

Professor: Davi de Barros

DESENVOLVIMENTO WEB EM HTML5, CSS, JAVASCRIPT E PHP

PROJETO FINAL EM GRUPO DE 5 PESSOAS – FORA O PROJETO NÃO HAVERÁ AVALIAÇÃO

DATA DE ENTREGA:

**22/11/2023 - MANHÃ (apresentação na biblioteca) | pré-apresentação em sala
dia 15/11/2023 (para o professor)**

**23/11/2023 – NOITE (apresentação na biblioteca) | pré-apresentação em sala
dia 16/11/2023 (para o professor)**

**O PROJETO DEVE SER APRESENTADO PELA EQUIPE. DEVE
SER USADO NO PROJETO O QUE SE APRENDEU EM PHP,
JAVASCRIPT (JQUERY), CSS3/HTML5, BOOTSTRAP, SQLITE
COM PHP.**

O SITE DEVERÁ SER RESPONSIVO.

INFRAESTRUTURA PARA CASA - EXERCÍCIO

- Instalação da IDE VSCode e Google Chrome.
- Download do Bootstrap 4.6 e JQuery 3.5.1.
- Instalação do PHP versão 7.4 em diante e habilitação do módulo do SQLite para PHP.

INFRAESTRUTURA PARA AS AULAS / SALA - EXERCÍCIO

Utilização do ambiente replit.com.

Introdução a HTML e CSS

– Aula 1

HTML (Hypertext Markup Language) é uma linguagem de marcação usada para criar e estruturar páginas na internet. É uma espécie de "esqueleto" que dá forma ao conteúdo de um site, permitindo definir títulos, parágrafos, imagens, links e outros elementos que são exibidos no navegador da web.

Cada elemento é representado por uma "tag" ou "etiqueta" que é colocada entre colchetes angulares, como <tag>. Essas tags indicam como o conteúdo deve ser exibido e interagido pelos usuários. Em resumo, o HTML é a base para construir páginas da web e torna possível navegar pela internet e acessar todo tipo de informação e conteúdo online.

A linguagem de marcação, como o HTML (Hypertext Markup Language), é chamada assim porque sua função principal é marcar ou estruturar o conteúdo em um documento, indicando como o texto e outros elementos devem ser exibidos e interpretados pelo navegador da web. Em vez de ser uma linguagem de programação, como o JavaScript ou o Python, o HTML é usado para criar documentos que consistem em texto formatado e outros elementos, como imagens, links e formulários.

O termo "marcação" refere-se ao processo de adicionar "marcas" ou "tags" no texto para indicar sua estrutura e significado. Essas tags são colocadas entre colchetes angulares ("<" e ">") e são interpretadas pelo navegador para apresentar o conteúdo de acordo com as instruções fornecidas.

Os arquivos devem ser salvos com a extensão .html, sendo a home page salva com o nome index.html.

Qual a finalidade de cada comando abaixo na estrutura base da página em html?

<!DOCTYPE html>

<html>

<head>

<title></title>

<meta charset="utf-8">

</head>

<body>

</body>

</html>

Títulos

Quando queremos indicar que um texto é um título em nossa página, utilizamos as tags de heading em sua marcação:

```
<h1>Mirror Fashion.</h1>
```

```
<h2>Bem-vindo à Mirror Fashion, sua loja de roupas e acessórios.</h2>
```

Parágrafos

<p>Nenhum item na sacola de
compras.**</p>**

Quebra de linha

**
**

Imagens

```
<figure>
```

```

```

```
<figcaption>Fuzz Cardigan por R$ 129,90</figcaption>
```

```
</figure>
```


O HTML 5 introduziu duas novas tags específicas para imagem: `<figure>` e `<figcaption>`. A tag `<figure>` define uma imagem com a conhecida tag ``. Além disso, permite adicionar uma legenda para a imagem por meio da tag `<figcaption>`.

