

Dados do Aluno

Nome: _____

Número da matrícula: _____

Endereço: _____

Bairro: _____

Cidade: _____

Telefone: _____

Anotações Gerais: _____



HTML & CSS

O HTML é responsável por montar o "esqueleto", ou seja, a estrutura de uma página da web. Já o CSS é responsável por dar "uma cara" ao esqueleto da página, permitindo adicionar cores, fontes, efeitos e vida ao site. Este curso procura passar a base do HTML e do CSS, permitindo que o aluno inicie seu caminho no mundo do front-end.

Marcas Registradas:

Todas as marcas e nomes de produtos apresentados nesta apostila são de responsabilidade de seus respectivos proprietários, não estando a editora associada a nenhum fornecedor ou produto apresentado nesta apostila.



Método CGD® - Todos os direitos reservados.

Protegidos pela Lei 5988 de 14/12/1973.

Nenhuma parte desta apostila poderá ser copiada sem prévia autorização.

O Método CGD é um produto da Editora CGD.

[illegible]

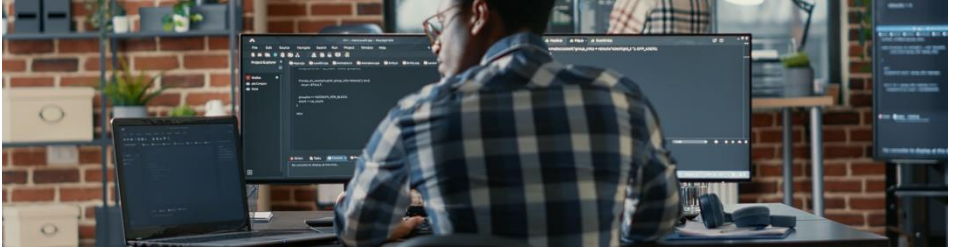
01 – PREPARAÇÃO DO AMBIENTE E ESTRUTURA BÁSICA DO HTML	7
• INTRODUÇÃO	7
• UM RESUMO DA HISTÓRIA DO HTML	7
• O CONCEITO BÁSICO	8
• ESTRUTURA BÁSICA DO HTML.....	9
• A ESTRUTURA DE UMA PÁGINA HTML	10
02 - COMPREENDENDO OS ELEMENTOS E SEUS ATRIBUTOS.....	12
• ELEMENTOS E ATRIBUTOS	12
• ELEMENTOS ANINHADOS	13
• ELEMENTOS EM BLOCO E ELEMENTOS EM LINHA (INLINE)	14
• ELEMENTOS VAZIOS.....	15
03 - TÍTULOS, PARÁGRAFOS E FORMATAÇÃO.....	16
• TÍTULOS, PARÁGRAFOS E FORMATAÇÃO	16
• TÍTULOS	16
• PARÁGRAFOS	17
• FORMATAÇÃO	19
04 - USANDO IMAGENS	20
• A IMAGEM NO HTML	20
• A SINTAXE DO ELEMENTO	20
• A IMPORTÂNCIA DO ATRIBUTO ALT	22
• OS FORMATOS MAIS COMUNS	22
• O TAMANHO DA IMAGEM	23
05 - CRIANDO LINKS (HYPERLINKS)	24
• A SINTAXE DO ELEMENTO	24
• O ATRIBUTO HREF	25
• URL ABSOLUTA E URL RELATIVA.....	25
• OS ATRIBUTOS TARGET E TITLE	26
06 - CRIANDO TABELAS HTML.....	27
• LINHAS E CÉLULAS.....	29
• CABEÇALHOS.....	30
• BORDAS	31
07 - CRIANDO LISTAS HTML	32
• AS LISTAS NÃO ORDENADAS	32
• LISTAS ORDENADAS	33
08 - CRIANDO FORMULÁRIOS	34
• O ELEMENTO FORM.....	34
• O ELEMENTO INPUT.....	35
• O ELEMENTO LABEL	36
• O ELEMENTO TEXTAREA	36

●	O ELEMENTO SELECT	37
●	O INPUT TYPE "SUBMIT"	38
●	O ATRIBUTO NAME	38
●	OS ATRIBUTOS METHOD E ACTION	40
09	- INTRODUÇÃO AO CSS: SINTAXES E SELETORES	41
●	O QUE É CSS?	41
●	A SINTAXE DO CSS	41
●	OS SELETORES CSS	42
●	O SELETOR DE ELEMENTO HTML	42
●	O SELETOR DE ID	42
●	O SELETOR DE CLASSE	43
●	O SELETOR UNIVERSAL	43
●	COMO UTILIZAR O CSS	43
●	O CSS EXTERNO	43
●	O CSS INTERNO	44
●	O CSS EM LINHA	45
10	- O MODELO DE CAIXA DO CSS	45
●	DEFININDO A LARGURA E A ALTURA DE UM ELEMENTO	47
●	A PROPRIEDADE BOX-SIZING	48
11	- CSS RESPONSIVO: RWD	51
●	O VIEWPORT	51
●	O CONCEITO DO GRID-VIEW	53
●	CONSTRUINDO O GRID-VIEW RESPONSIVO	54
●	A PROPRIEDADE FLOAT	55
12	- AJUSTANDO O FUNDO DOS ELEMENTOS	56
●	AS PROPRIEDADES BACKGROUND	56
●	A PROPRIEDADE BACKGROUND-COLOR	56
●	A PROPRIEDADE BACKGROUND-IMAGE	58
●	A PROPRIEDADE BACKGROUND-REPEAT	59
●	A PROPRIEDADE BACKGROUND-POSITION	60
●	A PROPRIEDADE BACKGROUND-ATTACHMENT	61
13	- AJUSTANDO O TEXTO DO SITE	62
●	A PROPRIEDADE COLOR	62
●	A PROPRIEDADE TEXT-ALIGN	65
●	A PROPRIEDADE TEXT-DECORATION	65
●	A PROPRIEDADE TEXT-TRANSFORM	69
●	A PROPRIEDADE FONT-FAMILY	69
●	A PROPRIEDADE FONT-STYLE	71
●	A PROPRIEDADE FONT-WEIGHT	71
●	A PROPRIEDADE FONT-SIZE	71
●	GOOGLE FONTS	73

14 - UTILIZANDO ÍCONES NO SITE	76
● O FONT AWESOME	76
● A CRIAÇÃO DA CONTA E DO KIT	77
● INSERINDO OS ÍCONES EM SUA PÁGINA	77
● ALTERANDO O TAMANHO DOS ÍCONES.....	79
15 - PERSONALIZANDO OS LINKS COM O USO DO CSS.....	80
16 - AJUSTANDO LISTAS E TABELAS COM O USO DO CSS	83
● A PROPRIEDADE LIST-STYLE-TYPE.....	83
● A PROPRIEDADE LIST-STYLE-IMAGE	87
● A PROPRIEDADE BACKGROUND	88
● O USO DE LISTAS NA CRIAÇÃO DE MENUS	89
● AJUSTANDO TABELAS COM O USO DO CSS	93
● EXEMPLO 01 :: TABELA COM BORDA SIMPLES.....	95
● EXEMPLO 02 :: TABELA COM BORDAS DIFERENTES.....	96
● EXEMPLO 03 :: TABELA COM BORDA INFERIOR	97
● EXEMPLO 04 :: TABELA COM EFEITO ZEBRADO.....	98
● EXEMPLO 05 :: TABELA ESTILOSA	100
17 - AJUSTANDO FORMULÁRIOS COM O USO DO CSS	101
● ALTERANDO OS INPUTS	102
● AJUSTANDO OS LABELS.....	105
● AJUSTANDO O BOTÃO DO FORMULÁRIO.....	108
18 - AJUSTANDO O LAYOUT COM O CSS.....	113
19 - FINALIZANDO O PROJETO	115

01 – Preparação do ambiente e estrutura básica do HTML

• Introdução



- Neste curso você entrará em contato com as principais características do HTML e do CSS necessárias para a construção de um site.
- Esta primeira lição servirá como uma introdução, então vamos começar pelo início...

• Um resumo da história do HTML

- HTML é a abreviação da expressão inglesa "HyperText Markup Language" que significa "Linguagem de Marcação de Hipertexto", ele é responsável pelo significado e pela estrutura de um conteúdo web.
- Tim Berners-Lee, um físico britânico, criou o HTML original para resolver um problema que ele tinha: a comunicação e disseminação das pesquisas entre ele e o seu grupo de colegas. A sua solução, combinada com a então emergente internet pública, ganhou atenção mundial.
- As primeiras versões do HTML foram definidas com regras sintáticas flexíveis, o que ajudou aqueles sem familiaridade com a publicação na Web. Com o passar do tempo, a utilização de ferramentas para autoria de HTML aumentou, assim como a tendência em tornar a sintaxe cada vez mais rígida. Apesar disso, por questões históricas (retro compatibilidade), os navegadores ainda hoje conseguem interpretar páginas web que estão longe de ter um código HTML válido.
- De 1990 para cá o HTML passou por várias revisões e atualizações, em 1996 suas especificações passaram a ser mantidas, com o

auxílio de algumas empresas de software, pelo World Wide Web Consortium (W3C).

- Várias versões de HTML foram publicadas, confira a tabela abaixo:

VERSÃO	ANO
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014
HTML5.1	2016 e 2017 (2ª ed.)
HTML5.2	2017

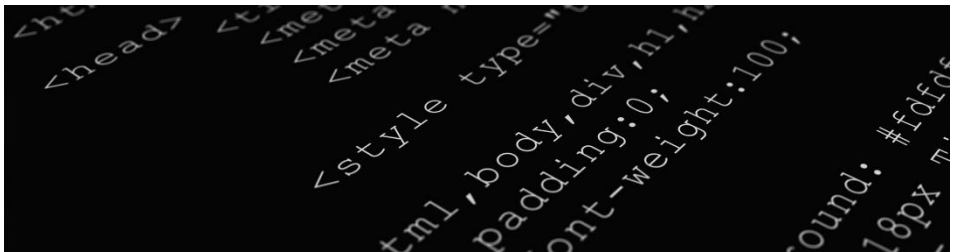
• O conceito básico

- É importante fixar que o HTML não é uma linguagem de programação, ele é uma linguagem de marcação!
- O HTML usa "marcação" para definir texto, imagem e outros conteúdos para exibição em um navegador, essa marcação consiste de uma série de elementos, que você usa para delimitar ou agrupar diferentes partes do conteúdo para que ele apareça ou atue de determinada maneira.
- Confira alguns exemplos de "elementos":
 - <head>
 - <title>
 - <body>
 - <footer>
 - <p>
 - <div>
 -
 -
- Um elemento HTML é separado de outro texto do documento por "tags", que consistem no nome do elemento entre "<" e ">". O nome de um elemento dentro de uma tag é insensível a maiúsculas

e minúsculas. Isto é, pode ser escrito em maiúsculas, minúsculas ou uma mistura. Por exemplo, a tag <title> pode ser escrita como <Title>, <TITLE> ou de qualquer outra forma. Porém, o mais usual é manter a escrita em letras minúsculas, é assim que seguiremos neste curso.

- As tags anexas podem transformar uma palavra ou imagem num hiperlink, pode colocar palavras em itálico, pode aumentar ou diminuir a fonte e assim por diante.

• Estrutura Básica do HTML



- Um elemento HTML é composto três partes, são elas:
 - Tag de abertura
 - Tag de fechamento
 - Conteúdo
- A **tag de abertura** é composta do nome do elemento, envolto dos sinais de menor (<) e maior (>) - Essa composição indica onde o elemento HTML começa, por exemplo: **<p>**
- A **tag de fechamento** é basicamente igual à tag de abertura, porém, com a inclusão de uma barra diagonal (/) após o sinal de menor (<) - Essa composição indica o final do elemento HTML, por exemplo: **</p>**
- O **conteúdo** é responsável por definir o que o objeto HTML irá exibir, no caso dos exemplos acima, as tags definem um parágrafo...
A composição completa seria a seguinte: **<p>Eu sou um parágrafo</p>**

- Confira o elemento do exemplo acima com suas partes identificadas:



- Esquecer o fechamento de uma tag é um dos erros mais comuns de quem está iniciando no uso do HTML, esse erro pode levar a resultados inesperados, então é importante que você sempre preste atenção nas tags que está criando.
- Uma das vantagens de utilizar o Visual Studio Code na criação de páginas HTML é que, ao inserir uma tag de abertura, o próprio programa já cria a sua tag de fechamento. Você verá isso a seguir.
- Sempre confira as tags de abertura e de fechamento de seus elementos HTML

• A estrutura de uma página HTML

- A estrutura básica de uma página HTML é composta de alguns elementos, são eles:
 - `<!DOCTYPE html>`
 - `<html></html>`
 - `<head></head>`
 - `<meta charset="utf-8">`
 - `<title></title>`
 - `<body></body>`
- O elemento **`<!DOCTYPE html>`** define que o documento é uma página HTML5.
- O elemento **`<html>`** envolve todo o conteúdo da página, ele também é conhecido como **"elemento raiz"**.
- O elemento **`<head>`** serve como um recipiente para informações que você deseja incluir em sua página e que não são conteúdo para o usuário final. É neste local que incluímos palavras-chave,

descrições para o resultado de buscas, folhas de estilo (CSS) e outras declarações.

- O elemento **<meta charset="utf-8">** define que o conjunto de caracteres utilizado em sua página é o "UTF-8", que inclui praticamente todos os caracteres dos idiomas escritos.
- O elemento **<title>** define o título da sua página. Este título é exibido na guia do navegador quando a sua página é visitada.
- O elemento **<body>** engloba todo o conteúdo que você deseja exibir ao usuário. É aqui onde você irá inserir seus textos, suas imagens, seus links e muito mais...
- A construção de uma página, utilizando os elementos acima seria feita da seguinte maneira:

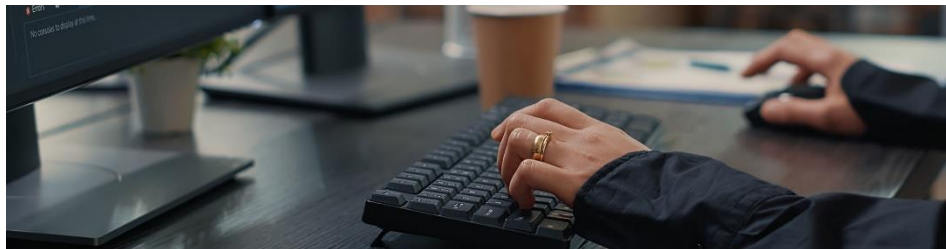
```
<html>
  <head>
    <meta charset="utf-8">
    <title>Minha página de teste</title>
  </head>
  <body>

    Entre as tags body, você irá inserir seu conteúdo...

  </body>
</html>
```

02 - Compreendendo os elementos e seus atributos

• Elementos e atributos



- Conforme visto na aula passada, um elemento HTML é composto três partes. Confira o elemento a seguir: **<p>Eu sou um parágrafo</p>** - As partes do elemento são:
 - Tag de abertura: **<p>**
 - Tag de fechamento: **</p>**
 - Conteúdo: **Eu sou um parágrafo**



- Além dessas três partes, todos os elementos HTML podem possuir **atributos**. Um atributo serve para passar informações adicionais sobre o elemento em que for adicionado.
- Alguns elementos HTML **necessitam** de um atributo para funcionar corretamente, como por exemplo o elemento de **hyperlink**, ele precisa do atributo **href** para indicar o seu destino (você verá mais sobre ele em breve).
- Vamos pegar como exemplo um elemento visto na aula anterior, o **meta**...

```
<meta charset="utf-8">
```

- Neste caso, temos o elemento **meta** com o atributo **charset**, acompanhado do valor **utf-8**
- Um atributo sempre deve ter:
 - Um espaço entre ele e o nome do elemento (ou o atributo anterior, se o elemento já tiver um).
 - O nome do atributo, seguido por um sinal de igual.
 - Aspas de abertura e fechamento, envolvendo todo o valor do atributo.
- Durante a criação de sua página, você poderá utilizar aspas simples (') ou duplas (") em seu código, esta é uma questão puramente estética, ambas são equivalentes.
- É importante se certificar de não misturar os dois tipos de aspas juntos. Se a abertura foi feita com ", o fechamento deverá ser feito com " também.
- Por mais que o padrão do HTML não requeira que os atributos sejam escritos em letras minúsculas, é recomendado que, assim como para o nome do elemento, você escreva-o em letras minúsculas.

• Elementos aninhados

- No HTML elementos podem ser inseridos dentro de outros elementos, isso é conhecido como **aninhamento**, utilizando o exemplo visto na aula passada

```
<head>  
  <meta charset="utf-8">  
  <title>Minha primeira página HTML</title>  
</head>
```

- Os elementos **meta** e **title** estão aninhados, ou seja, dentro do elemento **head**. Esse aninhamento pode ocorrer com diversos elementos HTML e não existe um limite para tal.

- É importante apenas prestar atenção em uma coisa, checar se os objetos estão corretamente aninhados, por exemplo, ao criar um parágrafo (<p>) e inserir dentro dele um efeito de negrito (), é importante abrir e fechar as tags em ordem. Confira:

```
<p>Este parágrafo <strong>possui</strong> uma palavra em destaque</p>
```

- A vantagem de utilizar o VS Code para a criação do nosso código é que, como visto anteriormente, ao abrir um elemento, o programa automaticamente produz a tag de fechamento dele.
- Os elementos precisam ser abertos e fechados corretamente para que eles estejam claramente visíveis dentro ou fora um do outro.

