degree	Para baixo
270degree	Para a esquerda

Podemos ainda especificar as posições das cores em gradiente (veja a seguir).



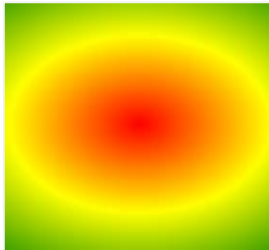
```
<style>
  body {
    background-image: linear-
      gradient(90deg, red 0%, yellow 50%,
        green 100%);
  }
</style>
```

Gradiente Radial

background-image: radial-gradient(tamanho da forma na posição, cor inicial, ..., cor final);

Um gradiente radial é definido pelo seu centro.

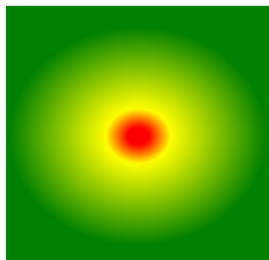
▲ Evenly Spaced



*Espaçamento
uniforme das cores
(padrão)*

```
<style>  
  body {  
    background-image: radial-gradient(red,  
    yellow, green);  
  }  
</style>
```

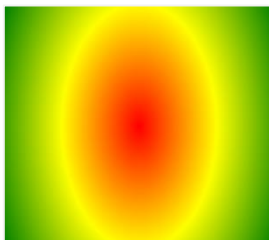
▲ Differently Spaced



*Espaçamento variável
das cores*

```
<style>  
  body {  
    background-image: radial-gradient(red  
    5%, yellow 15%, green 60%);  
  }  
</style>
```

▲ Set Shape



*Forma ajustada
(circle)*

```
<style>  
  body {  
    background-image: radial-  
    gradient(circle, red, yellow, green);  
  }  
</style>
```

Fontes em CSS

Demos início ao conteúdo de fontes, aplicando primeiramente a propriedade **font-family** nas configurações gerais do documento.

```
font-family: fonte principal, fonte secundária, fonte genérica;
```

Comece com a fonte desejada e termine com uma família genérica (para permitir que o navegador escolha uma fonte semelhante na família genérica, se nenhuma outra fonte estiver disponível).⁴

```
<style>
  body {
    font-family: 'Courier New', Courier, monospace;
  }
</style>
```

(Nomes com mais de uma palavra devem estar entre aspas)

Damos preferência a mais de uma fonte para que funcionem como um sistema de substituição, garantindo a compatibilidade entre o máximo de navegadores/ sistemas operacionais.

Famílias de Fontes Genéricas

Todas fontes diferentes pertencem a uma das famílias genéricas.⁵

Famílias	Descrição	Exemplos
Serif	As fontes têm um pequeno traço nas bordas de cada letra.	Georgia
Sans-serif	As fontes têm linhas limpas (sem pequenos traços anexados).	Arial
Monospace	As letras têm a mesma largura fixa, que criam uma aparência mecânica.	Lucida console
Cursive	As fontes imitam a caligrafia humana.	<i>Lucida Handwriting</i>
Fantasy	São fontes decorativas / divertidas.	Papyrus

⁴ [Safe Fonts](#)

⁵ [Generic Font Family](#)

Tamanho de Fontes

Os tamanhos são configurados em unidades de medida⁶, alterando o corpo da fonte. Nós usamos as medidas (px) e (em), recomendadas pela W3C, para CSS com a propriedade **font-size**.

```
<style>
  body {
    font-family: 'Courier New', Courier, monospace;
    /*
    16px geralmente = 1em
    */
    font-size: 1em;
  }
</style>
```



Nas CSS, ainda é possível fazer uso dos comentários, utilizando (/*) para abertura, e (*/) para o fechamento do comentário.

⁶ CSS: Unidades de Medida

A medida (em) será relativa ao tamanho da fonte, que geralmente está representada em 16 pixels, sendo 16px = 1em; 32px = 2em; etc. Ambas as unidades (px) e (em) são categorizadas, respectivamente, como absoluta e relativa.

As unidades de **medida absoluta** são fixas e não variam, expressando-se da mesma forma para qualquer tipo de uso. Porém, não são recomendadas para uso em telas, porque tamanhos de tela variam muito.

As unidades de **medida relativa** são flexíveis e variam de acordo a fonte.