Estrutura de um projeto

▶  css

▶  img

 js

 index.html

 sobre.html

IDES

Comentários

<!-- início do cabeçalho -->

<div id="header">

<p>Esse parágrafo está "dentro" da área principal.</p>

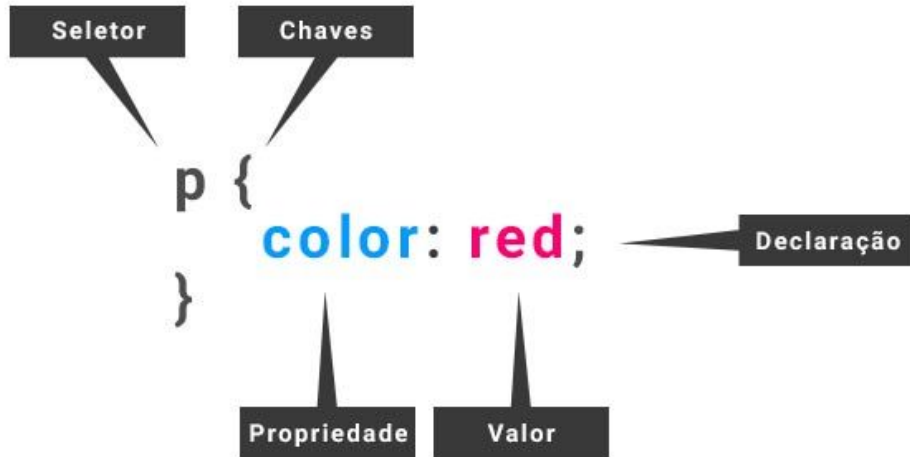
</div>

<!-- fim do cabeçalho -->

CSS - CONCEITOS

TAG, CLASSE, ID – OS SELETORES

Estilizando com CSS



Os seletores em CSS são usados para selecionar elementos HTML específicos e aplicar estilos a eles. Tipos de seletores em CSS:

Seletores por ID: Os seletores por ID selecionam um elemento HTML baseado no valor do seu atributo `id`. A sintaxe para um seletor por ID é `#valor_id`, onde `valor_id` é o valor do atributo `id` do elemento que você deseja selecionar. Por exemplo, para selecionar um elemento com o atributo `id="meu-id"`, você pode usar o seletor `#meu-id`.

Seletores por classe: Os seletores por classe selecionam elementos HTML baseados no valor do seu atributo `class`. A sintaxe para um seletor por classe é `.valor_classe`, onde `valor_classe` é o valor do atributo `class` dos elementos que você deseja selecionar. Por exemplo, para selecionar todos os elementos com o atributo `class="minha-classe"`, você pode usar o seletor `.minha-classe`.

Seletores por tag: Os seletores por tag selecionam elementos HTML baseados no nome da sua tag. A sintaxe para um seletor por tag é simplesmente o nome da tag. Por exemplo, para selecionar todos os elementos `<p>` em uma página, você pode usar o seletor `p`.

Estilizando com CSS

```
p {
```

```
    color: blue;
```

```
    background-color: yellow;
```

```
}
```

Estilizando com CSS

Chamando o arquivo css para o html.

```
<link rel="stylesheet" href="css/style.css">
```

Estilizando com CSS os tipos de fontes

Podemos definir fontes com o uso da propriedade font-family.

```
body {  
    font-family: Arial, Helvetica, “sans-serif”;  
}
```

Estilizando com CSS o texto

```
.estilotexto {  
    text-align: right;  
    line-height: 3px; /* tamanho da altura de cada linha */  
    letter-spacing: 3px; /* tamanho do espaço entre cada letra */  
    word-spacing: 5px; /* tamanho do espaço entre cada palavra */  
    text-indent: 30px; /* tamanho da margem da primeira linha do texto */  
}
```

Aplicando a classe ao html.

```
<p class="estilotexto">Conteúdo novo.</p>
```

Bordas e Imagens de fundo

A propriedade `background-image` permite indicar um arquivo de imagem para ser exibido ao fundo do elemento. Por exemplo:

```
background-image: url(sobre-background.jpg);
```

Com essa declaração, o navegador vai requisitar um arquivo `sobre-background.jpg`, que deve estar na mesma pasta do arquivo CSS onde consta essa declaração.