• Elementos em bloco e elementos em linha (inline)

- Existem dois tipos de elementos que você precisa conhecer no HTML, os **elementos em bloco** e os **elementos em linha**.
- Os **elementos em bloco** formam um bloco visível na página, eles sempre aparecerão em uma nova linha após o elemento anterior e qualquer elemento posterior também será exibido em uma nova linha.
- Confira alguns exemplos de elementos em bloco:
 - Parágrafos <p>
 - Listas e
 - Tabelas <table>
- Um elemento em bloco não deve ser aninhado dentro de um elemento em linha, mas pode ser aninhado dentro de outro elemento em bloco.
- Os **elementos em linha (inline)** são contidos dentro de elementos em bloco, envolvendo pequenas partes do conteúdo e não agrupamentos inteiros. Um elemento inline não fará com que uma nova linha apareça na página.
- Confira alguns exemplos de elemento em linha (inline):
 - Links <a>
 - Ênfase (negrito) ou

- Imagens
- Um elemento em linha não pode conter um elemento em bloco.

● **Elementos vazios**

- Nem todos os elementos HTML seguem o padrão de: <tag de abertura>conteúdo</tag de fechamento>. Alguns elementos são compostos apenas por uma única tag, podendo ou não conter atributos para a especificação de informações extras.
- Confira alguns exemplos de elementos vazios:
 - Quebra de linha

 - Linha horizontal <hr>
 - Imagem
- No caso dos exemplos acima, apenas o elemento img necessita de um atributo (src) para funcionar corretamente.

03 - Títulos, parágrafos e formatação

• Títulos, parágrafos e formatação

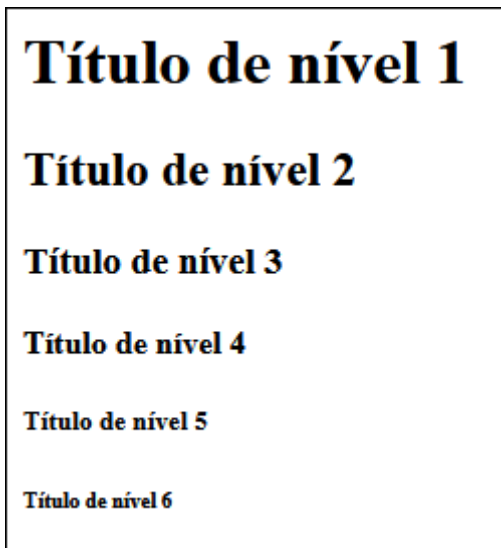


• Títulos

- O HTML possui elementos próprio para a produção de títulos e subtítulos que deseja incluir em sua página.
- Eles são importantes pois os mecanismos de busca, como o google, utilizam estes elementos para a indexação da estrutura e do conteúdo do site, além disso, eles servem para guiar o usuário pelo seu site, facilitando a localização de um determinado conteúdo, por exemplo.
- Os elementos de título estão divididos em seis níveis, começando do principal (mais importante) até o menos importante, a criação de um cabeçalho é simples, basta utilizar o elemento **h** seguido do número desejado, confira abaixo:

```
<h1>Título de nível 1</h1>
<h2>Título de nível 2</h2>
<h3>Título de nível 3</h3>
<h4>Título de nível 4</h4>
<h5>Título de nível 5</h5>
<h6>Título de nível 6</h6>
```


- Cada um dos elementos acima possui uma configuração própria para o tamanho da fonte, este tamanho pode ser modificado através do uso do CSS, veremos isso em uma outra parte do curso. O código acima produz o seguinte resultado:



- É importante utilizar os elementos de título apenas para títulos, não os utilize para deixar o texto maior ou em negrito.

• Parágrafos

- Nas primeiras lições você digitou o texto de forma solta dentro do elemento body, por mais que isso funcione, essa ação está errada, você sempre deve utilizar o elemento de parágrafo para digitar um texto.
- A criação de um parágrafo é simples, basta inserir o elemento **p** com o conteúdo desejado... Lembre-se, por não se tratar de um elemento vazio, o **p** depende da tag de abertura e de fechamento, confira:

```
<p>Eu sou um parágrafo!</p>  
<p>Eu sou um outro parágrafo!!</p>
```

- Um parágrafo sempre começa em uma nova linha, produzindo um bloco de texto, e os navegadores sempre adicionam

automaticamente um espaço em branco antes e depois deles (margem). O código acima produz o seguinte resultado:

Eu sou um parágrafo!

Eu sou um outro parágrafo!!

- Confira o seguinte exemplo:

```
<p>
  Eu sou um parágrafo
  criado com várias
  quebras de linha!
</p>
```

- No navegador, o resultado seria todo o parágrafo em uma única linha, conforme a imagem:

Eu sou um parágrafo criado com várias quebras de linha!

- Isso acontece pois o HTML ignora as quebras de linha criadas no editor, para criar as quebras de linha efetivamente, você precisa utilizar o elemento **br** visto na aula anterior, então o código deve ser o seguinte:

```
<p>
  Eu sou um parágrafo<br>
  criado com várias<br>
  quebras de linha!
</p>
```

- Desta forma o resultado seria o esperado:

Eu sou um parágrafo
criado com várias
quebras de linha!

• Formatação

- O HTML possui diferentes elementos para formatar o seu texto, definindo significados especiais para o trecho onde eles forem aplicados, alguns destes elementos possuem a mesma aparência, porém, eles foram pensados para diferentes aplicações, confira uma lista com os principais elementos abaixo:
 - Negrito: ``
 - Texto importante: ``
 - Itálico: `<i>`
 - Texto enfatizado: ``
 - Texto marcado: `<mark>`
 - Texto menor: `<small>`
 - Texto excluído: ``
 - Texto inserido: `<ins>`
 - Texto subscrito: `<sub>`
 - Texto sobrescrito: `<sup>`
- Os elementos apresentados acima são do tipo inline (em linha) e devem ser aplicados dentro de um outro elemento do tipo bloco, como o elemento de parágrafo (p). Confira o exemplo abaixo:

```
<p><b>Eu sou um texto com o elemento b</b></p>  
<p><strong>Eu sou um texto com o elemento strong</strong></p>
```

- O resultado do código acima é o seguinte:

Eu sou um texto com o elemento b

Eu sou um texto com o elemento strong

- Visualmente, ambos os textos têm a mesma característica (negrito), porém eles têm usos diferentes...
- O elemento **b** apenas define o texto em negrito, sem nenhuma importância extra. Já o elemento **strong** define um texto com uma importância maior, geralmente é exibido em negrito.

04 - Usando imagens

• A imagem no HTML



- A aplicação de imagens em uma página html é feita através do elemento **img**. O elemento img é vazio, ou seja, não possui conteúdo, sendo assim ele não possui uma tag de fechamento, possuindo dois atributos obrigatórios, são eles:
 - **src** - especifica o local da imagem;
 - **alt** - especifica um texto alternativo para a imagem;
- O uso de imagens em sua página pode aprimorar o design e a aparência de seu site.
- Analisando corretamente, as imagens não são tecnicamente inseridas na sua página, elas são vinculadas através de um outro local. Dessa forma, o elemento img é responsável por criar um espaço que vai conter a imagem.

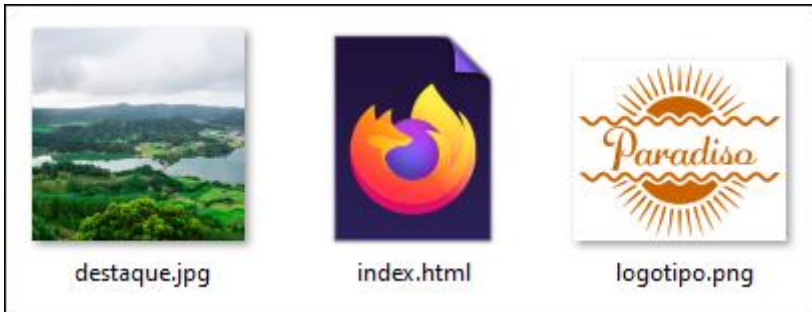
• A sintaxe do elemento

- Como dito anteriormente, o elemento img não possui uma tag de fechamento, e possui dois atributos, o src para especificar o local da imagem e o alt para definir um texto alternativo, confira a sintaxe do elemento:

```

```

- Imagine o seguinte cenário: as imagens que você deseja vincular à sua página estão no mesmo local que a própria página, ou seja, todos os arquivos estão na mesma pasta (mesmo nível)

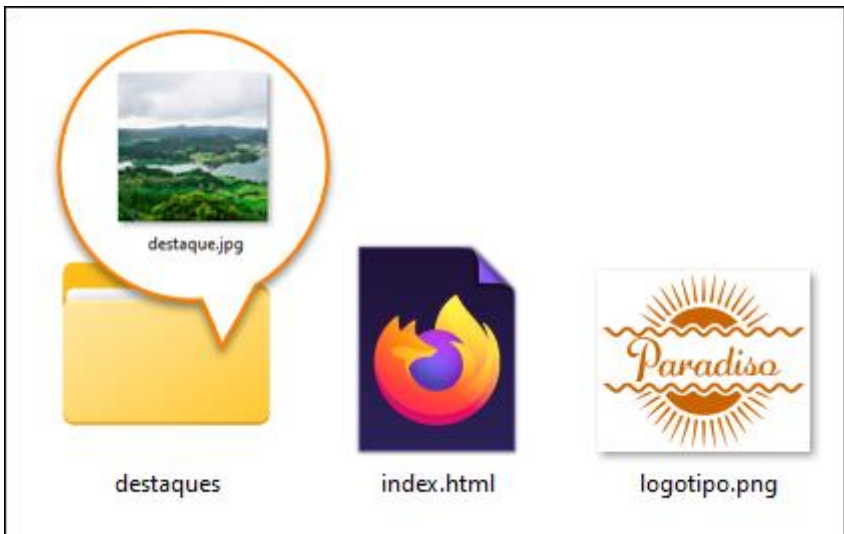


- O código necessário para a vinculação das imagens seria o seguinte:

```


```

- Veja que é importante especificar o nome e o formato do arquivo (.jpg e .png) - Se o nome do arquivo não for informado de forma correta, a imagem não será exibida na página.
- Agora, imagine a seguinte situação, a imagem "destaque.jpg" não está no mesmo nível que o arquivo index.html - A imagem está dentro de uma pasta chamada "destaques"



- Neste caso, o código para vinculação das imagens seria:

```


```

- Confira que o atributo **src** indica o "caminho" que o arquivo index.html deve fazer até chegar na imagem desejada...

• A importância do atributo alt

- O atributo alt é necessário pois fornece um texto alternativo para a imagem, esse texto é exibido se por algum motivo o usuário não puder visualizar a imagem, isso pode ocorrer devido à lentidão na conexão, problemas com o atributo src ou se o usuário utilizar um leitor de tela.
- Um leitor de tela é um programa de software que lê o código HTML e permite ao usuário "ouvir" o conteúdo.
- Os leitores de tela são úteis para pessoas com deficiência visual ou com deficiência de aprendizado.

• Os formatos mais comuns

- Confira a tabela abaixo com os formatos mais comuns e suportados nos principais navegadores

Abreviação	Formato do arquivo	Extensão do arquivo
GIF	Graphics Interchange Format	.gif
ICO	Microsoft Icon	.ico, .cur
JPEG	Joint Photographic Expert Group image	.jpg, .jpeg, .jif, .jpeg, .jpg
PNG	Portable Network Graphics	.png
SVG	Scalable Vector Graphics	.svg

- Importante: o carregamento de imagens grandes leva tempo e pode tornar sua página da web mais lenta. Use as imagens com cuidado.

- **O tamanho da imagem**

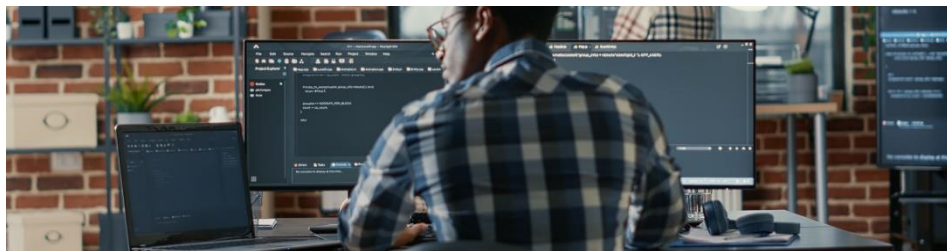
- Você pode ajustar o tamanho de uma imagem utilizando o atributo **style** (que usa) ou através dos atributos **width** e **height** - Nesta lição utilizaremos o width e o height, porém, é recomendado utilizar o atributo **style**.
- Os atributos **width** e **height** sempre definem a largura e a altura da imagem em pixels. É sempre interessante especificar a largura e a altura de uma imagem, se elas não forem especificadas, a página da Web poderá piscar enquanto a imagem é carregada.
- Os atributos width e height podem ser aplicados individualmente, a utilização dos atributos é bem simples, confira abaixo:

```

```

- O código acima deixaria a imagem com 200px de largura e ajustaria a altura da imagem de forma automática (proporcionalmente).

05 - Criando Links (Hyperlinks)



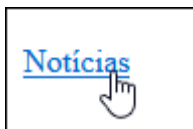
- Os links são encontrados em quase todas as páginas da internet. Eles que permitem a navegação de uma página para outra.

Os links, também conhecidos como hyperlinks, são elementos clicáveis que, ao serem clicados, direcionam o visitante para um outro local ou documento. Ao posicionar o mouse sobre um link, o cursor do mouse se transforma em uma pequena mão com o dedo indicador levantado.

• A sintaxe do elemento

- Um link html é definido através do elemento **a**. Ele é composto das tags de abertura e fechamento, do atributo **href** que é responsável por indicar o destino do usuário, e do conteúdo (que fica entre as tags) que será a parte visível (clicável) pelo usuário.
- Confira um exemplo:

```
<a href="noticias.html">Notícias</a>
```

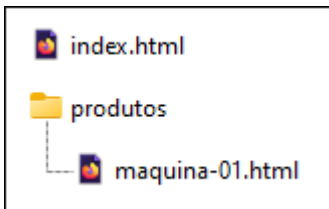


- O código acima resultaria no seguinte:
- Veja que o conteúdo do atributo href não é visível para o usuário;
- Um link não precisa ser texto. Um link pode ser uma imagem ou qualquer outro elemento HTML!
- Veja que a cor do texto também ficou diferente, o texto ficou azul com um sublinhado, essa é a aparência padrão de um link. Por padrão, nos browser, você verá o seguinte:
 - Um link não visitado fica sublinhado e azul
 - Um link visitado fica sublinhado e roxo

- Um link ativo fica sublinhado e vermelho
- A aparência de um link pode ser alterada através do uso do CSS.

• O atributo href

- Assim como ocorre com o atributo src do elemento img, o atributo href necessita da indicação correta de onde o arquivo que será carregado está, ou seja, imagine a seguinte situação:



- A página "index.html" está no mesmo nível que a pasta "produtos", a página de destino que desejamos acessar (maquina-01.html) está dentro da pasta produtos... Neste caso, o link precisaria ser construído da seguinte maneira:

```
<a href="produtos/maquina-01.html">Máquina 01 - Corte</a>
```

- Se as páginas estivessem no mesmo nível, bastaria inserir o nome do arquivo (maquina-01.html) para criar um link.