Medidas Absolutas	Medidas Relativas
pixel (px)*	em
centímetro (cm)	ex
milímetro (mm)	rem
polegada (in)	vw
ponto (pt)	vh
paica (pc)	%
*O pixel pode ser relativo, dependendo do PPI ⁷ (pixel por polegada) da tela do dispositivo.	

As unidades de medida estão relacionadas com a mídia especificada (tela ou papel), portanto é necessário saber como elas se adequam.

	Recomendado	Uso ocasional	Não recomendado
Tela	em, px, %	ex	pt, cm, mm, in, pc
Impresso	em, cm, mm, in, pt, pc, %	px, ex	-

⁷ Saiba sobre PPI

Peso, Estilo e Shorthand Font

A propriedade **font-weight** especifica o peso da fonte.

- Normal – Fonte padrão
- Bold – Fonte cheia
- Bolder – Fonte mais cheia
- Lighter – Fonte mais clara

A propriedade **font-style** especifica o estilo da fonte.

- Normal – Fonte padrão
- Italic – Fonte em itálico
- Oblique – Fonte oblíqua

```
<style>
  body {
    font-style: normal;
    font-weight: normal;
  }
</style>
```

É chamado de shorthand font a propriedade que especifica todas as propriedades individuais da fonte simultaneamente.⁸

```
<style>
  body {
    font: italic bolder 3em 'Work Sans', sans-serif;
  }
</style>
```

As propriedades são dispostas nessa ordem: **font-style**, **font-weight**, **font-size** e **font-family**.



Na verdade, existem outros tipos de propriedade shorthand, que permitem definir os valores de várias outras propriedades CSS.

⁸ CSS Font Property

Fontes Externas

A partir daqui demos um novo passo, pois começamos a usar as regras das CSS.

Importadas

São geradas a partir da regra `@import` das CSS, que é fornecida pelo próprio código da fonte externa. Nesse caso, utilizamos o Google Fonts que disponibiliza esses códigos.

400 itálico regular

Considerando que o reconhece

— Seleccione este estilo

Use na web

Para incorporar uma fonte, copie o código no <head> de seu html

☐ <link> ☒ @importar

```
<style>
@import url('https://fonts.googleapis.com/css2?family=Noto+Serif:ital@1&display=swap');
</style>
```

```
<style>
  body {
    @import url('https://fonts.googleapis.com/css2?family=Noto+Serif:ital@1&display=swap');

    font-family: 'Noto Serif', serif;
  }
</style>
```

Baixadas

São geradas a partir da regra `@font-face` das CSS. Primeiramente, é necessário adicionar o nome da fonte.⁹

```
@font-face {
  font-family: 'Bubble Shine';
}
```

⁹ CSS `@font-face` Rule

E em seguida, adicionaremos a propriedade **src** para definir:

1. **url**, que é o local onde a fonte está, e
2. **format**, que é o formato da fonte.¹⁰

```
@font-face {  
    font-family: 'Bubble Shine';  
    src: url('fontes/BubbleShine.otf') format('opentype'),  
        url('fontes/BubbleShine.ttf') format('truetype');
```

Feito isso, temos que adicionar um seletor para aplicar o estilo.

```
h1 {  
    font-family: 'Bubble Shine' Times, serif;  
    font-size: 4em;  
    font-weight: normal;  
}
```

Pronto, está feito!



No exemplo, estou usando uma fonte baixada do DaFont, serviço online para download de fontes, em formatos (otf.) e (ttf.).

¹⁰ [Different Font Formats](#)

IDENTIFICANDO FONTES

Existem alguns sites, onde você pode detectar fontes dentro de imagens.

No site, vá até a página Principal.



No site, clique em “Font Identifier”.



No site, clique em “WhatTheFont”.



Alinhamento de Textos

A propriedade **text-align** especifica o alinhamento horizontal do texto em um elemento.

- Center – Centraliza o texto
- Left – Alinha o texto à esquerda
- Right – Alinha o texto à direita
- Justify – Espaço adicionado entre as palavras para que ambas as bordas de cada linha sejam alinhadas com as margens.

```
<style>
  body {
    text-align: justify;
  }
</style>
```

ENVELOPAMENTO CASCATA

Você pode envelopar com mais de uma tag ao mesmo tempo, como neste exemplo das listas que eu configurei “ul>li*” para envelopar.

```
ul>li*  
  
<body>  
<ul>  
  <li>item</li>  
  <li>item</li>  
  <li>item</li>  
</ul>  
</body>
```

O asterisco (*) aplica uma para cada item.