As propriedades do CSS para definirmos as bordas de um elemento nos apresentam uma série de opções.

Podemos, para cada borda de um elemento, determinar sua cor, seu estilo de exibição e sua largura. Por exemplo:

```
border-color: red;  
border-style: solid;  
border-width: 1px;
```

CORES NA WEB

RGB e HEXADECIMAL

```
#cabecalho {  
    color: rgb(255, 200, 0);  
}
```

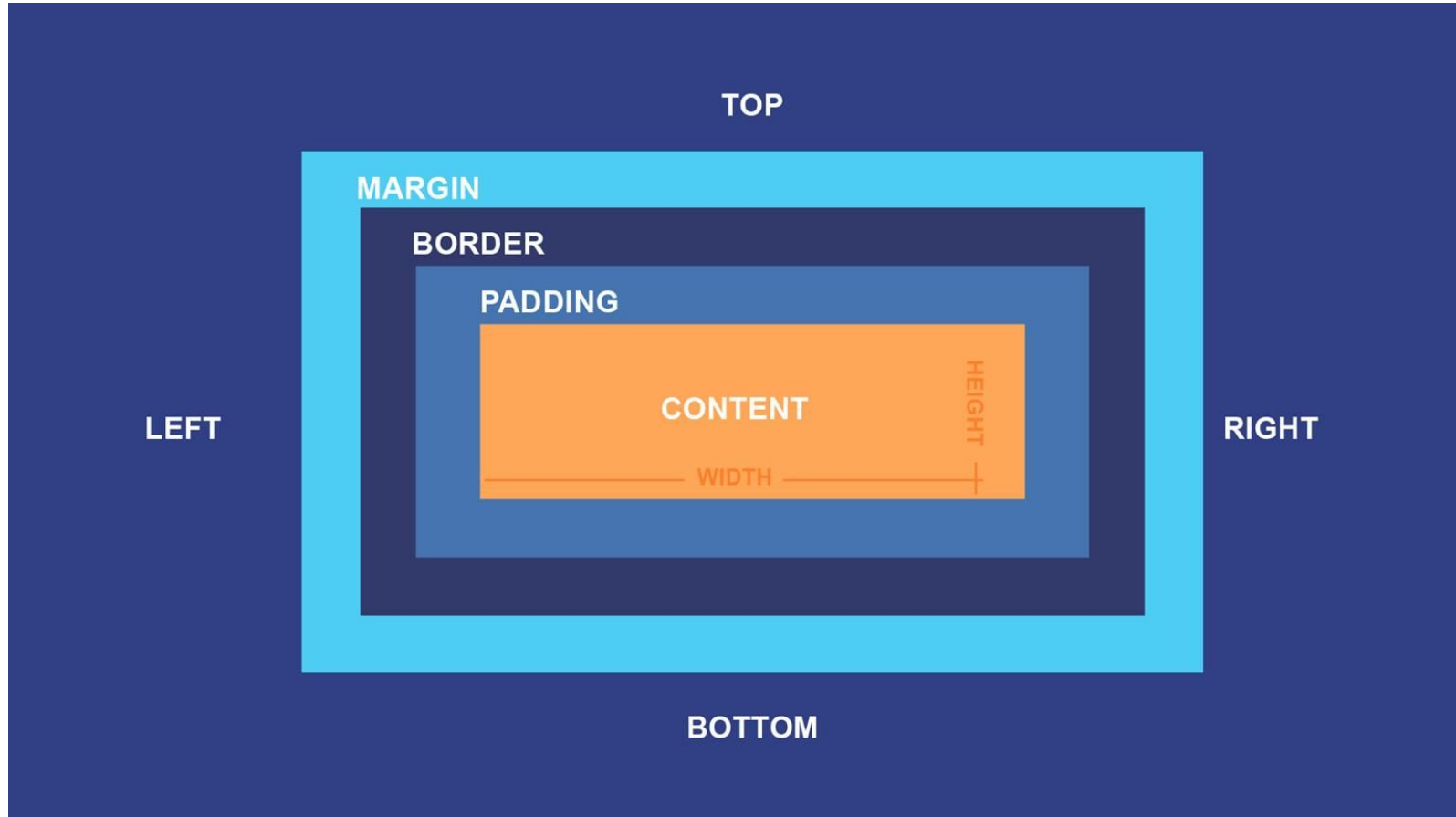
```
#rodape {  
    background: #F2EDED;  
}
```