• URL absoluta e URL relativa

- O conteúdo do atributo href pode ser uma URL absoluta ou relativa, a diferença entre elas é simples...
- Se o conteúdo de destino estiver em algum outro site, você deverá preencher o atributo href utilizando o endereço completo do site, isso é conhecido como **url absoluta**, por exemplo:

```
<a href="https://g1.globo.com">Confira as notícias de hoje!</a>
```

- Caso o conteúdo esteja no mesmo site (link local) basta inserir o nome do arquivo, conforme os exemplos vistos anteriormente, esse formato é conhecido como **url relativa**, por exemplo:

```
<a href="noticias.html">Notícias</a>
```

- Ou

```
<a href="paginas/noticias.html">Confira as notícias de hoje!</a>
```

- Uma url relativa nunca possuirá a parte "https://" ou "https://www"

• Os atributos target e title

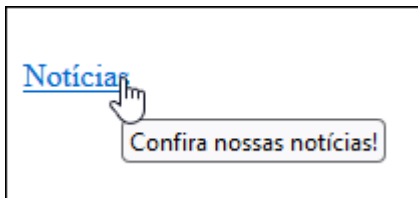
- Além o atributo **href** (que é obrigatório) o elemento a também pode possuir mais alguns atributos, falaremos sobre eles agora...
- O atributo **target** permite definir onde o link será aberto, por padrão, um link é aberto sempre na mesma guia que o usuário está utilizando atualmente, porém, isso pode ser modificado, confira os possíveis valores do atributo target:
 - **_self** - É o valor padrão. Abre o documento na mesma janela/guia em que foi clicado.
 - **_blank** - Abre o documento em uma nova janela ou guia.
 - **_parent** - Abre o documento no quadro pai.
 - **_top** - Abre o documento em todo o corpo da janela.
- A utilização do atributo segue o padrão visto para todos os outros atributos, basta incluí-lo na tag de abertura do elemento:

```
<a href="noticias.html" target="_blank">Notícias</a>
```

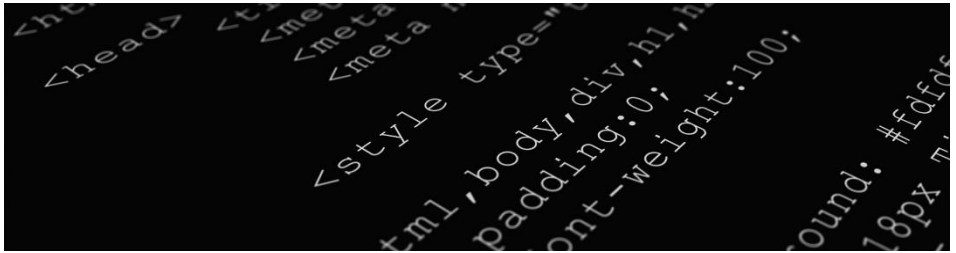
- No exemplo acima, a página noticias.html seria aberta em uma nova janela ou guia quando o link fosse clicado.
- Já o atributo **title** define uma informação extra sobre o elemento, o conteúdo dele é visível quando o usuário passa o mouse sobre o link, exemplo:

```
<a href="noticias.html" title="Confira nossas notícias!">Notícias</a>
```

- O conteúdo do atributo title seria visualizado da seguinte maneira:



06 - Criando tabelas HTML



- As tabelas em HTML permite que você arrume dados em linhas e colunas. Uma tabela básica é composta por três elementos, são eles:
 - **table** - define o espaço da tabela
 - **tr** - define a linha da tabela
 - **td** - define a célula da tabela
- A construção da tabela é simples, o elemento inicial é o **table**. Dentro dele, devemos inserir um **tr**, para identificar a primeira linha... Com a primeira linha definida, agora basta inserir a quantidade desejada de **td**, por exemplo:

```
<table>
  <tr>
    <td>Célula 01</td>
    <td>Célula 02</td>
    <td>Célula 03</td>
  </tr>
</table>
```

- O código acima resultaria em uma tabela composta por uma única linha, com três células:

Célula 01	Célula 02	Célula 03
-----------	-----------	-----------

- Para criar uma nova linha, você precisaria inserir um novo **tr**, contendo a mesma quantidade de **tds** da linha anterior

```
<table>
  <tr>
    <td>Célula 01</td>
    <td>Célula 02</td>
    <td>Célula 03</td>
  </tr>
  <tr>
    <td>Célula 01</td>
    <td>Célula 02</td>
    <td>Célula 03</td>
  </tr>
</table>
```

- Confira o resultado do código acima:

Célula 01	Célula 02	Célula 03
Célula 01	Célula 02	Célula 03

- Ignorando a parte visual (que foi feita com CSS), confira a seguinte tabela:

Nome	Telefone	Email
Claudia	3333-3333	claudia@email.com.br
Diego	4444-4444	diego@email.com
Paulo	5555-5555	paulo@email.com

- O código responsável pela criação dela seria o seguinte:

```
<table>
  <tr>
    <td>Nome</td>
    <td>Telefone</td>
    <td>Email</td>
  </tr>
  <tr>
    <td>Claudia</td>
    <td>3333-3333</td>
    <td>claudia@email.com.br</td>
  </tr>
  <tr>
    <td>Diego</td>
    <td>4444-4444</td>
    <td>diego@email.com</td>
  </tr>
  <tr>
    <td>Paulo</td>
    <td>5555-5555</td>
    <td>paulo@email.com</td>
  </tr>
</table>
```

• Linhas e Células

- Toda linha é iniciada com a tag **<tr>** e é finalizada com a tag **</tr>**.
- O **tr** significa table row (linha da tabela)
- Toda célula é iniciada com **<td>** e é finalizada com **</td>** - tudo que estiver presente entre essas duas tags é o conteúdo da célula
- O **td** significa table data (dados da tabela). As células são os containers da tabela e podem conter todos os tipos de elementos HTML: texto, imagens, listas, outras tabelas, etc.

- **Cabeçalhos**

- Se necessário, você pode criar um cabeçalho para a sua tabela, para isso você deve utilizar o elemento **th** ao invés do **td** na primeira linha da sua tabela.
- Por padrão, os textos presentes dentro do th são centralizados e apresentados em negrito. Confira:

```
<table>
  <tr>
    <th>Nome</th>
    <th>Telefone</th>
    <th>Email</th>
  </tr>
  <tr>
    <td>Claudia</td>
    <td>3333-3333</td>
    <td>claudia@email.com.br</td>
  </tr>
  <tr>
    <td>Diego</td>
    <td>4444-4444</td>
    <td>diego@email.com</td>
  </tr>
  <tr>
    <td>Paulo</td>
    <td>5555-5555</td>
    <td>paulo@email.com</td>
  </tr>
</table>
```

Nome	Telefone	Email	<th>
Claudia	3333-3333	claudia@email.com.br	<td>
Diego	4444-4444	diego@email.com	
Paulo	5555-5555	paulo@email.com	

- **Bordas**

- Por padrão, as bordas da tabela e das células não possuem um visual definido, o ajuste da aparência é feito através do CSS. Com o CSS podemos deixar as bordas arredondadas, com diferentes tipos de linha e ajustarmos outras diversas opções... Porém, existe um atributo que permite exibir uma aparência básica na tabela, o atributo **border**.
- O uso dele é muito simples, basta inserir o atributo **border** na tag de abertura do elemento **table**, o valor do atributo é numérico e determina a grossura da borda, confira:

```
<table border="2">
  <tr>
    <th>Nome</th>
    <th>Telefone</th>
    <th>Email</th>
  </tr>
  <tr>
    <td>Claudia</td>
    <td>3333-3333</td>
    <td>claudia@email.com.br</td>
  </tr>
  <tr>
    <td>Diego</td>
    <td>4444-4444</td>
    <td>diego@email.com</td>
  </tr>
  <tr>
    <td>Paulo</td>
    <td>5555-5555</td>
    <td>paulo@email.com</td>
  </tr>
</table>
```

Nome	Telefone	Email
Claudia	3333-3333	claudia@email.com.br
Diego	4444-4444	diego@email.com
Paulo	5555-5555	paulo@email.com

07 - Criando listas HTML



- Como o nome já diz, as listas html permitem que você agrupe um conjunto de itens relacionados em listas, elas podem ser de dois tipos: **Listas não ordenadas** ou **listas ordenadas**.

• As listas não ordenadas

- Por padrão, as listas não ordenadas apresentam seus itens com pequenos marcadores circulares pretos.
- Para iniciar a criação da sua lista, primeiro você precisa definir qual o tipo da lista, no caso, para a lista não ordenada utilizamos o elemento **ul** que significa unordered list (lista não ordenada).
- Para criar um item da sua lista, você deve utilizar o elemento **li** que significa list item (item da lista).
- A estrutura segue a ideia utilizada para a tabela, imaginando que o tr representa o ul e o td representa o li, ou seja:

```
<ul>
  <li>Item 01</li>
  <li>Item 02</li>
  <li>Item 03</li>
</ul>
```

- O código acima resultaria na seguinte lista:

- Item 01
- Item 02
- Item 03

• Listas ordenadas

- Por padrão, uma lista ordenada apresenta seus itens com números em ordem crescente.
- A criação da lista ordenada é bem parecida com a criação da lista não ordenada, a única diferença é o elemento inicial, ao invés de utilizar o **ul**, você deve utilizar o **ol** que significa ordered list (lista ordenada), confira:

```
<ol>
  <li>Item 01</li>
  <li>Item 02</li>
  <li>Item 03</li>
</ol>
```

- O resultado do código acima seria o seguinte:

```
1. Item 01
2. Item 02
3. Item 03
```

- Dica! A criação das listas também pode ser feita com a abreviação Emmet, neste caso, após definição do tipo da lista (elemento inicial), basta inserir o elemento li multiplicado (*) pela quantidade de itens desejadas, por exemplo:

```
ul>li*8
```

```
ul>li*8
```

```
<ul>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

08 - Criando formulários



- Um formulário HTML é utilizado para coletar dados preenchidos pelo usuário. Normalmente os dados digitados são enviados para um servidor onde são processados.
- Você aprenderá a montar a estrutura de um formulário HTML, ele não será funcional pois apenas o HTML não é capaz de efetuar a comunicação com o servidor, isso deve ser feito através de uma outra linguagem como PHP ou ASP.

• O elemento form

- Assim como as listas e as tabelas html, o formulário possui um elemento para indicar a sua criação, esse elemento é conhecido como **form**.
- O elemento **form** é o container para todos os outros elementos do seu formulário

```
<form>  
  * elementos do form *  
</form>
```

- **O elemento input**

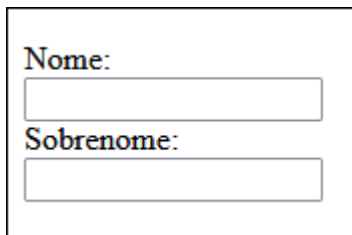
- O elemento **input** é o mais utilizado no formulário, ele pode ser exibido de diversas maneiras, variando de acordo com o valor do atributo **type**, confira os diferentes tipos abaixo:
 - `<input type="button">`
 - `<input type="checkbox">`
 - `<input type="color">`
 - `<input type="date">`
 - `<input type="datetime-local">`
 - `<input type="email">`
 - `<input type="file">`
 - `<input type="hidden">`
 - `<input type="image">`
 - `<input type="month">`
 - `<input type="number">`
 - `<input type="password">`
 - `<input type="radio">`
 - `<input type="range">`
 - `<input type="reset">`
 - `<input type="search">`
 - `<input type="submit">`
 - `<input type="tel">`
 - `<input type="text">`
 - `<input type="time">`
 - `<input type="url">`
 - `<input type="week">`
- O valor padrão do atributo type é o "text".
- Além do atributo **type**, é importante adicionar o atributo **id** que serve para identificar o elemento, o valor deste atributo é livre, podendo ser preenchido como você desejar.

• O elemento label

- O elemento **label** define uma "etiqueta", um "rótulo", para um determinado elemento do formulário. Ele é importante pois ajuda usuários que tem dificuldade em clicar em regiões pequenas (como caixas de seleção) e também porque é lido pelos leitores de tela, ajudando a navegação de usuários que problemas de visão.
- O elemento label possui um atributo chamado **for** que deverá ter o mesmo valor que o atributo **id do input que será vinculado**.
- Confira um exemplo de formulário abaixo, contendo o label vinculado com seus respectivos inputs:

```
<form>
  <label for="campo-nome">Nome:</label><br>
  <input type="text" id="campo-nome"><br>
  <label for="campo-sobrenome">Sobrenome:</label><br>
  <input type="text" id="campo-sobrenome">
</form>
```

- O resultado do código seria seguinte:



Nome:

Sobrenome:

• O elemento textarea

- O elemento textarea define uma caixa de texto com múltiplas linhas, geralmente utilizada para que o usuário digite uma mensagem, confira um exemplo muito comum:

```
<textarea id="campo-mensagem" cols="50" rows="10"></textarea>
```

- Confira que, além do atributo id, o textarea também possui dois outros atributos: **cols** e **rows**.
- O atributo **cols** é responsável por definir a largura da caixa de texto; O atributo **rows** é responsável por definir a altura da caixa de texto.

- Confira o **textarea** acima em conjunto com o formulário criado anteriormente:

```
<form>
  <label for="campo-nome">Nome:</label><br>
  <input type="text" id="campo-nome"><br>
  <label for="campo-sobrenome">Sobrenome:</label><br>
  <input type="text" id="campo-sobrenome"><br>
  <label for="campo-mensagem">Digite sua mensagem:</label><br>
  <textarea id="campo-mensagem" cols="50" rows="10"></textarea>
</form>
```

• O elemento **select**

- O elemento **select** cria uma lista do tipo "drop-down", muito utilizada para a listagem de estados do país, por exemplo.
- A criação dele é simples, inicialmente você cria a tag de abertura e de fechamento do elemento **select**

```
<select id="campo-diaria">

</select>
```

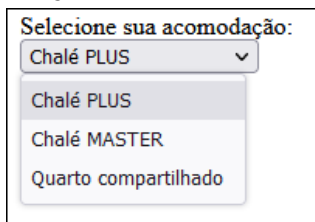
- Como conteúdo do elemento **select**, você irá criar os elementos **options** responsáveis por indicar as opções disponíveis para a seleção

```
<select id="campo-diaria">
  <option value="chale-plus">Chalé PLUS</option>
  <option value="chale-master">Chalé MASTER</option>
  <option value="quarto">Quarto compartilhado</option>
</select>
```

- O elemento **option** precisa do atributo **value** para informar a opção que o usuário selecionar... O conteúdo do elemento será visualizado pelo usuário no navegador
- Confira o resultado do código:

```
<label for="campo-diaria">Selecione sua acomodação:</label><br>
<select id="campo-diaria">
  <option value="chale-plus">Chalé PLUS</option>
  <option value="chale-master">Chalé MASTER</option>
  <option value="quarto">Quarto compartilhado</option>
</select>
```

- Quando o usuário clicar na lista, as opções ficarão visíveis para a seleção



• O input type "submit"

- Para o envio do formulário, você deve criar um input com o tipo "submit" - Além do atributo type, você também deve utilizar o atributo "value" para definir o texto que será apresentado dentro do botão, confira:

```
<input type="submit" value="Enviar">
```

- O código acima resultaria no seguinte botão



• O atributo name

- É importante adicionar o atributo **name** em seus inputs e em todos os elementos principais do formulário (textarea, select), ele garante que (quando o formulário estiver funcional) o valor do

campo seja enviado corretamente. Este atributo serve para identificar o campo e pode conter o mesmo valor do atributo **id**;

- Confira um form criado com todos os atributos necessários:

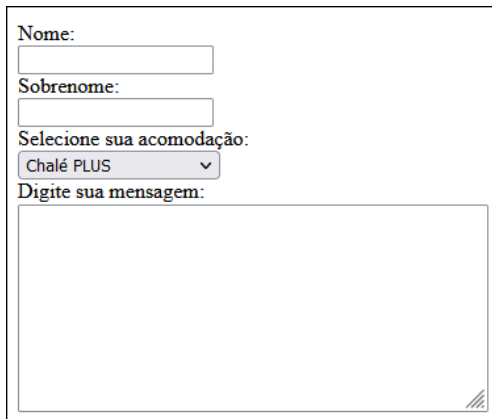
```
<form>
  <label for="campo-nome">Nome:</label><br>
  <input type="text" id="campo-nome" name="campo-nome"><br>

  <label for="campo-sobrenome">Sobrenome:</label><br>
  <input type="text" id="campo-sobrenome" name="campo-sobrenome"><br>

  <label for="campo-diaria">Selecione sua acomodação:</label><br>
  <select id="campo-diaria" name="campo-diaria">
    <option value="chale-plus">Chalé PLUS</option>
    <option value="chale-master">Chalé MASTER</option>
    <option value="quarto">quarto compartilhado</option>
  </select><br>

  <label for="campo-mensagem">Digite sua mensagem:</label><br>
  <textarea id="campo-mensagem" name="campo-mensagem" cols="50" rows="10"></textarea>
</form>
```

- O resultado do código seria o seguinte:



Nome:

Sobrenome:

Selecione sua acomodação:

Digite sua mensagem:

- Ao criar os elementos do formulário (input, textarea, select) com o uso da abreviação Emmet, eles já são criados com os atributos necessários.
- Para visualizar a lista de inputs disponíveis com o Emmet basta digitar **input**

• Os atributos method e action

- A tag de abertura do formulário pode possuir dois atributos que definem como o formulário funcionará, são eles:
 - **Action:** O atributo define a ação a ser executada quando o formulário é enviado. Normalmente, os dados do formulário são enviados para um arquivo no servidor quando o usuário clica no botão enviar.
 - **Method:** O atributo method especifica o método HTTP a ser usado ao enviar os dados do formulário. Os dados do formulário podem ser enviados como variáveis de URL (com method="get") ou como transação HTTP post (com method="post"). Por padrão, o HTML utiliza o método get.
- Ambos os atributos são utilizados para a criação de um formulário funcional, conforme explicado anteriormente, apenas com o uso do HTML, um formulário não é funcional.

09 - Introdução ao CSS: Sintaxes e Seletores



• O que é CSS?

- O CSS é um acrônimo para cascading style sheets, que em português significa "folhas de estilo em cascata", ele é uma linguagem de folhas de estilo usada para descrever a apresentação de um documento escrito em HTML. O CSS descreve como elementos devem ser mostrados na tela.
- É interessante ressaltar que, assim como o HTML, o CSS não é uma linguagem de programação. Porém, diferentemente do HTML ele não é uma linguagem de marcação, ele é uma linguagem de folhas de estilo.

• A Sintaxe do CSS

- Confira um exemplo de estrutura CSS:



- O **seletor** determina o que será afetado em sua página. No exemplo acima, todos os parágrafos existentes seriam afetados.
- A **declaração** é composta por todas as propriedades e valores existentes dentro das chaves { }.
- As **propriedades** determinam a alteração que será feita. No exemplo acima, os parágrafos terão o tamanho e a cor modificados.

- Os **valores** complementam as propriedades. No exemplo acima, os parágrafos ficariam com a cor azul e com o tamanho 12px.
- Outras informações importantes sobre a sintaxe:
- Cada linha de comando deve ser envolvida em chaves ({}).
- Dentro de cada declaração, você deve usar dois pontos (:) para separar a propriedade de seus valores.
- Dentro de cada conjunto de regras, você deve usar um ponto e vírgula (;) para separar cada declaração da próxima.

• Os seletores CSS

- Os seletores CSS são usados para selecionar os elementos HTML que você deseja estilizar. Podemos dividir os seletores em cinco categorias:
 - Seletores simples (seleciona elementos com base no nome, id, classe)
 - Seletores de combinação (seleciona elementos com base em um relacionamento específico entre eles)
 - Seletores de pseudo-classe (seleciona elementos com base em um determinado estado)
 - Seletores de pseudo-elementos (seleciona e estiliza uma parte de um elemento)
 - Seletores de atributo (seleciona elementos com base em um atributo ou valor de atributo)

• O seletor de elemento HTML

- O seletor de elemento HTML como o nome indica, permite a seleção de todos os elementos HTML do tipo indicado, confira um exemplo:

```
h1 {color: blue; font-size: 32px;}
```

- No exemplo acima, todos os elementos h1 da página teriam a mesma configuração (cor azul e tamanho 32)

• O seletor de id

- O seletor de elemento id permite selecionar os elementos que possuírem o ID especificado. A especificação de um ID é feita através do caractere #, confira um exemplo:

```
#header {color: blue; font-size: 32px;}
```

- No exemplo acima, qualquer elemento HTML que tiver o atributo id com o valor header, ficará com a mesma configuração

• O seletor de classe

- O seletor de classe funciona da mesma maneira que o seletor de id, a diferença é que a especificação da classe é feita através do caractere . (ponto final), confira um exemplo:

```
.menu {color: blue; font-size: 32px;}
```

- No exemplo acima, qualquer elemento HTML que tiver o atributo class com o valor menu, ficará com a mesma configuração

• O seletor universal

- O seletor universal afeta todos os elementos HTML da página, ele é indicado através do caractere *, confira o exemplo:

```
* {color: blue; font-size: 32px;}
```

- No exemplo acima, todos os textos ficariam com a cor azul e com o tamanho 32px, independentemente do elemento.

• Como utilizar o CSS

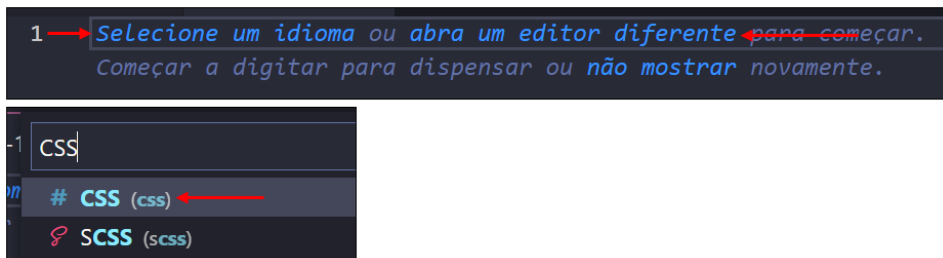
- Existem três formas de utilizar o CSS para formatar a sua página HTML, são elas:
 - CSS externo (External CSS)
 - CSS interno (Internal CSS)
 - CSS em linha (Inline CSS)

• O CSS externo

- Neste modo você indica um caminho externo para uma folha de estilo (arquivo css) e, utilizando apenas este arquivo, tem a capacidade de modificar um site inteiro.
- Para usar este modo você deve incluir uma referência ao arquivo de folha de estilo dentro da seção head do seu html, para cada página que você desejar formatar, a referência deverá ser incluída conforme o exemplo abaixo:

```
<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>Pousada Paradiso</title>
  <link rel="stylesheet" href="css/estilo.css">
</head>
```

- Veja que o atributo **rel** é responsável por indicar o tipo do arquivo e o atributo **href** é responsável por indicar o local do arquivo, seguindo a ideia utilizada para imagens e links.
- Uma folha de estilo pode ser escrita em qualquer editor de texto e deverá ser salva com o formato .css - No Visual Studio Code é possível selecionar a linguagem da mesma forma que feito com o HTML:



• O CSS interno

- O CSS interno é construído diretamente na página HTML que você deseja modificar, ele não terá a capacidade de modificar outras páginas.
- Para utilizar este modo, basta criar um elemento chamado style, dentro do elemento head do seu site...

- A construção do código CSS segue padrão. Confira o exemplo:

```

<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>Pousada Paradiso</title>
  <style>
    body {
      font-family: 'Quicksand', sans-serif;
      padding: 0px;
      margin: 0px;
      background: #fff;
      text-align: justify;
    }

    .container {
      max-width: 940px;
      margin: auto;
      overflow: auto;
    }
  </style>
</head>

```

• O CSS em linha

- O CSS em linha é feito pensando em modificar apenas um elemento HTML, para usá-lo, basta inserir o atributo style no elemento desejado e, como valor do atributo, utilizar as propriedades CSS desejadas, confira:

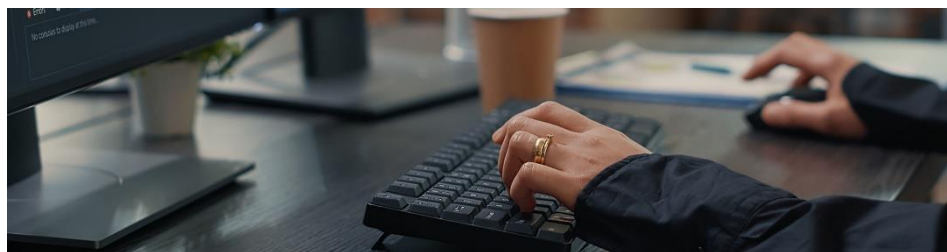
```

<h1 style="color: #ff9900;">Pousada Paradiso</h1>

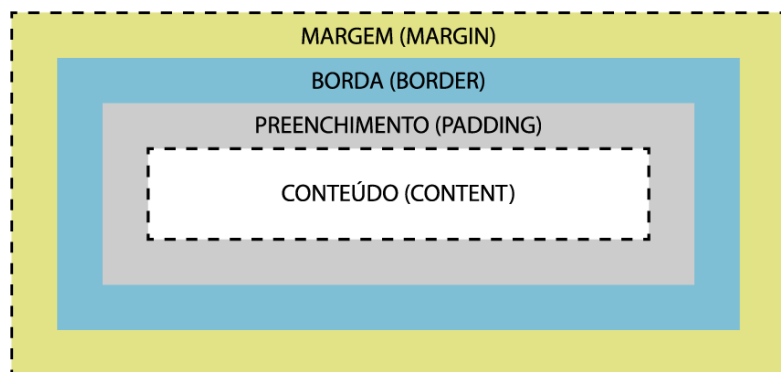
```



10 - O modelo de caixa do CSS



- No CSS, o "modelo de caixa" diz respeito ao design e o layout do site. Esse "modelo de caixa" é encarado basicamente como uma caixa que envolve cada elemento do HTML, ele consiste em:
 - Margem (margin)
 - Borda (border)
 - Preenchimento (padding)
 - Conteúdo (content)
- Confira uma imagem que representa o modelo:



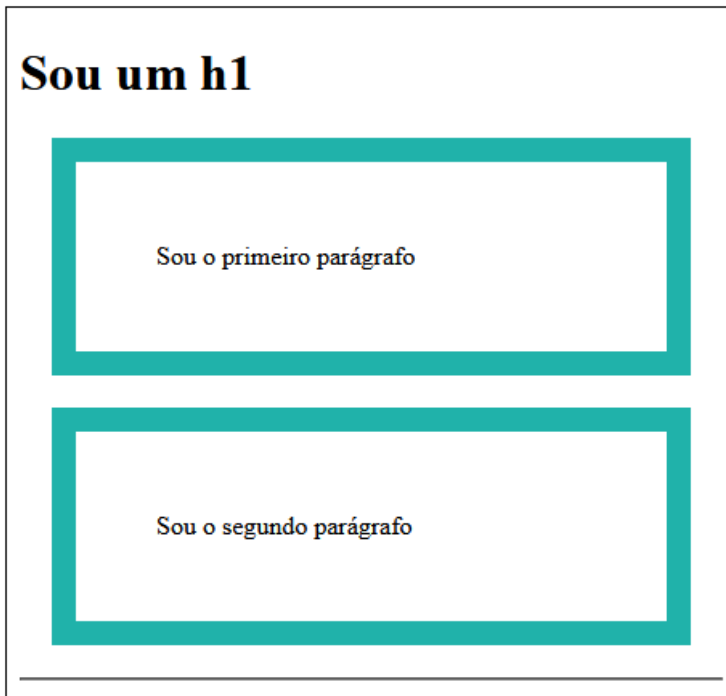
- **Conteúdo (content):** É o conteúdo da caixa, a área onde a imagem ou o texto é exibido;
- **Preenchimento (padding):** Livra uma área ao redor do conteúdo. Esse preenchimento é transparente.
- **Borda (border):** É a área que vai ao redor do preenchimento (padding) e do conteúdo (content);
- **Margem (margin):** É a área ao redor da borda. Essa margem é transparente.
- No código CSS, a aplicação seria feita da seguinte maneira:

```
p{
  border: 15px solid lightseagreen;
  padding: 50px;
  margin: 20px;
}
```

- O código acima afetaria todos os parágrafos (p) da página; Imagine o seguinte código html

```
<h1>Sou um h1</h1>
<p>Sou o primeiro parágrafo</p>
<p>Sou o segundo parágrafo</p>
<hr>
```

- Com a aplicação do CSS visto anteriormente, o resultado seria o seguinte:



-
- **Definindo a largura e a altura de um elemento**

- Ao utilizar as propriedades CSS width (largura) e/ou height (altura) você define o tamanho da área de conteúdo, para saber o tamanho total do seu objeto você deve considerar o espaço utilizado por todos os itens que compõem a caixa do CSS. Ou seja, se você analisar o código abaixo:
- Notará que a largura do elemento div está definida em 300px, porém, ainda é necessário adicionar as medidas presentes nas propriedades padding e border; o resultado seria o seguinte:

```
div {  
  width: 300px;  
  height: 100px;  
  padding: 10px;  
  border: 5px solid gray;  
  margin: 0;  
}
```

- A largura seria definida da seguinte maneira:
300px de width (largura) +
20px de padding (preenchimento esquerdo e direito) +
10px de border (borda esquerda e direita)
0px de margin
Totalizando: 330px de largura;
- A altura seria definida da seguinte maneira:
100px de height (altura) +
20px de padding (preenchimento superior e inferior) +
10px de border (borda superior e inferior)
0px de margin
Totalizando: 130px de altura;
- **Ou seja, o nosso elemento teria: 330px de largura por 130px de altura**

• A propriedade box-sizing

- A propriedade `box-sizing` permite a inclusão do `padding` e da `border` na largura e altura totais de um elemento. Existem três valores possíveis para a propriedade:
 - **content-box**: é o valor padrão, nele as propriedades `width` (largura) e `height` (altura) são medidas incluindo apenas o conteúdo, sem o `padding`, `border` ou `margin`.
 - **padding-box**: é um valor experimental, nele as propriedades `width` (largura) e `height` (altura)
 - **border-box**: com este valor as propriedades de largura (`width`) e de altura (`height`) incluem o tamanho `padding` e `border`, mas não incluem a propriedade `margin`.
- No exemplo visto anteriormente, se a propriedade `box-sizing` fosse adicionada com o valor `border-box`, o resultado seria um elemento com 300px de largura e 100px de altura, conforme indicado no código. Isso aconteceria, pois, as propriedades `padding` e `border` seriam incluídos no elemento final, sem a soma.
- Confira o código:

```
.div1{
  width: 300px;
  height: 100px;
  padding: 10px;
  border: 5px solid gray;
  margin: 0px;
}

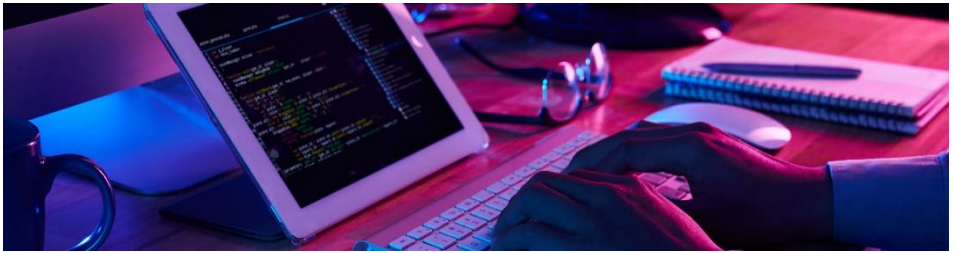
.div2{
  width: 300px;
  height: 100px;
  padding: 10px;
  border: 5px solid lightskyblue;
  margin: 0px;
  box-sizing: border-box;
}
```

- Confira o resultado de dois divs criados com as respectivas classes:

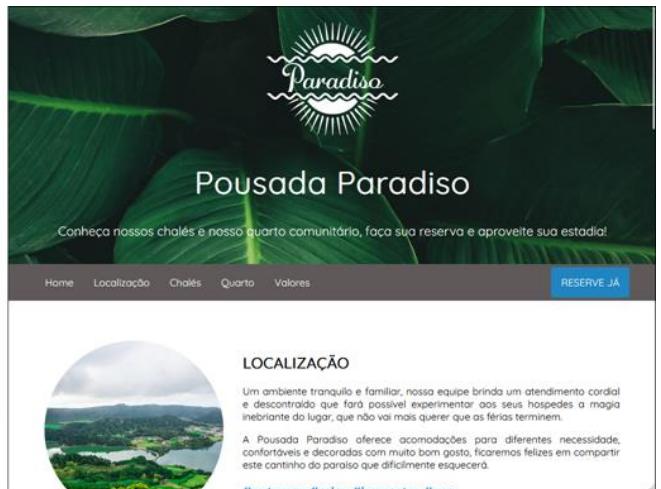
EU USO A CLASSE, DIV1, SEM O BOX-SIZING

EU USO A CLASSE, DIV2, COM O BOX-SIZING

11 - CSS Responsivo: RWD



- O termo RWD significa "responsive web design", em português pode ser traduzido como "design web responsivo", o conceito principal aqui é fazer o seu site ter uma boa aparência em todos os dispositivos, independentemente do tamanho da tela.
- O design responsivo para web utiliza apenas HTML e CSS, não necessitando de JavaScript, por exemplo. Confira o resultado final site criado no curso em duas versões, a mobile (celular) e a "normal" do navegador



- **O viewport**
- Viewport é o nome dado para a área visível do site pelo usuário, ela varia de acordo com o dispositivo que o usuário estiver utilizando. Antes do surgimento dos tablets e celulares, as páginas da web eram projetadas apenas para as telas de computador, e era comum que elas tivessem um design estático e um tamanho fixo.

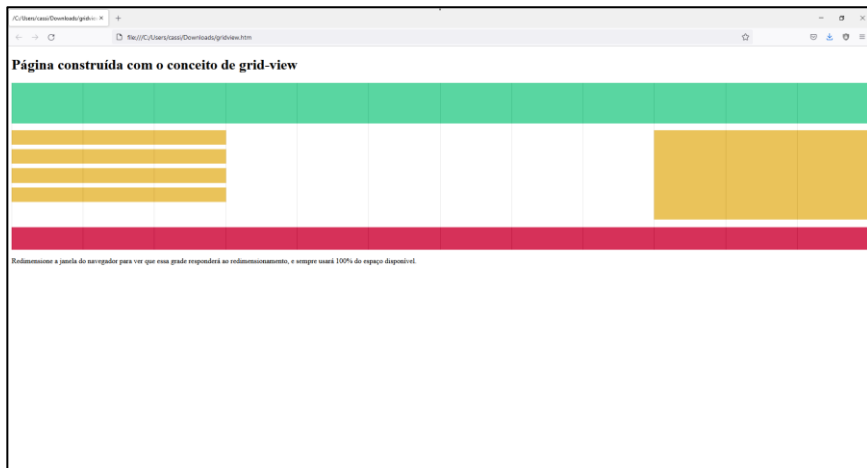
- O HTML5 introduziu um método para permite assumir o controle do viewport, por meio da tag <meta>. Você deve incluir o seguinte elemento em todas as suas páginas da web:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

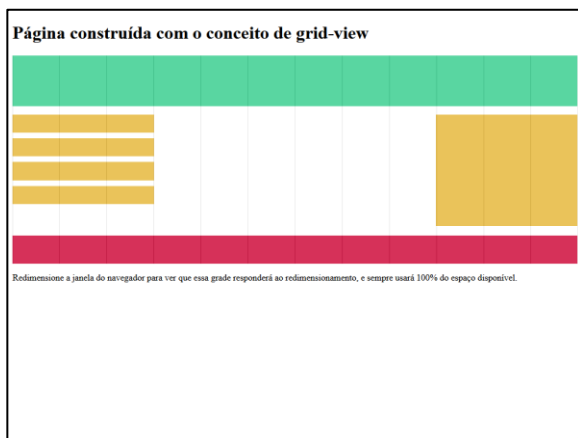
- Esse elemento fornece ao navegador instruções sobre como controlar as dimensões e o dimensionamento da página.
- A parte "**width=device-width**" define a largura da página para seguir a largura da tela do dispositivo (que varia dependendo do dispositivo).
- A parte "**initial-scale=1.0**" define o nível de zoom inicial quando a página é carregada pela primeira vez pelo navegador.
- É importante se atentar ao fato que os usuários estão acostumados a rolar sites verticalmente em computadores e dispositivos móveis - mas não horizontalmente!
- Com isso em mente, existem algumas regras a serem seguidas:
 1. NÃO use elementos grandes com largura fixa - Por exemplo, se uma imagem for exibida com uma largura maior que a viewport, ela poderá fazer com que a janela de visualização role horizontalmente. Lembre-se de ajustar este conteúdo para caber na largura da janela de visualização.
 2. NÃO permita que o conteúdo dependa de uma largura de viewport específica para renderizar corretamente - Como as dimensões da tela e a largura em pixels CSS variam muito entre os dispositivos, o conteúdo não deve depender de uma largura de janela de visualização específica para renderizar bem.
 3. Use consultas de mídia CSS para aplicar estilos diferentes para telas pequenas e grandes - Definir grandes larguras absolutas de CSS para elementos de página fará com que o elemento seja muito largo para a janela de visualização em um dispositivo menor. Em vez disso, considere usar valores de largura relativos, como largura: 100%. Além disso, tenha cuidado ao usar valores de posicionamento absolutos grandes. Isso pode fazer com que o elemento fique fora da janela de visualização em dispositivos pequenos.

- **O conceito do grid-view**

- Muitas páginas de hoje em dia são baseadas no modelo grid-view. Esse modelo nada mais é que dividir a página em colunas idênticas, geralmente as páginas são divididas em doze colunas, totalizando 100% da largura, e que vão se ajustando conforme o navegador for redimensionado.
- Confira os exemplos abaixo:



Página visualizada em 1920x1080



Página visualizada em 1024x768

- Confira que em ambas as visualizações os elementos foram adaptados para ocupar 100% da tela;

- **Construindo o grid-view responsivo**

- O primeiro passo para a construção do grid-view responsivo é determinar que todos os elementos do HTML terão as bordas (borders) e os preenchimentos (padding) incluídos na largura e na altura total do elemento, ou seja, você deve utilizar o **box-sizing** com o valor **border-box**, para que a propriedade seja aplicada a todos os elementos deve-se utilizar o seletor *, da seguinte maneira:

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

- Depois disso, podemos iniciar a construção das colunas, levando em consideração que iremos utilizar 12 colunas, precisamos pegar o valor total (100%) e dividi-lo por 12...
- $100\% / 12 = 8,33\%$
- **Ou seja, cada coluna deverá ter 8,33% de largura...**
- Então basta criar as classes responsáveis por cada coluna, seguindo a ideia abaixo:

```
.col-1 {width: 8.33%;}  
.col-2 {width: 16.66%;}  
.col-3 {width: 25%;}  
.col-4 {width: 33.33%;}  
.col-5 {width: 41.66%;}  
.col-6 {width: 50%;}  
.col-7 {width: 58.33%;}  
.col-8 {width: 66.66%;}  
.col-9 {width: 75%;}  
.col-10 {width: 83.33%;}  
.col-11 {width: 91.66%;}  
.col-12 {width: 100%;}
```

- Ou seja, o elemento que utilizar a classe "col-1" vai ocupar 8.33% da tela; O elemento que utilizar a classe "col-2" ocupará 16.66% - E assim sucessivamente.
- Agora, você precisa definir que todas essas classes "col-" precisam ficar lado a lado, para isso você deve utilizar a propriedade float

• **A propriedade float**

- A propriedade float define como um elemento deve flutuar em sua página, podendo conter os seguintes valores:
 - left - O elemento flutua à esquerda de seu contêiner
 - right - O elemento flutua à direita de seu contêiner
 - none - O elemento não flutua (será exibido exatamente onde ocorre no texto). Isso é padrão
 - inherit - O elemento herda o valor float de seu elemento pai
- No caso, para a criação do grid-view é necessário que todas as colunas estejam flutuando à esquerda uma da outra. Ao invés de inserir a propriedade manualmente em todas as classes, você pode criar um seletor de atributo.
- O seletor [atributo*="valor"] é usado para selecionar elementos cujo valor de atributo contém um valor específico, por exemplo:

```
[class*="col-"] {  
    float: left;  
    padding: 15px;  
}
```

- O exemplo acima adiciona a propriedade float e a propriedade padding à todos os elementos que possuírem o atributo class com o valor col-
- **Ou seja, com uma única declaração, todas as classes "col-x" (sendo x um número de 1 a 12) seriam afetadas.**

12 - Ajustando o fundo dos elementos



- **As propriedades background**

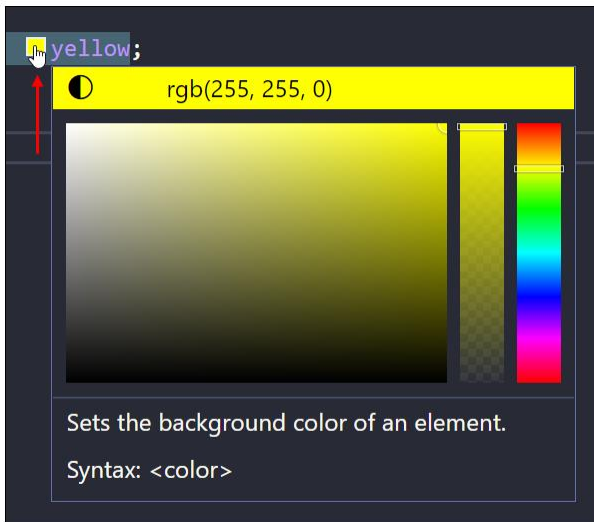
- As propriedades background são usadas para adicionar efeitos de plano de fundo para elementos. As propriedades disponíveis são:
 - background-color
 - background-image
 - background-repeat
 - background-attachment
 - background-position

- **A propriedade background-color**

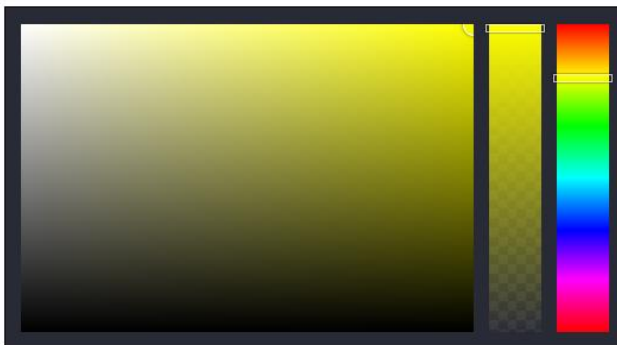
- A propriedade background-color permite definir uma cor para o plano de fundo de qualquer elemento HTML. Seu uso é muito simples, confira:

```
h1 {  
    background-color: ■green;  
}  
  
div {  
    background-color: ■lightblue;  
}  
  
p {  
    background-color: ■yellow;  
}
```

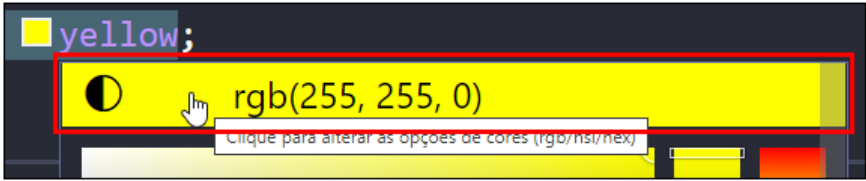

- Os valores da propriedade podem ser preenchidos de diferentes maneiras, confira as principais:
 - Nome de uma cor válida em inglês - como no exemplo apresentado acima;
 - Valor hexadecimal - como por exemplo: `#ff0000` (vermelho)
 - Valor RGB - como por exemplo: `rgb(255, 0, 0)`; (vermelho)
- O Visual Studio Code possui uma maneira de selecionar cores e de alternar entre os modos de preenchimento de uma forma bem simples, basta posicionar o mouse sobre o quadradinho que apresenta uma prévia da cor, e uma pequena janela seletora de cor será exibida



- Você pode selecionar o tom da cor através do seguinte local:



- Ou clicar na barra superior para alternar o modo de preenchimento da propriedade CSS



• A propriedade background-image

- A propriedade background-image especifica uma imagem a ser usada como plano de fundo de um elemento. Por padrão, a imagem é repetida para cobrir todo o elemento.
- Assim como a propriedade background-color, você pode utilizar a propriedade background-image para alterar o plano de fundo de qualquer elemento html.
- Para definir uma imagem de fundo para toda a página, utilize o seguinte:

```
body {  
    background-image: url("fundo-degrade.jpg");  
}
```

- **O valor da propriedade sempre deverá ser: url("local da imagem/nome da imagem")**
- Importante: Ao usar uma imagem de fundo, use uma imagem que não perturbe o texto.
- A imagem utilizada no exemplo acima possui 300x300px e a seguinte aparência:



- O resultado do CSS aplicado à página seria o seguinte:

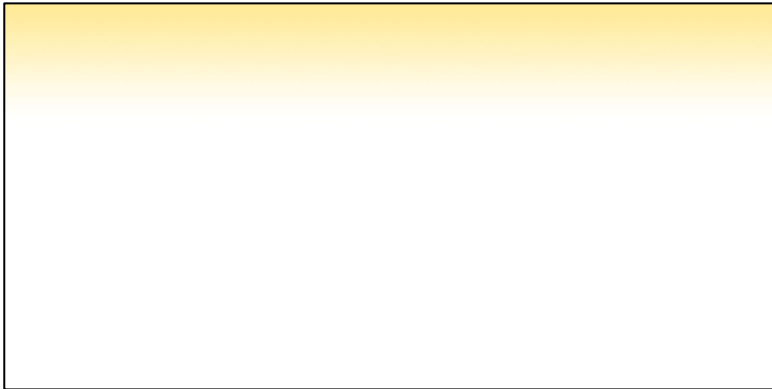


• A propriedade **background-repeat**

- Por padrão, a propriedade **background-image** repete uma imagem na horizontal e na vertical, conforme visto no exemplo anterior, isso pode resultar em um plano de fundo estranho. Se a imagem fosse repetida apenas na horizontal, o resultado seria melhor, e é pra isso que você pode utilizar a propriedade **background-repeat**. Os principais valores para esta propriedade são:
 - **repeat**: Este é o valor padrão, nele a imagem é repetida em ambas as direções.
 - **no-repeat**: Com este valor a imagem não será repetida, sendo aplicada apenas uma vez.
 - **repeat-x**: Com este valor a imagem será repetida na horizontal (eixo x)
 - **repeat-y**: Com este valor a imagem será repetida na vertical (eixo y)
- Confira o exemplo anterior com a aplicação do valor **repeat-x**:

```
body {  
    background-image: url("fundo-degrade.jpg");  
    background-repeat: repeat-x;  
}
```

- O resultado seria o seguinte:



- Junto da propriedade background-repeat, você pode utilizar a propriedade background-position;

- **A propriedade background-position**

- A propriedade background-position é usada para especificar a posição da imagem de fundo, ela é muito utilizada quando definimos que o background não se repetirá por toda a página, confira um exemplo:

```
body {  
    background-image: url("fundo-degrade.jpg");  
    background-repeat: no-repeat;  
    background-position: top center;  
}
```

- Com esse código, o resultado do exemplo anterior seria o seguinte:



- Os valores disponíveis para essa propriedade podem ser conferidos na imagem abaixo, cada valor em sua respectiva área de exibição:

left top

center top

right top

left center

center

right center

left bottom

center bottom

right bottom

• A propriedade **background-attachment**

- A propriedade **background-attachment** especifica se a imagem de fundo deve rolar ou ser fixa, não rolando com o resto da página. Essa propriedade possui dois valores:
 - **fixed**: Deixa a imagem fixada, ou seja, não acompanhando a rolagem da página.
 - **scroll**: Deixa a imagem "flutuando", acompanhando a rolagem da página.

13 - Ajustando o texto do site



- O CSS possui várias propriedades que permitem a formatação do texto em sua página, nesta lição você verá as seguintes:
 - color
 - text-align
 - text-decoration
 - text-transform
 - font-family
 - font-style
 - font-weight
 - font-size
- E também verá como utilizar o Google Fonts, uma biblioteca com mais de 800 fontes livres para uso em seu site.

• **A propriedade color**

- Como o nome indica, a propriedade color permite a modificação da cor do seu texto, o valor desta propriedade segue o mesmo padrão utilizado para a mudança dos fundos dos elementos, ou seja, você pode preenchê-la das seguintes maneiras:
 - Nome de uma cor válida em inglês - como por exemplo: red; (vermelho)
 - Valor hexadecimal - como por exemplo: #ff0000; (vermelho)
 - Valor RGB - como por exemplo: rgb(255, 0, 0); (vermelho)
- Você pode consultar diversos nomes de cores no seguinte site: <https://www.color-hex.com>
- Confira a aplicação da propriedade color no elemento body do html:

```
body{
    color: palevioletred;
}
```

- Desta forma, todos os textos presentes em seu site teriam a cor indicada, confira o código abaixo:

```
<!DOCTYPE html>
<head>
  <title>Teste de background</title>
  <style>
    body{
      color: palevioletred;
    }
  </style>
</head>
<body>
  <h1>Eu sou um H1, e terei a cor modificada!</h1>
  <p>Eu sou um P e também terei a cor modificada!</p>
  <ol>
    <li>Até eu,</li>
    <li>que sou uma lista,</li>
    <li>receberei as alterações.</li>
  </ol>
</body>
</html>
```


- O resultado seria o seguinte:


Eu sou um H1, e terei a cor modificada!

Eu sou um P e também terei a cor modificada!

1. Até eu,
2. que sou uma lista,
3. receberei as alterações.

- A aplicação da propriedade color pode ser feita de forma individual, para cada elemento, utilizando qualquer tipo de seletor. Confira a alteração de cor feita com diferentes seletores:

```
h1{
    color: palevioletred;
}

.paragrafo{
    color: cadetblue;
}

#lista{
    color: orangered;
}
```

- A aplicação no HTML seria feita da seguinte maneira:

```
<h1>Eu sou um H1, e terei a cor modificada!</h1>
<p class="paragrafo">Eu sou um P e também terei a cor modificada!</p>
<ol id="lista">
    <li>Até eu,</li>
    <li>que sou uma lista,</li>
    <li>receberei as alterações.</li>
</ol>
```

- O resultado seria o seguinte:


Eu sou um H1, e terei a cor modificada!

Eu sou um P e também terei a cor modificada!

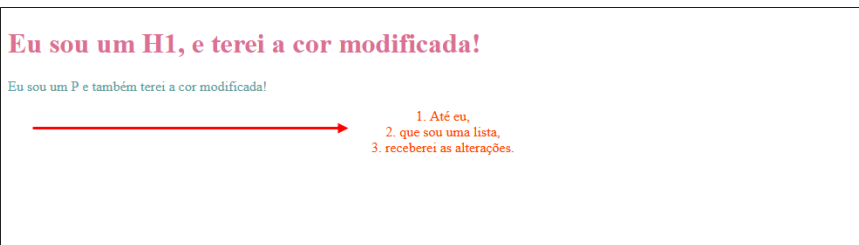
1. Até eu,
2. que sou uma lista,
3. receberei as alterações.

- **A propriedade text-align**

- A propriedade text-align permite a alteração do alinhamento horizontal do texto. Os valores podem ser:
 - **left** (padrão)
 - **right**
 - **centered**
 - **justified**
- O funcionamento do alinhamento de texto no HTML é idêntico ao do Word. Confira a aplicação do text-align com o valor center na lista do exemplo acima:

```
#lista{  
    color:  orangered;  
    text-align: center;  
}
```

- O resultado, seria o seguinte:



- Veja que o texto fica centralizado no espaço da página... Se o valor fosse alterado para right

- **A propriedade text-decoration**

- A propriedade text-decoration é uma forma prática de utilizar outras quatro propriedades, são elas:
 - text-decoration-line
 - text-decoration-color
 - text-decoration-style
 - text-decoration-thickness

- A propriedade **text-decoration-line** permite adicionar uma linha ao texto, seu valor pode ser:
 - **overline**: cria uma linha acima do texto
 - **line-through**: passa uma linha sobre o texto (tachado)
 - **underline**: cria uma linha abaixo do texto (sublinhado)
- Você também pode mesclar dois valores, confira o exemplo abaixo:

```
h1 {  
  text-decoration: overline;  
}  
  
h2 {  
  text-decoration: line-through;  
}  
  
h3 {  
  text-decoration: underline;  
}  
  
p.ex {  
  text-decoration: overline underline;  
}
```

```
<h1>H1 com overline</h1>  
<h2>H2 com line-through</h2>  
<h3>H3 com underline</h3>  
<p class="ex">P com combinação de overline e underline</p>
```

- O resultado dos códigos acima é o seguinte:

H1 com overline

H2 com line-through

H3 com underline

P com combinação de overline e underline

- A propriedade **text-decoration-color** permite definir a cor da linha criada, utilizando o padrão de uso de cores do CSS, confira:

```
h1 {  
  color: indianred;  
  text-decoration: overline underline;  
  → text-decoration-color: lightsalmon;  
}
```

<h1>Loja de ARTESANATOS da Carol</h1>

- O resultado seria o seguinte:

Loja de ARTESANATOS da Carol

- A propriedade **text-decoration-style** permite alterar o estilo da linha criada, seu valor pode ser:
 - **solid** (padrão)
 - **double**
 - **dotted**
 - **dashed**
 - **wavy**
- Confira a aplicação e o resultado do estilo "wavy" no exemplo anterior:

```
h1 {  
  color: indianred;  
  text-decoration: overline underline;  
  text-decoration-color: lightsalmon;  
  → text-decoration-style: wavy;  
}
```

Loja de ARTESANATOS da Carol

- Os demais estilos são os seguintes:

Loja de ARTESANATOS da Carol

text-decoration-style: solid;

Loja de ARTESANATOS da Carol

text-decoration-style: double;

Loja de ARTESANATOS da Carol

text-decoration-style: dotted;

Loja de ARTESANATOS da Carol

text-decoration-style: dashed;

- Por último, a propriedade **text-decoration-thickness** permite a definição da espessura do traçado, utilizando qualquer medida disponível no CSS, mas, comumente feita em pixels (px), confira o exemplo anterior com a definição da espessura:

```
h1 {
  color: indianred;
  text-decoration: overline underline;
  text-decoration-color: lightsalmon;
  text-decoration-style: wavy;
  → text-decoration-thickness: 1px;
}
```

- Veja que a propriedade text-decoration-thickness fica com a cor branca, diferente das demais, pois é uma propriedade que funciona, porém, está caindo em desuso.
- O resultado seria o seguinte:

Loja de ARTESANATOS da Carol

- Se o valor fosse modificado para 4x, o resultado seria o seguinte:

Loja de ARTESANATOS da Carol

- Conforme explicado anteriormente, a propriedade `text-decoration` permite a junção das quatro propriedades que acabamos de ver, a aplicação seria a seguinte:

```
h1 {  
  color: indianred;  
  text-decoration: overline underline lightsalmon wavy 4px;  
}
```

- Essa única linha surtiria o mesmo efeito que as quatro linhas feitas anteriormente...

• A propriedade `text-transform`

- A propriedade `text-transform` é usada para especificar letras maiúsculas e minúsculas em um texto. Com ela podemos forçar que as letras sejam exibidas conforme desejado, independentemente da forma que elas forem inseridas no código HTML. Os valores podem ser:
 - **uppercase**: deixa todas as letras do texto maiúsculas;
 - **lowercase**: deixa todas as letras do texto minúsculas;
 - **capitalize**: deixa o texto capitalizado, ou seja, com a primeira letra de cada palavra no formato maiúsculo;

• A propriedade `font-family`

- A propriedade `font-family` permite a seleção de uma fonte para o texto de sua página. É importante ao utilizar a `font-family` que você determine mais de um nome de fonte para a sua página, isso cria um "sistema reserva", caso uma fonte não funcione ou não exista no computador do usuário, outra fonte será utilizada.
- Além do nome da fonte, o CSS permite o uso de "famílias genéricas" de fonte, que são determinadas apenas pelo "tipo" da fonte, as cinco famílias disponíveis no CSS são:
 1. **Serif**: As fontes com serifa têm um pequeno traço nas bordas de cada letra. Eles criam uma sensação de formalidade e elegância.
 2. **Sans-serif**: As fontes sem serifa têm linhas limpas (sem pequenos traços anexados). Eles criam um visual moderno e minimalista.

3. **Monospace:** Fontes mono espaçadas - aqui todas as letras têm a mesma largura fixa. Eles criam uma aparência mecânica.
 4. **Cursive:** As fontes cursivas imitam a caligrafia humana
 5. **Fantasy:** As fontes do tipo "fantasia" são fontes decorativas/lúdicas.
- Confira exemplos de fonte de cada um dos tipos apresentados acima:

Serif	Bodoni MT Times New Roman
Sans-serif	Arial Lucida Sans
Monospace	Courier New Lucida Console
Cursive	<i>Brush Script MT</i> <i>Freestyle Script</i>
Fantasy	Cooper Black Papyrus

- Para o preenchimento da propriedade, se o nome da fonte for mais de uma palavra, deve estar entre aspas, como: "Times New Roman".
- O uso da propriedade é feito da seguinte maneira:

```
h1 {
    font-family: Arial, Helvetica, sans-serif;
}
```

- Veja que no exemplo acima, foram determinadas duas fontes (arial e helvetica), além de determinar a família genérica (sans-serif), isso garantiria que, caso as fontes arial e helvetica não estejam disponíveis, a página ainda utilizaria uma fonte sem serifa.

- Existem algumas fontes "seguras" para o uso em sua página, confira as principais abaixo:
 - Arial (sans-serif)
 - Verdana (sans-serif)
 - Tahoma (sans-serif)
 - Trebuchet MS (sans-serif)
 - Times New Roman (serif)
 - Georgia (serif)
 - Garamond (serif)
 - Courier New (monospace)
 - Brush Script MT (cursive)
- Antes de finalizar seu site e publicá-lo, verifique como as fontes aparecem em diferentes navegadores e dispositivos. Além disso, lembre-se sempre de utilizar fontes alternativas, mantendo um sistema de segurança.

• A propriedade font-style

- A propriedade font-style é comumente utilizada para deixar um texto em itálico. Ela pode possuir três valores:
 - **normal**: deixa a exibição do texto em seu estado normal
 - **italic**: deixa o texto em itálico
 - **oblique**: deixa o texto inclinado, bem parecido com o itálico, porém, com um suporte de navegadores inferior

• A propriedade font-weight

- A propriedade font-weight define o "peso" da sua fonte, normalmente utilizada para deixar o texto em negrito, além de alguns valores numéricos, ela pode conter os seguintes valores:
 - **normal**: deixa o texto com seu peso normal
 - **bold**: deixa o texto em negrito

• A propriedade font-size

- A propriedade font-size permite a definição do tamanho utilizado na fonte do seu site.

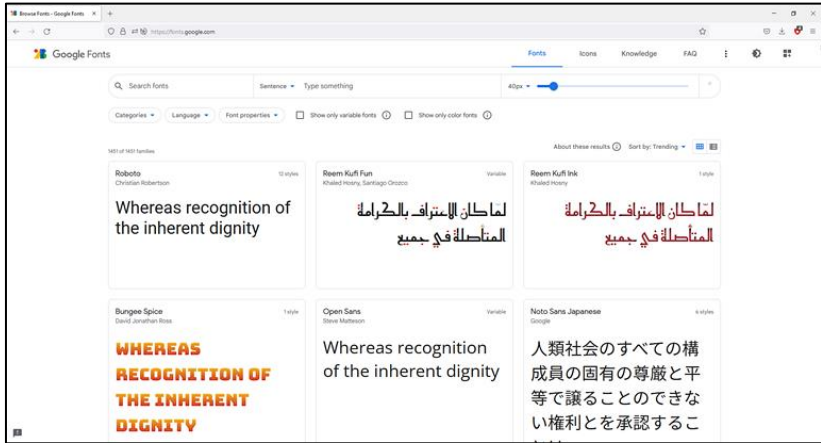
- É importante ressaltar que, essa propriedade não deve ser utilizada para a definição de títulos em sua página, para isso você deve utilizar os elementos h1 - h6 do HTML!
- O valor do tamanho da fonte pode ser um tamanho absoluto ou relativo.
- **Tamanho absoluto:**
 - Define o texto para um tamanho especificado
 - Não permite que um usuário altere o tamanho do texto em todos os navegadores (ruim por motivos de acessibilidade)
 - O tamanho absoluto é útil quando o tamanho físico da saída é conhecido
- **Tamanho relativo:**
 - Define o tamanho em relação aos elementos circundantes
 - Permite que um usuário altere o tamanho do texto nos navegadores
- Se você não especificar um tamanho de fonte, o tamanho padrão para texto normal, como parágrafos, é 16px (16px=1em)
- A propriedade é usada de forma simples, para definir o tamanho absoluto, você pode utilizar o valor em pixels, confira abaixo:

```
p {  
    font-size: 40px;  
}
```

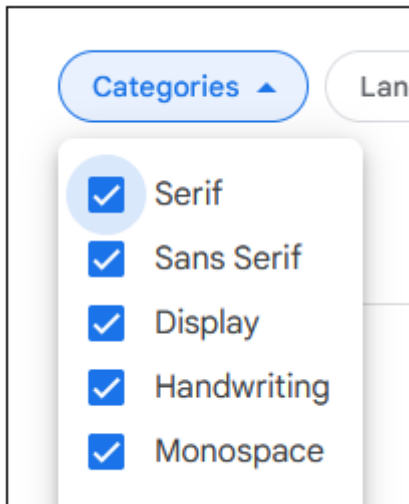
- Para a definição em tamanho relativo, você pode utilizar a medida **vw**, ela usa como base a largura da viewport (janela de exibição), a ideia para o uso dessa medida é a seguinte:
- **1vw = 1% da largura da janela de visualização. Se a janela de visualização tiver 50 cm de largura, 1vw é 0,5 cm. Ou seja, 10vw definirá o tamanho para 10% da largura da janela de visualização.**

- **Google Fonts**

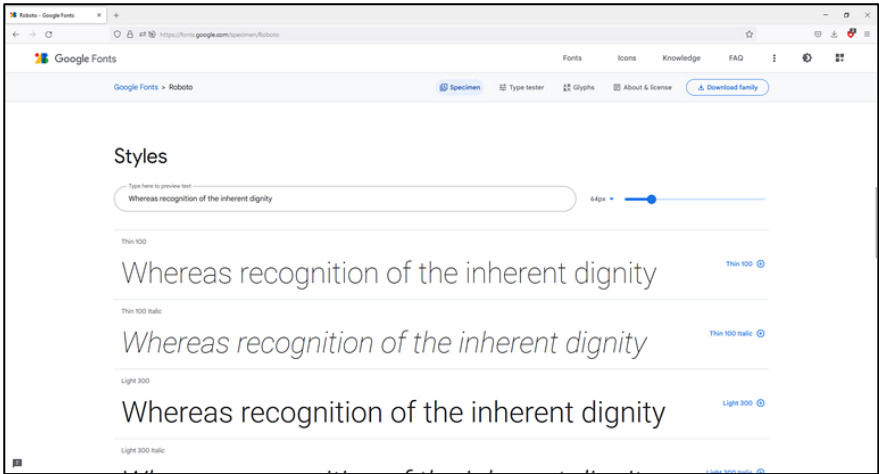
- Se você não desejar utilizar nenhuma das fontes padrão em HTML, poderá usar o Google Fonts.
- O uso é simples, inicialmente você precisa acessar o site do google fonts em: <http://fonts.google.com>
- Na página inicial você verá algumas sugestões de fontes, além de uma caixa de pesquisa



- Na caixa "Categories" você pode selecionar o tipo de fonte que visualizará



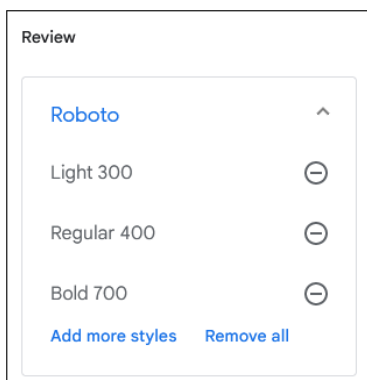
- Após localizar a fonte que deseja utilizar em seu site, clique sobre ela para visualizar uma página com os detalhes da fonte. Na página de detalhes, role até visualizar os estilos disponíveis para a fonte



- Na parte direita dos estilos, você possui um botão responsável por adicionar o estilo desejado ao código que será gerado para uso em seu site



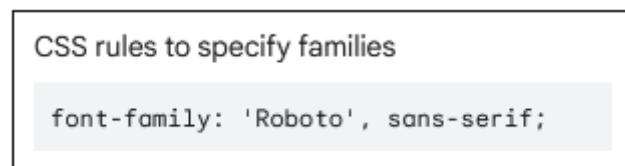
- Após clicar sobre os estilos desejados, um painel será exibido na parte direita da página, ele exibirá um review com os estilos selecionados:



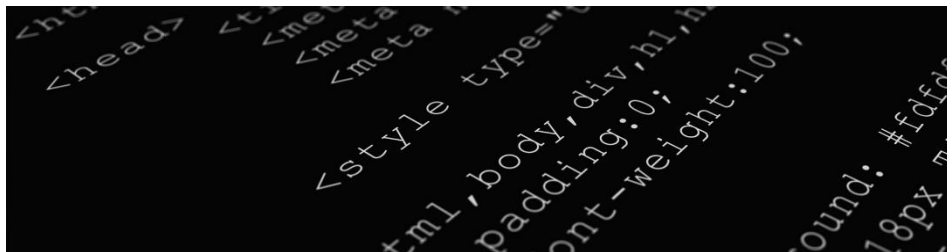
- E o elemento link que deverá ser incluído dentro do elemento head de seu site:



- Além disso, o site também apresenta como deve ser o uso da propriedade CSS com a fonte desejada:



14 - Utilizando ícones no site



- Nos sites atuais é muito comum o uso de ícones para auxiliar a localização de recursos, para ilustrar links entre outras aplicações possíveis...

• **O Font Awesome**

- Para a criação destes links iremos usar um serviço chamado Font Awesome. Ele é acessível através do seguinte link: <https://fontawesome.com>
- Como o próprio site deles diz:
- A maneira mais fácil de obter ícones em seu site é com um Kit. O Font Awesome permite a criação de seu próprio kit personalizado, tudo empacotado apenas com os ícones, ferramentas e configurações de que você precisa.
- A ideia é muito simples, você acessa o site, cria sua conta, cria o seu kit, insere o código no elemento <head> da sua página e pronto, basta inserir os ícones em sua página...
- O font awesome é um serviço pago, que tem a possibilidade de uso em um plano gratuito. Esse plano possui algumas limitações, porém, essas limitações não impedem sua utilização em um site de pequeno porte.
- As principais diferenças ficam por conta da quantidade de ícones e de estilos disponíveis, além da quantidade de visualizações disponíveis para a sua página (10.000 por mês). Para maiores informações, confira o seguinte link: <https://fontawesome.com/plans>

• A criação da conta e do kit

- Ao acessar o site, procure pelo link "Start" ou pelo botão "Start for free".
- Você será redirecionado para uma caixa "Enter your email to get started with a free Kit!"
- Preencha com um email válido, que nunca tenha sido utilizado no site do Font Awesome, e clique no botão "Send Kit Code"



Enter your email to get started with a free Kit! ?

e.g. violet@freshsqueezed.org

Send Kit Code 

 Powered by Cloudflare

Already have an account? Sign In →

- Você receberá um email para confirmação e finalização da criação da conta
- Após finalizar a criação da conta, você será redirecionado para a tela com o código responsável pelo seu kit



1 Add Your Kit's Code to a Project

```
<script src="https://kit.fontawesome.com/...js" crossorigin="anonymous"></script>
```

Copy your kit's code into the `<head>` of each template or page in your project where you want to use Font Awesome via this kit. If you've got other versions or flavors of Font Awesome already in use, make sure to remove them, mmkay?

 Download Example File

- Agora, basta copiar o elemento `<script>` apresentado e inseri-lo dentro do `<head>`

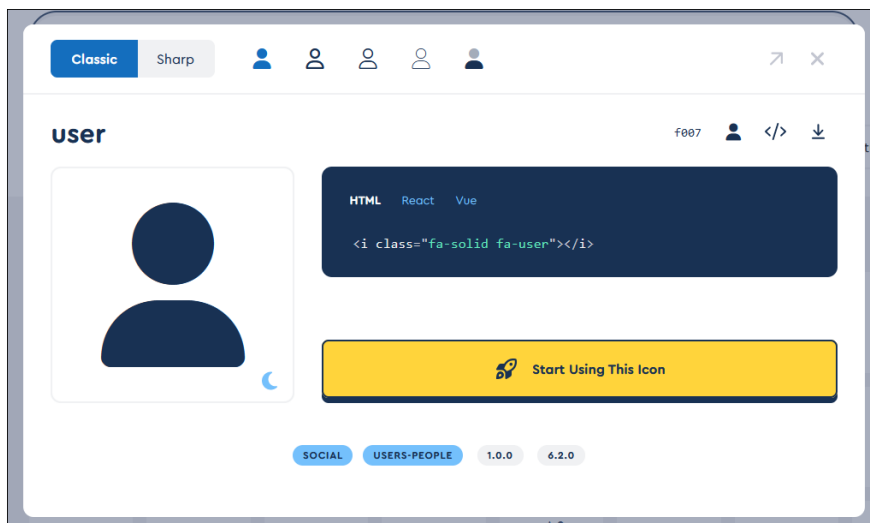
```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Pousada Paradiso</title>
  <script src="https://kit.fontawesome.com/...js" crossorigin="anonymous"></script>
  <link rel="stylesheet" href="css/estilo.css">
</head>
```

• Inserindo os ícones em sua página

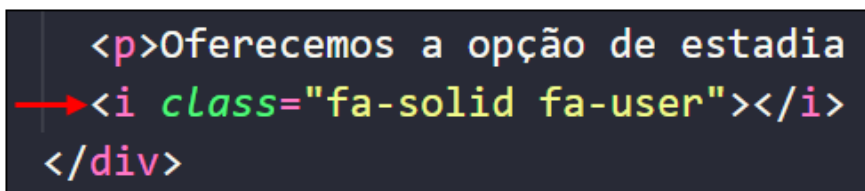
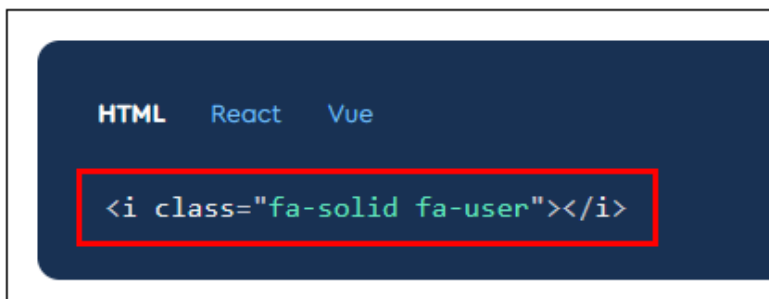
- Após a inclusão do elemento `<script>` você pode navegar pela galeria de ícones disponível no seguinte link:

<https://fontawesome.com/search?o=r&m=free>

- Ao localizar o ícone que deseja utilizar, basta clicar sobre ele para visualizar mais informações



- Copie o código HTML apresentado na caixa e cole-o no local onde deseja utilizá-lo em sua página




- **Alterando o tamanho dos ícones**

- Após criar o ícone em sua página, você pode utilizar algumas classes criadas pelo próprio Font Awesome, basta adicioná-la ao elemento `<i>` responsável pelo ícone. Confira as classes abaixo:

Classe	Tamanho em fonte	Tamanho em pixels
fa-2xs	0.625em	10px
fa-xs	0.75em	12px
fa-sm	0.875em	14px
fa-lg	1.25em	20px
fa-xl	1.5em	24px
fa-2xl	2em	32px
fa-2xs	0.625em	10px

- Confira o exemplo de aplicação:

```
<p>Oferecemos a opção de estadia em quartos  
<i class="fa-solid fa-user fa-2xl"></i>  
</div>
```



15 - Personalizando os links com o uso do CSS



- Com o CSS você pode personalizar seus links de diversas maneiras, com qualquer propriedade, ou seja, você pode mudar a cor do texto, a fonte utilizada, o preenchimento do link, etc.
- A ideia básica é modificar o elemento `<a>` com o uso do CSS, por exemplo:

```
a{  
  color: coral;  
}
```

- O código acima resultaria na alteração da cor do texto de todos os links de teu site:

```
<a href="index.html">HOME</a> |  
<a href="contato.html">FALE CONOSCO</a> |  
<a href="login.html">ACESSAR</a>
```

- O resultado seria o seguinte:

[HOME](#) | [FALE CONOSCO](#) | [ACESSAR](#)

- O CSS também pode alterar os diferentes estados do seu link. Por padrão, um link possui quatro estados, são eles:
 - a:link - um link normal e não visitado
 - a:visited - um link que o usuário visitou
 - a:hover - um link quando o usuário passa o mouse sobre ele
 - a:active - um link no momento em que é clicado
- É muito comum deixar os estados "link" e "visited" com a mesma configuração, isso garante que seus links sempre terão a mesma aparência, tendo sido visitados ou não. Confiar a modificação feita nos links do exemplo anterior:

```
<a href="index.html">HOME</a>  
<a href="contato.html">FALE CONOSCO</a>  
<a href="login.html">ACESSAR</a>
```

- Inicialmente, removemos os caracteres que estavam fazendo a separação do texto e incluímos novas propriedades para a modificação dos links em seu estado inicial e quando já tiverem sido visitados

```
a:link, a:visited{  
    background-color: ■ coral;  
    color: ■ white;  
    padding: 10px 25px;  
    text-align: center;  
    text-decoration: none;  
    display: inline-block;  
    border-bottom: 2px solid ■ chocolate;  
}
```

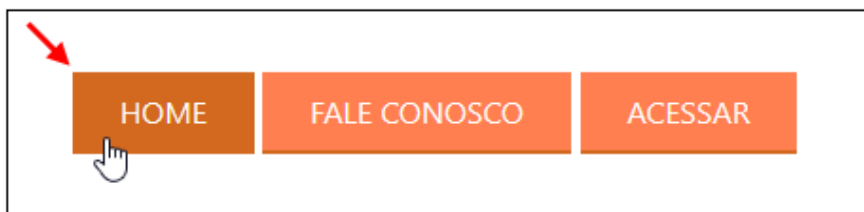
- Essas alterações, deixam os links com a aparência de um botão, confira o resultado:



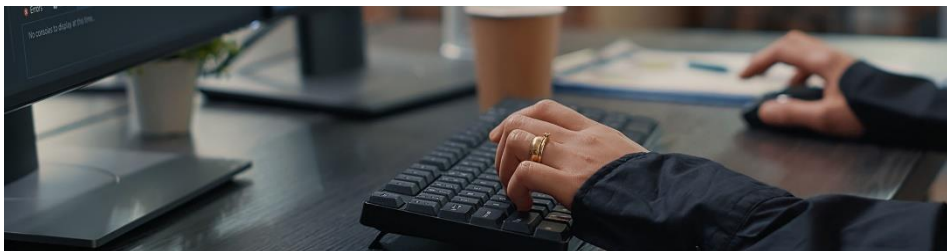
- Adicionando a seguinte propriedade CSS:

```
a:hover, a:active{  
    background-color: #8B4513; chocolate;  
}
```

-
- O estado de "hover" e "active" seriam alterados, quando o usuário posicionar o mouse sobre ou clicar no link, o fundo teria a cor modificada, resultando no seguinte:



16 - Ajustando listas e tabelas com o uso do CSS



- Como visto durante a parte em HTML, existem dois tipos de listas, as ordenadas (ol) e as não ordenadas (ul); Com o uso do CSS podemos modificar as seguintes propriedades das listas:
 - Definir diferentes marcadores para as listas não ordenadas
 - Definir diferentes estilos de números ou letras para as listas ordenadas
 - Definir uma imagem como marcador
 - Adicionar um fundo para ambas as listas

• A propriedade **list-style-type**

- A propriedade **list-style-type** permite definir o estilo do marcador ou da numeração utilizada em sua lista, ela pode conter diferentes valores para cada tipo de lista, confira os principais abaixo:
- **Listas não ordenadas:**
 - **none**: para remover os marcadores
 - **disc**: valor padrão, marcadores circulares pretos
 - **circle**: marcadores circulares vazados (sem preenchimento)
 - **square**: marcadores quadrados
- **Listas ordenadas:**
 - **none**: para remover os marcadores
 - **decimal**: valor padrão, lista numerada
 - **decimal-leading-zero**: semelhante ao valor padrão, porém, com os números iniciando com 0.
 - **lower-roman**: números romanos em letras minúsculas
 - **upper-roman**: números romanos em letras maiúsculas

- **lower-alpha:** lista feitas com letras (ordem alfabética) minúsculas
- **upper-alpha:** lista feitas com letras (ordem alfabética) maiúsculas
- Você também pode alterar os marcadores com o uso de strings, ou seja, definir um caractere específico ou até mesmo uma palavra para sua lista, confira o exemplo abaixo:

```
ul{  
  list-style-type:"Cpt. ";  
}
```

- Com o código CSS acima, todas as listas do tipo ul em seu site seriam apresentadas com a string "Cpt. " como marcador, confira o resultado:

Cpt. Carlos Eduardo
Cpt. Roberto D'Avilla
Cpt. Wilson Simões

- Você também pode utilizar o modelo de string para o uso de emojis em sua lista, por exemplo:

```
ul{  
  list-style-type:"✅";  
}
```

- O resultado seria o seguinte:

✅ Carlos Eduardo
✅ Roberto D'Avilla
✅ Wilson Simões

- Confira o seguinte código CSS:

```
ul.informatica{  
    list-style-type: "💻";  
}  
  
ul.idiomas{  
    list-style-type: "📄";  
}
```

- Com o código acima, definimos dois marcadores diferentes, um para cada lista ul com a determinada classe, ou seja, confira o seguinte código HTML:

```
<h2>Lista de cursos da nossa escola</h2>  
<ul>  
    <li>Inglês</li>  
    <li>Espanhol</li>  
    <li>Alemão</li>  
</ul>  
<ul>  
    <li>Windows</li>  
    <li>Word</li>  
    <li>Excel</li>  
</ul>
```

- Da forma apresentada, o resultado seria o seguinte:

Lista de cursos da nossa escola

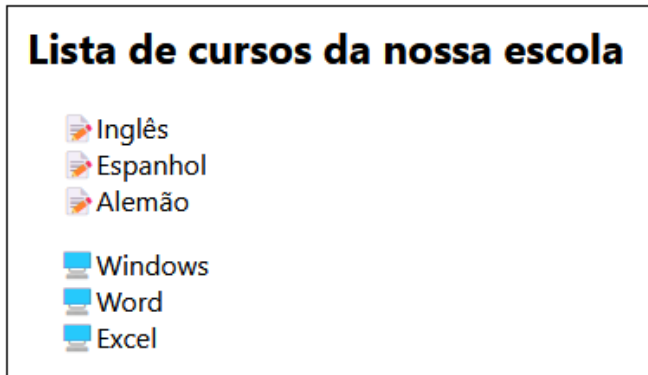
- Inglês
- Espanhol
- Alemão

- Windows
- Word
- Excel

- Acrescentando as classes em ambos os ul, conforme exemplificado abaixo:

```
<h2>Lista de cursos da nossa escola</h2>
<ul class="idiomas">
  <li>Inglês</li>
  <li>Espanhol</li>
  <li>Alemão</li>
</ul>
<ul class="informatica">
  <li>Windows</li>
  <li>Word</li>
  <li>Excel</li>
</ul>
```

- O resultado seria o seguinte:



• A propriedade **list-style-image**

- A propriedade `list-style-image` permite o uso de uma imagem como marcador para a lista, o uso é bem simples e segue o mesmo padrão visto para as imagens de fundo no CSS, confira:

```
ul{  
  list-style-image:url("imagem/marcador.png");  
}
```

- O valor da propriedade sempre será:
`url("caminho-e-imagem.extensão")`
- Confira a imagem e o resultado da lista:



- **A propriedade background**

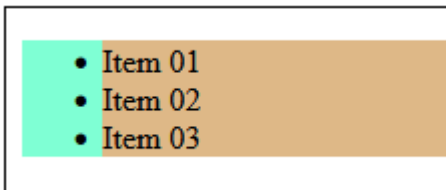
- Você pode utilizar a propriedade background para modificar o fundo da lista, neste caso, é possível alterar a cor de toda a lista, utilizando a propriedade no elemento ul, assim como alterar o fundo apenas dos itens, utilizando a propriedade no elemento li, confira abaixo:

```
ul{  
    background-color: ■ aquamarine;  
}  
li{  
    background-color: ■ burlywood;  
}
```

- O código CSS acima, com o código HTML abaixo...

```
<ul>  
    <li>Item 01</li>  
    <li>Item 02</li>  
    <li>Item 03</li>  
</ul>
```

- Resultaria na seguinte lista:



- O elemento ul, responsável por toda a área da lista, fica "atrás" do elemento li, por isso o preenchimento com a cor "aquamarine" fica visível apenas na área dos marcadores...

• O uso de listas na criação de menus

- Uma prática comum durante a criação de menus horizontais é a criação através do uso de listas, para isso você deve efetuar algumas modificações nos elementos ul e li, vamos conferir o código:

```
*{  
    box-sizing: border-box;  
}  
body{  
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
}
```

- Inicialmente definimos a propriedade box-sizing para todos os elementos HTML e a fonte que utilizaremos na nossa página...
- Agora, temos a nossa lista criada com algumas classes incorporadas em alguns elementos... A lista (ul) utilizará a classe menu e o último item (li) utilizará a classe destaque

```
<ul class="menu">  
    <li><a href="index.html">HOME</a></li>  
    <li><a href="servicos.html">SERVIÇOS</a></li>  
    <li><a href="contato.html">FALE CONOSCO</a></li>  
    <li class="destaque"><a href="login.html">ACESSAR</a></li>  
</ul>
```


- Por enquanto, o resultado seria o seguinte:



- Vamos iniciar a definição das propriedades para os elementos... Confira a criação de um seletor que afetará apenas o elemento ul que possuir a classe "menu" declarada

```
ul.menu{
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: auto;
    background-color: coral;
    border-bottom: 2px solid chocolate;
}
```

- Com o código acima, você remove a utilização de marcadores na lista, define que ela não possuirá margens e preenchimento (margin e padding), garante que o conteúdo será exibido dentro da área padrão da lista com a propriedade overflow, sem a necessidade de especificação de tamanhos. O código também define uma cor de fundo e uma borda inferior para o elemento ul, o resultado seria o seguinte:



- Agora, podemos definir a flutuação dos elementos li, eles precisam ser exibidos lado a lado, flutuando sempre à esquerda, para isso, basta utilizar a propriedade float, confira:

```
ul.menu li {
    float: left;
}
```

- O código acima afeta todos os elementos do tipo li que estão dentro do elemento ul que utilizar a classe menu. O resultado seria o seguinte:



- Agora, podemos iniciar a modificação dos links, conforme vimos anteriormente:

```
ul.menu li a:Link, a:visited{
    color: white;
    padding: 10px 25px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
}
```

- O código acima altera todos os links em seu estado inicial e no estado visitado, deixando o texto branco, adicionando o espaço de preenchimento (padding) e garantindo que os elementos sejam exibidos no formato de caixa em linha (inline-block), o resultado seria o seguinte:



HOME SERVIÇOS FALE CONOSCO ACESSAR

- Agora podemos modificar os links em seus estados de cursor sobre o link (hover) e quando estiverem clicados (active), a alteração pode ser feita apenas para o background:

```
ul.menu li a:hover, a:active{
    background-color: chocolate;
}
```

- Seguindo a mesma ideia, podemos alterar o último li (com a classe destaque), deixando-o flutuando à direita e com outra cor de fundo:

```
ul.menu li.destaque{
    float:right;
    background-color: brown;
    font-weight: bold;
}
```

- O resultado final seria o seguinte:



- Você pode conferir o código utilizado abaixo:

CSS:

```
*{
    box-sizing: border-box;
}
body{
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

ul.menu{
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: auto;
    background-color: coral;
    border-bottom: 2px solid chocolate;
}

ul.menu li {
    float: left;
}

ul.menu li a:link, a:visited{
    color: white;
    padding: 10px 25px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
}

ul.menu li a:hover, a:active{
    background-color: chocolate;
}

ul.menu li.destaque{
    float:right;
    background-color: brown;
    font-weight: bold;
}
```

HTML:

```
<!DOCTYPE html>
<html lang="pr-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>Exemplo de menu</title>
</head>
<body>
  <ul class="menu">
    <li><a href="index.html">HOME</a></li>
    <li><a href="servicos.html">SERVIÇOS</a></li>
    <li><a href="contato.html">FALE CONOSCO</a></li>
    <li class="destaque"><a href="login.html">ACESSAR</a></li>
  </ul>
</body>
</html>
```

• Ajustando tabelas com o uso do CSS

- Assim como visto para as listas, as tabelas HTML podem ser modificadas com o uso do CSS, as alterações para a tabela podem ser aplicadas em todos os elementos que compõem a tabela, ou seja:
 - table
 - th
 - tr
 - td
- Esses elementos podem ser afetados por diferentes propriedades CSS, podendo-se modificar a borda (border), o fundo (background), o alinhamento dos textos (text-align), a altura das linhas (height), entre outras opções...

Agora você irá conferir alguns exemplos de tabelas modificadas com o uso do CSS, para todos os exemplos, utilizaremos a mesma tabela:

```
<table>
  <tr>
    <th>Nome</th>
    <th>Sobrenome</th>
    <th>Cargo</th>
  </tr>
  <tr>
    <td>Carlos</td>
    <td>Roberto</td>
    <td>Técnico em Informática</td>
  </tr>
  <tr>
    <td>Luana</td>
    <td>Mendes</td>
    <td>Técnica em Informática</td>
  </tr>
  <tr>
    <td>Silvio</td>
    <td>Tadeu</td>
    <td>Analista de Sistemas</td>
  </tr>
  <tr>
    <td>Roberta</td>
    <td>Miura</td>
    <td>Analista de Sistemas</td>
  </tr>
</table>
```

O resultado desta tabela, sem CSS é o seguinte:

Nome	Sobrenome	Cargo
Carlos	Roberto	Técnico em Informática
Luana	Mendes	Técnica em Informática
Silvio	Tadeu	Analista de Sistemas
Roberta	Miura	Analista de Sistemas

- **Exemplo 01 :: Tabela com borda simples**

- Tabela ocupando toda a largura da tela, com bordas duplas e alinhamento do texto à esquerda:

```
table, th, td {  
    border: 1px solid;  
    text-align: left;  
}  
  
table{  
    width: 100%;  
}
```

- O primeiro grupo de seletores (table, th, td) aplica a borda e define o alinhamento dos elementos.
- O segundo seletor (table) define que a tabela ocupará 100% da largura da tela (width)
- O resultado é o seguinte:

Nome	Sobrenome	Cargo
Carlos	Roberto	Técnico em Informática
Luana	Mendes	Técnica em Informática
Silvio	Tadeu	Analista de Sistemas
Roberta	Miura	Analista de Sistemas

- Para deixar a borda simples, ou seja, com um único risco, bastaria adicionar a propriedade border-collapse, confira:

```
table, th, td {
  → border-collapse: collapse;
    border: 1px solid;
    text-align: left;
}
```

O resultado:

Nome	Sobrenome	Cargo
Carlos	Roberto	Técnico em Informática
Luana	Mendes	Técnica em Informática
Silvio	Tadeu	Analista de Sistemas
Roberta	Miura	Analista de Sistemas

• Exemplo 02 :: Tabela com bordas diferentes

- Tabela com tamanho definido através do conteúdo, com borda externa sólida em uma cor e bordas internas tracejadas em cor diferente:

```
table{
  border-collapse: collapse;
  border: 2px solid ■ #666;
}
th, td {
  padding: 10px;
  text-align: left;
  border: 1px dashed ■ #666;
}
```


- O resultado seria o seguinte:

Nome	Sobrenome	Cargo
Carlos	Roberto	Técnico em Informática
Luana	Mendes	Técnica em Informática
Silvio	Tadeu	Analista de Sistemas
Roberta	Miura	Analista de Sistemas

• Exemplo 03 :: Tabela com borda inferior

- Tabela ocupando toda a largura da página e com bordas inferiores:

```
table{
    width: 100%;
    border-collapse: collapse;
}
th, td {
    padding: 10px;
    text-align: left;
    border-bottom: 1px solid #666;
}
```

- O resultado:

Nome	Sobrenome	Cargo
Carlos	Roberto	Técnico em Informática
Luana	Mendes	Técnica em Informática
Silvio	Tadeu	Analista de Sistemas
Roberta	Miura	Analista de Sistemas

- Para adicionar um efeito ao passar o mouse sobre as linhas, bastaria adicionar um seletor para os tr no estado hover, confira:

```
tr:hover{
    background-color: khaki;
}
```

- O resultado:

Nome	Sobrenome	Cargo
Carlos	Roberto	Técnico em Informática
Luana	Mendes	Técnica em Informática
Silvio	Tadeu	Analista de Sistemas
Roberta	Miura	Analista de Sistemas

Exemplo 04 :: Tabela com efeito zebrado

- Tabela sem bordas e com diferenciação de cores à cada linha (efeito zebrado):

```
table {
    border-collapse: collapse;
    width: 100%;
}
th, td {
    text-align: left;
    padding: 8px;
}
tr:nth-child(even) {
    background-color: #d8f5ff;
}
```

- Para a mudança de cor de forma alternada nas linhas, ou seja, uma linha colorida e outra não, utilizamos o seletor **:nth-child()**, ele pode possuir dois valores:
 - :nth-child(**odd**) = Seleção dos elementos ímpares, ou seja, a primeira linha, terceira linha, quinta linha e assim sucessivamente
 - :nth-child(**even**) = Seleção dos elementos pares, ou seja, a segunda linha, quarta linha, sexta linha e assim sucessivamente
- O código acima resultaria na seguinte tabela:

Nome	Sobrenome	Cargo
Carlos	Roberto	Técnico em Informática
Luana	Mendes	Técnica em Informática
Silvio	Tadeu	Analista de Sistemas
Roberta	Miura	Analista de Sistemas

- Se alterássemos o seletor para **odd**, o resultado seria o seguinte:

Nome	Sobrenome	Cargo
Carlos	Roberto	Técnico em Informática
Luana	Mendes	Técnica em Informática
Silvio	Tadeu	Analista de Sistemas
Roberta	Miura	Analista de Sistemas

- **Exemplo 05 :: Tabela estilosa**

- Uma tabela com uma combinação de todos os conceitos vistos anteriormente:

```
body{
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
table {
    border-collapse: collapse;
    width: 100%;
}
th, td {
    text-align: left;
    padding: 10px;
    border: 1px solid #cacaca;
}
tr:nth-child(even) {
    background-color: #ecec;
}
tr:hover{
    background-color: #c0dced;
}
th{
    background-color: #0489aa;
    color: #fff;
    padding: 12px;
    text-transform: uppercase;
}
```

- O resultado seria o seguinte:

NOME	SOBRENOME	CARGO
Carlos	Roberto	Técnico em Informática
Luana	Mendes	Técnica em Informática
Silvio	Tadeu	Analista de Sistemas
Roberta	Miura	Analista de Sistemas

17 - Ajustando formulários com o uso do CSS



- Assim como feito com as listas e com as tabelas, os formulários podem ser modificados com o uso do CSS afetando seus principais elementos, labels, inputs, textboxes entre outros. Para esses elementos, podemos utilizar o CSS para modificar diferentes propriedades, alterando as bordas, o preenchimento, o fundo, e diferentes estados.
- Para esta lição, vamos seguir o mesmo modelo visto na lição anterior, vamos utilizar um formulário e trabalhar com diferentes opções para ele, confira o código abaixo:

```
<form action="">
  <label for="nome">Nome</label>
  <input type="text" id="nome" name="nomecompleto" placeholder="Seu nome completo...">

  <label for="email">Email</label>
  <input type="email" id="email" name="email" placeholder="Seu email...">

  <p>Selecione o curso que deseja?</p>
  <input type="checkbox" id="excel" name="excel" value="excel">
  <label for="excel"> Excel 365</label><br>
  <input type="checkbox" id="excel-av" name="excel-av" value="excel-av">
  <label for="excel-av"> Excel 365 - Avançado</label><br>
  <input type="checkbox" id="excel-dash" name="excel-dash" value="excel-dash">
  <label for="excel-dash"> Excel 365 - Dashboard</label><br>
  <input type="checkbox" id="excel-tabela" name="excel-tabela" value="excel-tabela">
  <label for="excel-tabela"> Excel 365 - Tabelas Dinâmicas</label><br>

  <label for="periodo">Período</label>
  <select id="periodo" name="periodo">
    <option value="manha">Manhã</option>
    <option value="tarde">Tarde</option>
    <option value="noite">Noite</option>
  </select>

  <p>Você tem experiência com Excel?</p>
  <input type="radio" id="exp-s" name="experiencia" value="S">
  <label for="exp-s">Sim</label><br>
  <input type="radio" id="exp-n" name="experiencia" value="N">
  <label for="exp-n">Não</label>

  <input type="submit" value="Enviar">
</form>
```

- Com um CSS modificando apenas a fonte utilizada na página...

```
body{  
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
}
```

- O resultado seria o seguinte:

Nome Email

Selecione o curso que deseja:

- ☐ Excel 365
- ☐ Excel 365 - Avançado
- ☐ Excel 365 - Dashboard
- ☐ Excel 365 - Tabelas Dinâmicas

Período

Você tem experiência com Excel?

- ☐ Sim
- ☐ Não

- Confira que as quebras de linhas manuais (br) foram criadas apenas na lista de checkboxes (curso desejado) e nos radios (experiência com excel).

• Alterando os inputs

- Você pode alterar todos os inputs utilizando o seletor sem especificação de tipo, confira abaixo:

```
input{  
  width: 100%;  
}
```

- Desta forma, todos os inputs ocupariam 100% da largura da tela, o resultado no formulário anterior seria o seguinte:

Nome
Seu nome completo...

Email
Seu email...

Selecione o curso que deseja:

Excel 365 ☐

Excel 365 - Avançado ☐

Excel 365 - Dashboard ☐

Excel 365 - Tabelas Dinâmicas ☐

Período Manhã ▾

Você tem experiência com Excel?

Sim ☐

Não ☐

Enviar

- Veja que os checkboxes e os radios também foram afetados, para evitar isso, deve-se utilizar a opção de seletor de atributos, confira a alteração do código CSS:

```
input[type=text], input[type=email], select{
  width: 100%;
}
```

- Esse seria o resultado:

Nome
Seu nome completo...

Email
Seu email...

Selecione o curso que deseja:

☐ Excel 365

☐ Excel 365 - Avançado

☐ Excel 365 - Dashboard

☐ Excel 365 - Tabelas Dinâmicas

Período Manhã ▾

Você tem experiência com Excel?

☐ Sim

☐ Não Enviar

- Veja que apenas os inputs do tipo "text" e "email" e o select foram afetados pelo código CSS.
- Confira mais uma alteração feita no CSS responsável pelos elementos alterados anteriormente:

```
input[type=text], input[type=email], select{
  width: 100%;
  border: none;
  border-bottom: 2px solid gray;
  padding: 12px 20px;
  box-sizing: border-box;
  background-color: white;
}
```

- A propriedade border com o valor none é responsável por retirar a borda de todos os elementos, posteriormente, é adicionado uma borda apenas na parte inferior dos elementos, com o uso da propriedade border-bottom. O padding é utilizado para ampliar o preenchimento das caixas, deixando-as com um espaço mais agradável para a visualização do usuário. A propriedade box-sizing garante que o preenchimento (padding) e as bordas sejam incluídas no valor final dos elementos.
- O resultado seria o seguinte:

Nome

Seu nome completo...

Email

Seu email...

Selecione o curso que deseja:

- ☐ Excel 365
- ☐ Excel 365 - Avançado
- ☐ Excel 365 - Dashboard
- ☐ Excel 365 - Tabelas Dinâmicas

Período

Manhã



Você tem experiência com Excel?

- ☐ Sim
- ☐ Não

- Em conjunto com a alteração anterior, podemos adicionar também um seletor responsável por modificar os campos selecionados (em foco), para isso, utiliza-se o seletor **:focus**, confira:

```
input[type=text]:focus, input[type=email]:focus, select:focus{
  outline: none;
  border-bottom: 2px solid #007bff;
}
```

- A propriedade "outline" com o valor "none" é responsável por remover a borda padrão criada pelos navegadores quando o campo for selecionado. Com o código acima, apenas a borda inferior do campo selecionado seria modificada, confira abaixo um exemplo com o campo "Email" selecionado:

Nome

Seu nome completo...

Email

Seu email...

Selecione o curso que deseja:

- ☐ Excel 365
- ☐ Excel 365 - Avançado
- ☐ Excel 365 - Dashboard
- ☐ Excel 365 - Tabelas Dinâmicas

Período

Manhã

Você tem experiência com Excel?

- ☐ Sim
- ☐ Não

• Ajustando os labels

- Utilizando a mesma ideia dos ajustes nos inputs, podemos também alterar os labels do nosso formulário. Geralmente as alterações nos labels envolvem as margens e a fonte utilizada, confira abaixo:

```
label{
  display: block;
  margin-top: 1em;
  font-weight: bold;
}
```

- Com o código acima, adicionamos uma margem na parte superior do label, definimos a exibição em bloco e também alteramos o texto para negrito, confira o resultado:

Nome

Seu nome completo...

Email

Seu email...

Selecione o curso que deseja:

☐

Excel 365

☐

Excel 365 - Avançado

☐

Excel 365 - Dashboard

☐

Excel 365 - Tabelas Dinâmicas

Período

Manhã



Você tem experiência com Excel?

☐

Sim

- Veja que os labels dos checkboxes e dos radios também foram modificados, a ideia seria alterar apenas os labels dos inputs e os parágrafos (p), para resolver este problema, precisamos alterar os labels específicos, fazendo-os voltar a exibição inline:

```
label{
  display: block;
  margin-top: 1em;
  font-weight: bold;
}
label[for*=excel], label[for*=exp]{
  display: inline;
  font-weight: normal;
}
```

- Com o código anterior, apenas os labels com o atributo "for" que possuam os valores que possuam as palavras "excel" e "exp" seriam afetados, no caso, apenas os labels do checkboxes e dos radios, confira o resultado:

Nome

Seu nome completo...

Email

Seu email...

Selecione o curso que deseja:

- ☐ Excel 365
- ☐ Excel 365 - Avançado
- ☐ Excel 365 - Dashboard
- ☐ Excel 365 - Tabelas Dinâmicas

Período

Manhã



Você tem experiência com Excel?

- ☐ Sim
- ☐ Não

- Os parágrafos "Selecione o curso que deseja:" e o "Você tem experiência com Excel?" não foram afetados, isso ocorre pois eles foram criados com o elemento p, para modificá-los o processo seria o seguinte:

```
label, form p{
    display: block;
    margin-top: 1em;
    margin-bottom: 0em; ←
    font-weight: bold;
}
label[for*=excel], label[for*=exp]{
    display: inline;
    font-weight: normal;
}
```

- A adição do seletor acima garante que todos os parágrafos presentes dentro do formulário também utilizem as mesmas

propriedades dos labels e também removeria a borda inferior padrão dos parágrafos, o resultado seria o seguinte:

Nome

Seu nome completo...

Email

Seu email...

Selecione o curso que deseja:

- ☐ Excel 365
- ☐ Excel 365 - Avançado
- ☐ Excel 365 - Dashboard
- ☐ Excel 365 - Tabelas Dinâmicas

Período

Manhã

Você tem experiência com Excel?

- ☐ Sim
- ☐ Não

• Ajustando o botão do formulário

- O botão do formulário é criado com o elemento input, porém, com o tipo "submit", a alteração dele segue a mesma ideia feita para os inputs do tipo texto e email visto anteriormente, confira o código abaixo:

```
input[type=submit] {  
    width: 100%;  
    background-color: #2085c0;  
    color: white;  
    padding: 14px 20px;  
    margin: 2em 0;  
    border: none;  
    border-radius: 4px;  
    cursor: pointer;  
    font-weight: bold;  
    text-transform: uppercase;  
}
```

- Com o código acima definimos que os inputs do tipo submit seguem outras configurações, que resultariam no seguinte:

Nome

Seu nome completo...

Email

Seu email...

Selecione o curso que deseja:

- ☐ Excel 365
- ☐ Excel 365 - Avançado
- ☐ Excel 365 - Dashboard
- ☐ Excel 365 - Tabelas Dinâmicas

Período

Manhã



Você tem experiência com Excel?

- ☐ Sim
- ☐ Não

ENVIAR

- Para adicionar um efeito extra ao botão, podemos modificar o estado "hover", ou seja, quando o usuário estiver com o mouse posicionado sobre o botão, confira:

```
input[type=submit]:hover {  
    background-color: #0e679b;  
}
```

- O resultado seria o seguinte:

ENVIAR



- O código utilizado para a produção deste formulário pode ser conferido abaixo:

HTML:

```
<form action="">

  <label for="nome">Nome</label>
  <input type="text" id="nome" name="nomecompleto" placeholder="Seu
nome completo...">

  <label for="email">Email</label>
  <input type="email" id="email" name="email" placeholder="Seu email...">

  <p>Selecione o curso que deseja: </p>
  <input type="checkbox" id="excel" name="excel" value="excel">
  <label for="excel"> Excel 365</label> <br>
  <input type="checkbox" id="excel-av" name="excel-av" value="excel-av">
  <label for="excel-av"> Excel 365 - Avançado</label> <br>
  <input type="checkbox" id="excel-dash" name="excel-dash" value="excel-
dash">
  <label for="excel-dash"> Excel 365 - Dashboard</label> <br>
  <input type="checkbox" id="excel-tabela" name="excel-tabela"
value="excel-tabela">
  <label for="excel-tabela"> Excel 365 - Tabelas Dinâmicas</label> <br>

  <label for="periodo">Período</label>
  <select id="periodo" name="periodo">
    <option value="manha">Manhã</option>
    <option value="tarde">Tarde</option>
    <option value="noite">Noite</option>
  </select>

  <p>Você tem experiência com Excel?</p>
  <input type="radio" id="exp-s" name="experiencia" value="S">
  <label for="exp-s">Sim</label> <br>
  <input type="radio" id="exp-n" name="experiencia" value="N">
  <label for="exp-n">Não</label>

  <input type="submit" value="Enviar">
</form>
```

CSS:

```
body{
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    margin:20px;
}

input[type=text], input[type=email], select{
    width: 100%;
    border: none;
    border-bottom: 2px solid gray;
    padding: 12px 20px;
    box-sizing: border-box;
    background-color: white;
}

input[type=text]:focus, input[type=email]:focus, select:focus{
    outline: none;
    border-bottom: 2px solid deepskyblue;
}

label, form p{
    display: block;
    margin-top: 1em;
    margin-bottom: 0em;
    font-weight: bold;
}

label[for*=excel], label[for*=exp]{
    display: inline;
    font-weight: normal;
}
```

```
input[type=submit] {  
    width: 100%;  
    background-color: #2085c0;  
    color: white;  
    padding: 14px 20px;  
    margin: 2em 0;  
    border: none;  
    border-radius: 4px;  
    cursor: pointer;  
    font-weight: bold;  
    text-transform: uppercase;  
}  
input[type=submit]:hover {  
    background-color: #0e679b;  
}
```


18 - Ajustando o layout com o CSS



- Nesta lição você utilizou o conteúdo visto durante o curso para ajustar alguns pontos do seu layout.
- Uma das principais mudanças foi a aplicação da propriedade "position" com o valor "sticky" à barra de menu do site

```
/* Barra de navegação */  
.topo {  
  padding: 5px 0px;  
  overflow: hidden;  
  background-color: rgb(95, 90, 90);  
  → position: -webkit-sticky; /* Safari */  
  → position: sticky;  
  top: 0;  
}
```

- Isso alterou o menu do site, deixando-o fixo na parte superior, quando o usuário rolasse por ele. Abaixo temos o site em seu estado inicial



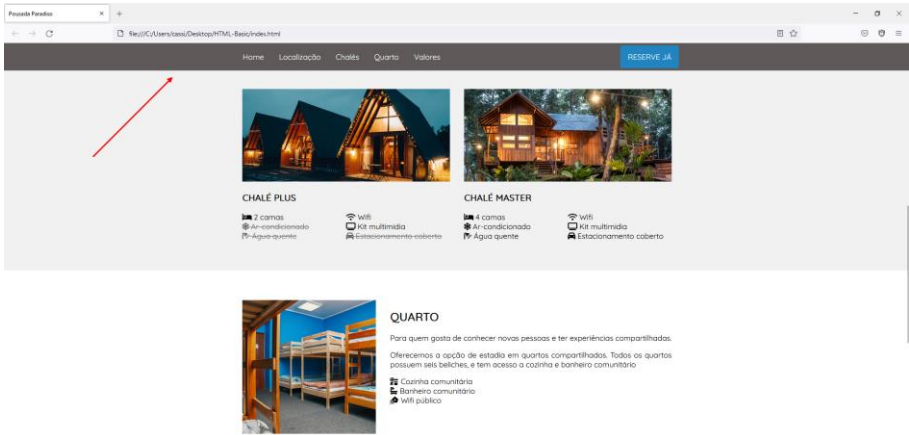
LOCALIZAÇÃO

Um ambiente tranquilo e familiar, nossa equipe brinda um atendimento cordial e despretensioso que fará possível experimentar das suas hospedagens a magia insuperável do lugar, que não vai mais querer que se vá sem ela.

A Pousada Paradiso oferece acomodações para diferentes necessidades, confortáveis e decoradas com muito bom gosto. Ficamos felizes em compartilhar este cenário do paraíso que oficialmente equiparamos.

#natureza #relax #bemestar #paz

- E agora o site após o usuário utilizar a barra de rolagem, ultrapassando a área inicial do menu



- Confira o menu fixo na parte superior do site;

19 - Finalizando o projeto



- Nesta lição você finalizou o projeto desenvolvido durante o curso.

FIM DA APOSTILA

Método CGD ® - Todos os direitos reservados.

Protegidos pela Lei 5988 de 14/12/1973.

**Nenhuma parte desta apostila poderá ser copiada sem prévia
autorização.**