Uma variação para esse exemplo seria uma para cada .

```
ul*>li  
  
<ul>  
  <li>item</li>  
</ul>  
<ul>  
  <li>item</li>  
</ul>  
<ul>  
  <li>item</li>  
</ul>
```

Seletores Personalizados

Os seletores CSS são usados para “encontrar” (ou selecionar) os elementos HTML que você deseja estilizar.¹¹

¹¹ [CSS Selectors](#)

Identity

O atributo **id** especifica uma identidade única para um elemento HTML.

```
<body>
  <h1 id="principal">
    Criando sites com HTML e CSS
  </h1>
</body >
```

Em CSS usa-se uma hashtag (#) seguida pelo nome/identidade.

```
#principal {
    text-align: center;
    background-color: rgb(4, 109, 0);
    color: white;
}
```

Class

O atributo **class** especifica uma classe para um elemento HTML.

```
<body>
  <h2 class="basico">
    HTML Básico
  </h2>
</body >
```

Em CSS usa-se um ponto (.) seguido pelo nome da classe.

```
.basico {
    color: #3c805e;
}
```



Em um mesmo documento HTML, o **class** pode ser usado por vários elementos diferentes. Porém, só podemos usar um **id** para um único elemento.

Pseudo-classes

Uma pseudo-classe CSS é usada para definir um estado especial de um elemento.¹²

▲ :visited

O **:visited** define um estilo aos links, que já foram visitados.

```
a:visited {  
    color: #8b0000;  
}
```

▲ :hover

O **:hover** é usado para acionar um estilo quando o mouse passa sobre o elemento.

```
a:hover {  
    color: #ff0000;  
}
```

▲ :active

O **:active** é usado para aplicar um estilo aos links ativos. Um link torna-se ativo quando você clica nele. Pode ser usado em todos os elementos, não apenas em links.

```
a:active {  
    color: #ffff00;  
}
```

¹² Pseudo-classes CSS

Pseudo-elementos

Um pseudo-elemento CSS é usado para estilizar partes específicas de um elemento.¹³

▲ ::before

O **::before** insere algo antes do conteúdo de cada elemento selecionado.

```
h1::before {  
    content: '>';  
}
```

▲ ::after

O **::after** insere algo depois do conteúdo de cada elemento selecionado.

```
h1::after {  
    content: '<';  
}
```

▲ ::first-letter

O **::first-letter** é usado para adicionar um estilo à primeira letra do seletor especificado.

```
p.intro::first-letter {  
    color: #ffff00;  
}
```

▲ ::first-line

O seletor **::first-line** é usado para adicionar um estilo à primeira linha do seletor especificado.

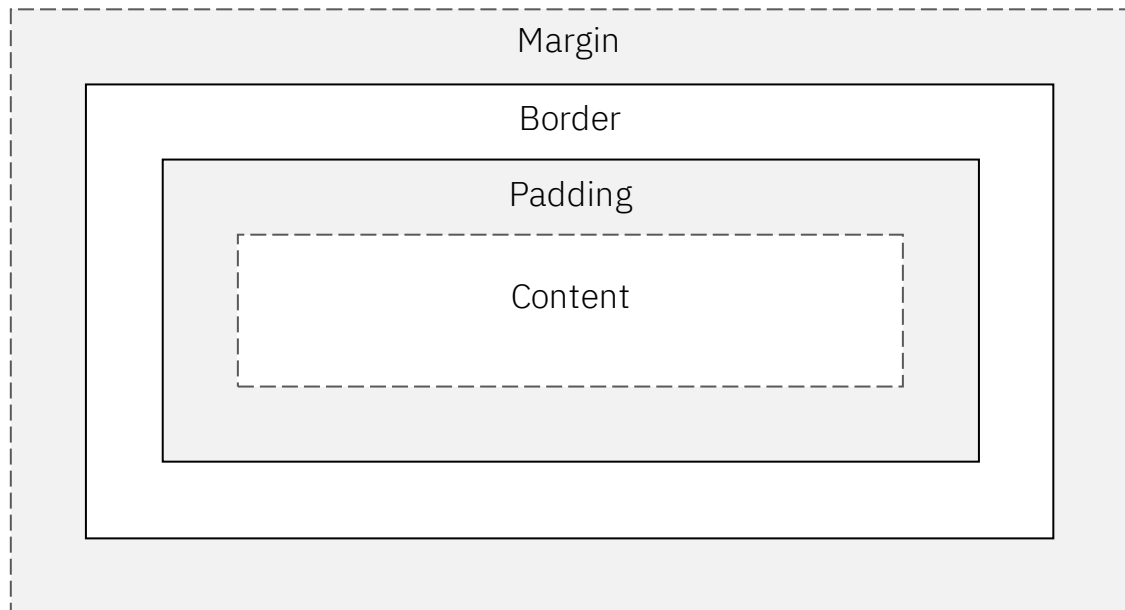
```
p::first-line {  
    color: #9b9a9a;  
}
```