TÓPICOS ADICIONAIS

- listas ordenadas e não ordenadas
- padding
- margin
- width e height
- links interno/externo

Para criar um link com um arquivo html local ou site externo usamos a tag **a** com a propriedade **href**.

BOX MODEL



HTML SEMÂNTICO - PROTOTIPAGEM

SEMÂNTICA

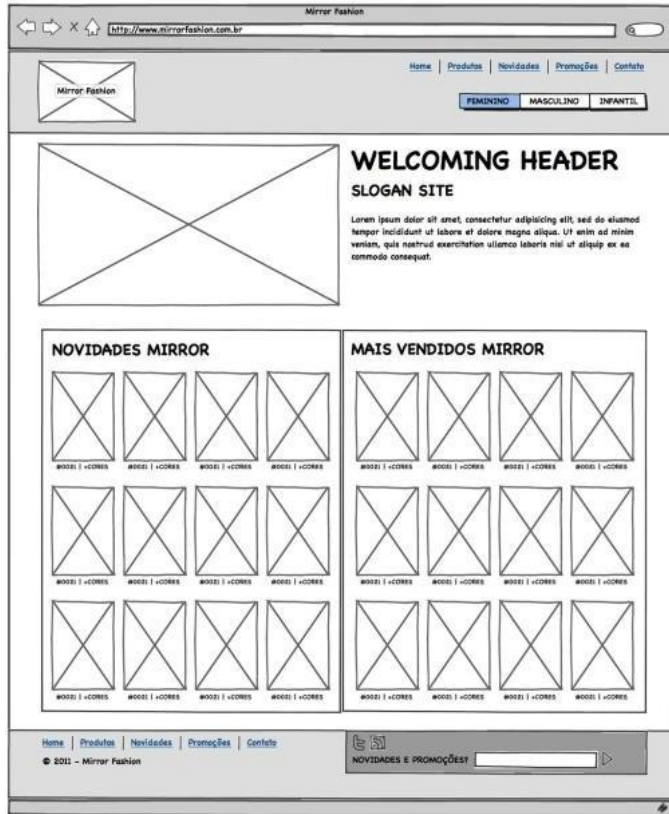
As novas tags do HTML5 trazem novos significados semânticos para usarmos em elementos HTML. Em vez de simplesmente agrupar os elementos do cabeçalho em um div genérico e sem significado, usamos uma tag header que carrega em si o significado de representar um cabeçalho.

SEMÂNTICA

Tag **<aside>**: A tag `<aside>` é usada para marcar um conteúdo que está relacionado ao conteúdo principal da página, mas que pode ser considerado separado dele. Por exemplo, você pode usar a tag `<aside>` para marcar uma barra lateral com informações adicionais ou links relacionados ao conteúdo principal.

Tag **<article>**: A tag `<article>` é usada para marcar um conteúdo independente em uma página da web. Isso significa que o conteúdo dentro de uma tag `<article>` deve ser capaz de ser entendido por si só, mesmo que seja removido do resto da página. Por exemplo, você pode usar a tag `<article>` para marcar uma postagem de blog ou uma notícia.