¹³ Pseudo-elements CSS

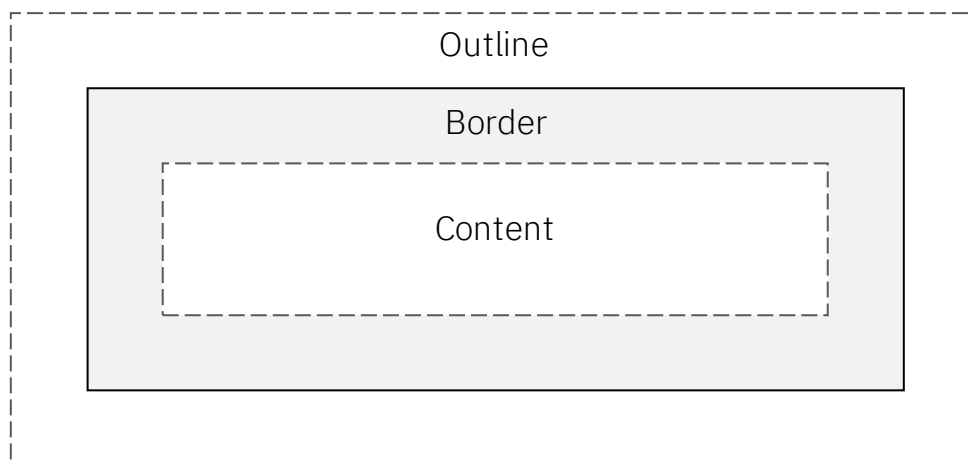
Modelo de Caixas

Todos os elementos HTML estão envolvidos por uma caixa, que delimita o conteúdo e possui altura e largura.

O modelo de caixa consiste em: margens, bordas, preenchimento e o conteúdo. Neste exemplo, ocorre o aninhamento de caixas, estando elas uma dentro da outra.



Além disso, entre a margem e a borda, existe o contorno. É a linha fora da borda que destaca o elemento.



Altura e largura total

- Altura

= height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

- Largura

= width + left padding + right padding + left border + right border + left margin + right margin

Propriedades padding e margin

São propriedades shorthand para padding/margin:

- | | | |
|----------|------|--------|
| • Top | | Top |
| • Right | | |
| • Bottom | Left | Right |
| • Left | | Bottom |

Por exemplo, para definir esses valores adicionaremos **padding-top** ou **margin-top**, etc.

```
h1 {  
    height: 300px;  
    width: 300px;  
  
    padding-top: 10px;  
    padding-right: 10px;  
    padding-bottom: 10px;  
    padding-left: 10px;  
  
    margin-top: 20px;  
    margin-right: 40px;  
    margin-bottom: 20 px;  
    margin-left: 40px;  
}
```


De maneira abreviada, usa-se apenas **padding** e **margin**.

```
h1 {  
    padding: 10px;  
    margin: 20px 40px;  
}
```

Propriedades border e outline

São propriedades shorthand para border/outline:

- Width
- Style¹⁴
- Color

Por exemplo, para definir esses valores adicionaremos **border-width** ou **outline-width**, etc.

```
h1 {  
    height: 300px;  
    width: 300 px;  
  
    border-width: 10px;  
    border-style: solid;  
    border-color: darkslategray;  
  
    outline-width: 5px;  
    outline -style: dashed;  
    outline-color: salmon;  
}
```

De maneira abreviada, usa-se apenas **border** e **outline**.

```
h1 {  
    border: 10px solid darkslategray;  
    outline: 5px dashed salmon;  
}
```

¹⁴ [Border-style](#) e [Outline-style](#)

MARGIN:AUTO

Usando o `margin: auto` o navegador calcula automaticamente as margens e centraliza o elemento.

```
margin: 20px auto 40px auto;
```

Tipos de Caixa

Cada elemento HTML tem um valor de exibição padrão, dependendo de que tipo de elemento ele é.

Existem dois valores de exibição: block e inline.¹⁵

Elementos Block-level

Esse tipo de elemento sempre se inicia em uma nova linha, ocupando a largura total disponível e possui uma margem superior e inferior.

O elemento `<div>`, por exemplo é do tipo block-level.

Veja outros elementos HTML como esse a seguir:

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

¹⁵ [HTML Block and Inline Elements](#)

Elementos Inline

Esse tipo de elemento não se inicia em uma nova linha, somente ocupa a largura necessária.

O elemento ``, por exemplo está dentro do parágrafo.

Veja outros elementos HTML como esse a seguir:

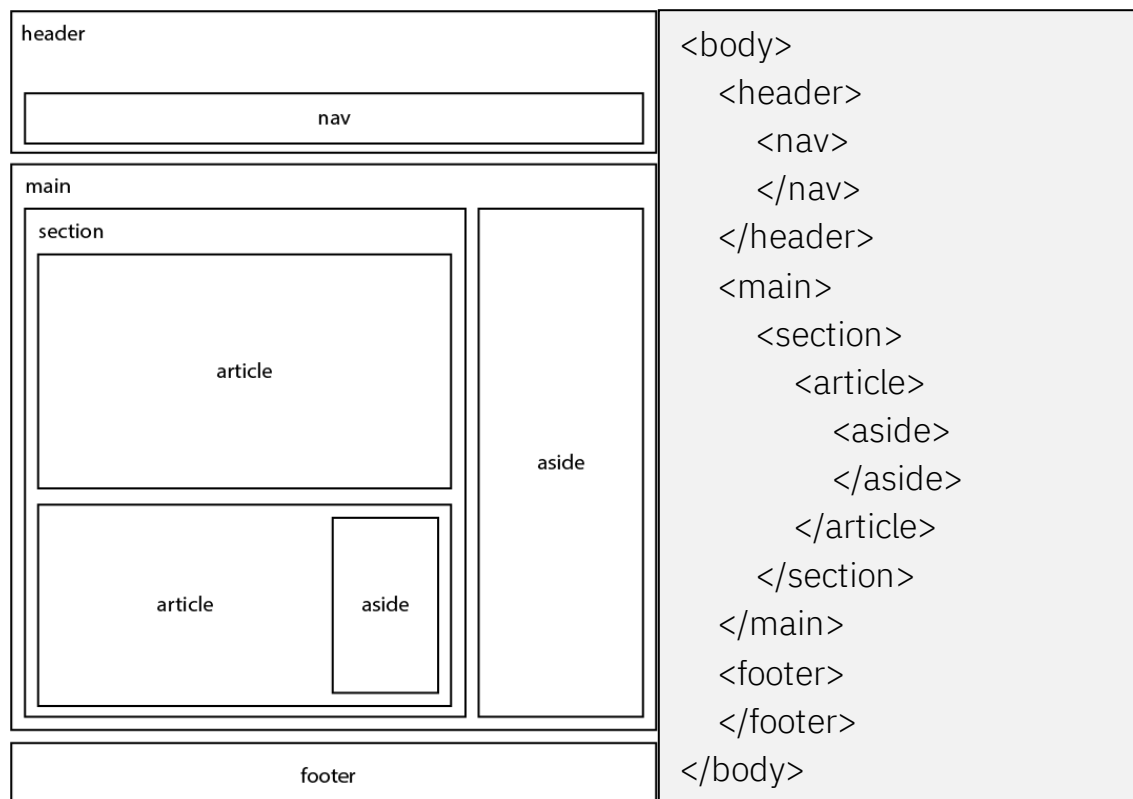
<code><a></code>	<code><abbr></code>	<code><acronym></code>	<code></code>
<code><button></code>	<code><cite></code>	<code><code></code>	<code><dfn></code>
<code><input></code>	<code><kbd></code>	<code><label></code>	<code><map></code>
<code><samp></code>	<code><script></code>	<code><select></code>	<code>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

Grouping Tags

As grouping tags agrupam o conteúdo do site, tornando-o mais organizado.

<header>	<nav>
É o cabeçalho que contém o conteúdo introdutório da seção ou artigo relacionado.	Define uma área com links de navegação
<main>	<section>
Especifica o conteúdo principal do site.	Define uma seção em um documento.
<article>	<aside>
Define os conteúdos que devem ser lidos de forma independente.	Define o conteúdo, indiretamente, relacionado ao conteúdo principal.
<footer>	
Define um rodapé para um documento ou seção.	