Tag **<section>**: A tag `<section>` é usada para definir uma seção em um documento HTML. Ela é usada para agrupar conteúdo temático, geralmente incluindo um cabeçalho. Uma página da web pode normalmente ser dividida em seções para introdução, conteúdo, e contato.



PROJETO WEB

- Requisitos
- UX/UI
- Wireframe (imagem ao lado)
- Criação de protótipo (figma)
- Layout
- Projeto final

O link abaixo mostra um site com apoio para montagem do código para layout.

<https://htmlboilerplates.com>

POSICIONAMENTO EM CSS

O position é uma propriedade CSS que define como um elemento deve ser posicionado página, ou seja, no corpo do documento o body. Quais valores esta propriedade pode assumir?

absolute – Sua posição está relacionada ao seu elemento pai.

fixed - elemento fixo na página. Não se move com a barra de rolagem.

relative – Posiciona o elemento em relação ao elemento anterior (div, img).

static – elemento fica fixo na página, porém se move com a barra de rolagem. Top, bottom, etc, não impactam em sua posição.

HEADER E FOOTER

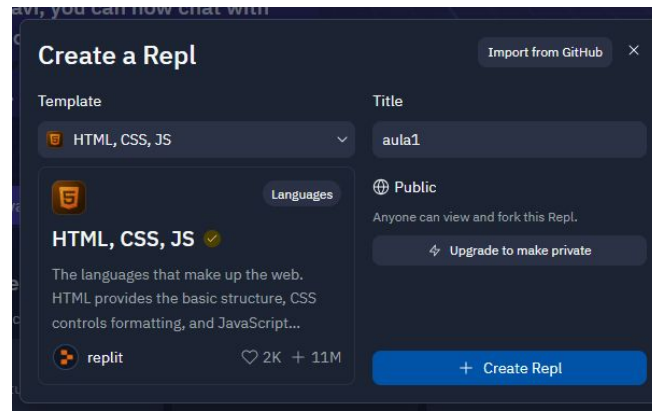
A tag header serve para criar o cabeçalho na página e a tag footer cria o rodapé.

A tag nav serve para criar a barra de navegação do site.

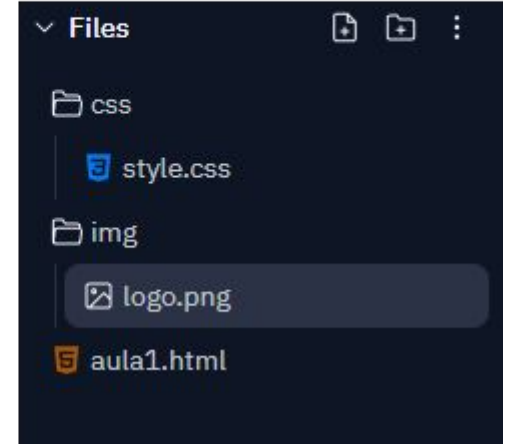
Vamos fazer a atividade prática da aula 1?

Acesse o replit.com e crie um novo projeto com Template em HTML, CSS e Java Script (JS). Nome do title aula1.

Clique no botão create repl.



Crie a estrutura da página assim:



Digite o código. Depois de concluir clique no botão RUN, para visualizar seu projeto. O projeto aparecerá na tela ao lado (Webview).