A seguir, veja uma representação ilustrativa dessa estrutura:



Sombras nas Caixas

As sombras são geradas a partir da propriedade **box-shadow**.¹⁶

Uma **box-shadow** é descrita pelos deslocamentos (offset) X e Y em relação ao elemento, desfoco e propagação do raio e cor.

- x-offset – O deslocamento horizontal da sombra
- y-offset – O deslocamento vertical da sombra
- blur – O raio de desfoco.
- spread – O raio de propagação
- color – A cor da sombra

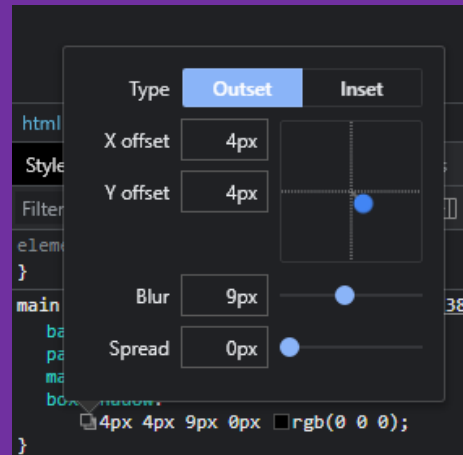
```
main {  
    background-color: white;  
    padding: 10px;  
    margin: 10px;  
    box-shadow: 4px 4px 9px 0px rgba(0, 0, 0, 384);  
}
```

¹⁶ [Box-shadow](#)

EDITOR DE SOMBRAS

Você pode usar o editor de sombras do DevTools para ajudá-lo na personalização das sombras.

Após ter feito a edição, você deve copiar o código e colar no VSCode, alterando os valores da `box-shadow`.



Caixas com Vértices

A propriedade `border-radius` permite adicionar cantos arredondados aos elementos. Essa propriedade pode ter de um a quatro valores (veja a seguir).¹⁷

`border-top-left-radius`

`border-top-right-radius`



`border-bottom-left-radius`

`border-bottom-right-radius`

¹⁷ [CSS border-radius Property](#)

▲ Quatro valores



```
main {  
    height: 100px;  
    width: 100px;  
    background-color: darkviolet;  
    border-radius: 15px 50px 30px 5px;  
}
```

▲ Três valores



```
main {  
    height: 100px;  
    width: 100px;  
    background-color: darkviolet;  
    border-radius: 15px 50px 30px;  
}
```

▲ Dois valores



```
main {  
    height: 100px;  
    width: 100px;  
    background-color: darkviolet;  
    border-radius: 15px 50px;  
}
```

▲ Um valor



```
main {  
    height: 100px;  
    width: 100px;  
    background-color: darkviolet;  
    border-radius: 15px;  
}
```

Variáveis em CSS

São propriedades personalizadas, que contêm valores específicos a serem reutilizados em um documento, variando de acordo com a necessidade. São configuradas por notação e acessadas pela função **var()**.¹⁸

```
--nome: valor; | var(--nome);
```

Escopo Global e Local

Variáveis globais podem ser usadas para o documento inteiro, enquanto que as variáveis locais só podem ser usadas dentro do seletor onde estão declaradas.

▲ Global

```
:root {  
    --cor0: #c5ebd6;  
    --font-padrao: Arial, Verdana, Helvetica sans-serif;  
}  
  
body {  
    background-color: var(--cor0);  
    font-family: var(--font-padrao);  
}
```

▲ Local

```
footer {  
    --cor-fundo: black;  
    --largura: 40%;  
}
```

¹⁸ [Utilizando Variáveis CSS](#)

SELETOR * CSS

Usando o seletor * (asterisco) você pode selecionar todos os elementos.

```
* {  
    declaração css;  
}
```

Por exemplo, definindo `margin: 0px;` e `padding: 0px;` todos os elementos do documento adotariam essas configurações.

Responsividade para Sites

Sites responsivos são aqueles adaptáveis ao tamanho das telas que estão sendo exibidos para facilitar a visualização do conteúdo.

Caixas Flexíveis

A largura e altura são adaptáveis ao elemento, usando valores mínimos e máximos.