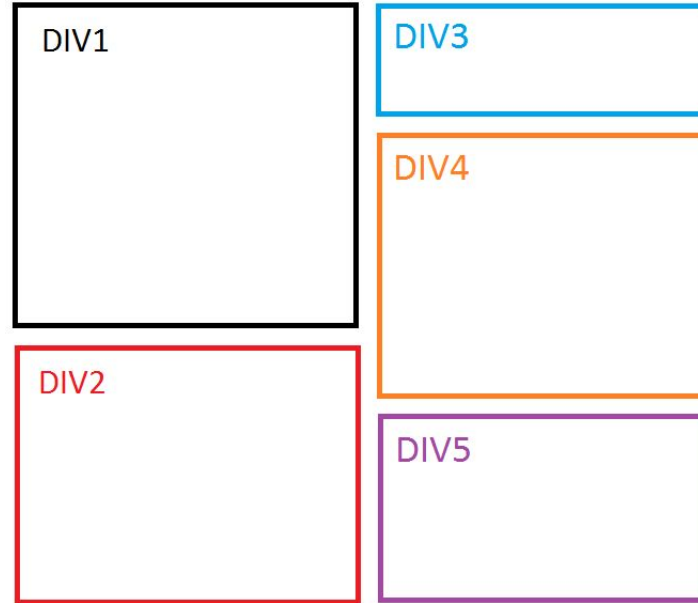
Formulários, animações e JavaScript – Aula 2

Formulários

- Um formulário é um elemento que permite ao usuário enviar dados para o servidor. Ele é definido pela tag `<form>` e pode conter vários tipos de inputs.
- Um input é um elemento que permite ao usuário inserir dados em um formulário. Ele é definido pela tag `<input>` e pode ter vários atributos, como `type`, `id`, `name` e `value`. O atributo `type` define o tipo de dados que o input aceita, como `text`, `email`, `password`, `checkbox`, etc.
- Um input do tipo text aceita qualquer texto simples. Ele é definido pelo atributo `type="text"` na tag `<input>`.
- Um input do tipo email aceita apenas endereços de e-mail válidos. Ele é definido pelo atributo `type="email"` na tag `<input>`.
- Um textarea é um elemento que permite ao usuário inserir um texto de várias linhas em um formulário. Ele é definido pela tag `<textarea>` e pode ter atributos como `id`, `name`, `rows` e `cols` para especificar o tamanho da área de texto.
- Um input do tipo submit cria um botão que envia os dados do formulário para o servidor. Ele é definido pelo atributo `type="submit"` na tag `<input>` e pode ter um atributo `value` para especificar o texto do botão.

Divs em html

A tag HTML **<div>**, que corresponde a abreviatura de Division, é um elemento HTML utilizado para dividir e agrupar os itens que serão exibidos em uma página.



Float e Clear em CSS

O float é uma propriedade que permite posicionar um elemento de forma que ele flutue à esquerda ou à direita de outro elemento. Por exemplo, você pode usar o float para fazer uma imagem ficar à esquerda de um texto em um container. O float pode ter os valores left, right ou none.

O clear é uma propriedade que permite controlar o comportamento dos elementos flutuantes, evitando que eles se sobreponham a outros elementos. Por exemplo, você pode usar o clear para fazer um parágrafo ficar abaixo de uma imagem flutuante, e não ao lado dela. O clear pode ter os valores left, right, both, none ou inherit.

Outras propriedades em CSS

O border-radius é uma propriedade que permite arredondar os cantos de um elemento. Você pode definir o raio de cada canto individualmente ou usar um valor único para todos. A sintaxe é:

border-radius: valor1 valor2 valor3 valor4;

/* ou */

border-radius: valor;

Outras propriedades em CSS

O box-shadow é uma propriedade que permite adicionar uma sombra em quase qualquer elemento. Você pode definir a posição, o tamanho, a cor e o tipo da sombra. A sintaxe é:

box-shadow: horizontal vertical blur spread color inset;

Outras propriedades em CSS

Propriedades

- **horizontal** é a distância horizontal entre o elemento e a sombra. Um valor positivo significa que a sombra fica à direita do elemento, e um valor negativo significa que a sombra fica à esquerda do elemento.
- **vertical** é a distância vertical entre o elemento e a sombra. Um valor positivo significa que a sombra fica abaixo do elemento, e um valor negativo significa que a sombra fica acima do elemento.
- **blur** é o raio de desfoque da sombra. Um valor maior significa que a sombra fica mais borrada, e um valor zero significa que a sombra fica nítida.
- **spread** é o tamanho da sombra. Um valor positivo significa que a sombra fica maior que o elemento, e um valor negativo significa que a sombra fica menor que o elemento.
- **color** é a cor da sombra. Você pode usar qualquer formato de cor válido em CSS, como nomes, valores hexadecimais, RGB ou HSL.
- **inset** é uma palavra-chave opcional que indica se a sombra é interna ou externa ao elemento. Se você usar inset, a sombra fica dentro do elemento. Se você não usar inset, a sombra fica fora do elemento.