▲ Altura mínima e máxima

```
main {  
    min-height: 400px;  
    max-height: 800px;  
}
```

▲ Largura mínima e máxima

```
main {  
    min-width: 400px;  
    max-width: 800px;  
}
```

<picture>

A tag **<picture>** oferece aos desenvolvedores da web mais flexibilidade na especificação de recursos de imagem. Seu uso mais comum será para criação de web designs responsivos, sendo assim, várias imagens podem ser projetadas para preencher melhor a janela de visualização do navegador.

O elemento **<picture>** contém duas tags: uma ou mais tags **<source>** e uma tag ****.

```
<body>
  <picture>
    <source media="max-width: 600px" srcset="imagens/irina-blok-
    pq.jpg">
    
  </picture>
</body>
```

Configurando Textos e Linhas

Text-indent

A propriedade **text-indent** especifica o recuo da primeira linha em um bloco de texto.

```
main p {  
    text-align: justify;  
    text-indent: 30px;  
}
```

Line-height

A propriedade **line-height** especifica a altura dos espaços entrelinhas.

```
main p {  
    text-align: justify;  
    text-indent: 30px;  
    line-height: 2em;  
}
```

Text-shadow

A propriedade **text-shadow** adiciona sombra ao texto.

```
header > h1 {  
    color: white;  
    font-family: var(--font-destaque);  
    text-shadow: 2px 2px 0px rgba(0, 0, 0, 0.267);  
}
```

Personalizando as Listas

List-style-position

A propriedade **list-style-position** especifica a posição dos marcadores de item da lista.

▲ Outside

Significa que os marcadores estarão fora do item da lista.

- | | |
|---|------|
| • | Item |
| • | Item |

▲ Inside

Significa que os marcadores estarão dentro do item da lista.

- | | |
|---|------|
| • | Item |
| • | Item |

List-style-type

A propriedade **list-style-type** especifica o tipo de marcador de item de lista em uma lista. ¹⁹

Columns

A propriedade **columns** é uma shorthand para:

- column-width – largura da coluna
- column-count – contagem de colunas

```
aside > ul {  
    list-style-type: '\2714\0020\0020';  
    list-style-position: inside;  
    columns: 2;  
}
```

¹⁹ [CSS list-style-type Property](#)

Vídeo Responsivo

Para tornar um vídeo responsivo foi necessário saber do macete do Guanabara.

Primeiramente, envelopamos o `<iframe>` com uma `<div>` para torna-lo um bloco. Após isso, foi determinado uma `class` para a `<div>`.

```
<body>
  <main>
    <article>
      <div class="video"><iframe "incorporação"></iframe></div>
    </article>
  </main>
</body>
```

Feito isso, vamos para as configurações de estilo.

CSS Position

A propriedade `position` especifica o tipo de método de posicionamento usado para um elemento (`static`, `relative`, `fixed`, `absolute` ou `sticky`).²⁰

▲ Static (Padrão)

Um elemento com a posição estática não está posicionado de forma especial está sempre posicionado de acordo com o fluxo normal da página.

Uma `<div>` com `position: static;`

```
div.static {
  position: static;
  border: 3px solid;
}
```

²⁰ [Position](#)

▲ Relative

Um elemento com a posição relativa está posicionado relativo à sua posição normal. Essa posição pode ser determinada pelas propriedades: **top**, **right**, **bottom** e **left**.

Uma <div> com position: relative;

```
div.relative {  
    position: relative;  
    left: 30px;  
    border: 3px solid;  
}
```

▲ Absolute

O elemento é removido do fluxo normal de documentos e nenhum espaço é criado para o elemento no layout da página. Ele é posicionado em relação ao seu ancestral posicionado mais próximo, se houver; caso contrário, ele é colocado em relação ao bloco inicial.

```
div.relative {  
    position: relative;  
    width: 400px;  
    height: 200px;  
    border: 3px solid;  
}
```

```
div.absolute {  
    position: absolute;  
    top: 80px;  
    right: 0px;  
    width: 200px;  
    height: 100px;  
    border: 3px solid;  
}
```

Uma <div> com position: relative;

Uma <div> com position:
absolute;

Configurações de Estilo

Nas configurações de estilo adotamos estes valores para torna o vídeo responsivo:

```
div.video {  
    position: relative;  
    background-color: var(--cor4);  
    height: 0px;  
    margin: 0px -20px 15px -20px;  
    padding: 20px;  
    padding-bottom: 58px;  
}  
  
div.video > iframe {  
    position: absolute;  
    top: 5%;  
    left: 5%;  
    width: 90%;  
    height: 90%;  
}
```