Outras propriedades em CSS

Um exemplo de uso é:

```
div {  
  border-radius: 10px;  
  box-shadow: 5px 5px 10px -2px rgba(0,0,0,0.5) inset;  
}
```

Nesse exemplo, a sombra tem 5 pixels de distância horizontal e vertical do elemento, 10 pixels de raio de desfoque, -2 pixels de tamanho, cor preta com 50% de opacidade e é interna ao elemento.

Outras propriedades em CSS

As cores gradientes são uma forma de criar um efeito de transição entre duas ou mais cores. Você pode usar gradientes lineares ou radiais. A sintaxe é:

```
background: linear-gradient(direção, cor1, cor2, ...);  
/* ou */  
background: radial-gradient(forma, cor1, cor2, ...);
```

Um exemplo de uso é:

```
div {  
  border-radius: 10px;  
  background: linear-gradient(to right, red, yellow, green);  
}
```

Capitalizando em CSS

A tag `text-transform` em CSS é uma propriedade que especifica como capitalizar um texto de um elemento. Você pode usar essa propriedade para que o texto apareça com todas as letras maiúsculas, todas minúsculas, ou com cada palavra em maiúscula. A sintaxe é:

```
text-transform: valor;
```

O valor pode ser um dos seguintes:

none: não altera o texto.

uppercase: transforma o texto em maiúsculas.

lowercase: transforma o texto em minúsculas.

capitalize: transforma a primeira letra de cada palavra em maiúscula.

Organizando elementos em CSS

A propriedade `display` especifica o comportamento de exibição de um elemento.

A propriedade `display` em CSS define se um elemento HTML é tratado como um bloco ou um elemento inline, grid ou flex. Por exemplo, se você usar `display: block`, o elemento vai ocupar toda a linha. Se você usar `display: inline`, o elemento vai ocupar apenas o espaço necessário e não criar novas linhas. Se você usar `display: none`, o elemento vai ser removido da tela e do fluxo do documento.

Pseudo-classes em CSS

As pseudo-classes em CSS são palavras-chave que se adicionam aos seletores e que especificam um estado especial do elemento selecionado. Por exemplo, `:hover` pode ser usado para alterar a cor de um link quando o usuário passar o cursor sobre ele.

As pseudo-classes `a:hover` e `a:visited` são usadas para estilizar os links de acordo com o seu estado:

- `a:hover` aplica um estilo quando o usuário passa o cursor sobre o link, mas não clica nele.
- `a:visited` aplica um estilo quando o usuário já visitou o link anteriormente.

Pseudo-classes em CSS

Um exemplo de uso é:

```
/* link não visitado */  
a:link {  
  color: blue;  
}
```

```
/* link visitado */  
a:visited {  
  color: purple;  
}
```

```
/* link com cursor sobre ele */  
a:hover {  
  color: red;  
}
```

Nesse exemplo, os links não visitados ficam azuis, os visitados ficam roxos, e os que estão com o cursor sobre eles ficam vermelhos.

Transformação e transição em CSS

A propriedade `transition` faz com que as mudanças em um elemento sejam mais suaves e demoradas. Por exemplo, se você quiser que um botão mude de cor quando o usuário clicar nele, você pode usar `transition` para fazer a mudança de cor durar alguns segundos, em vez de ser imediata.

A propriedade `transform` faz com que um elemento mude de forma ou de posição. Por exemplo, se você quiser que um quadrado vire um losango quando o usuário passar o cursor sobre ele, você pode usar `transform` para girar o quadrado 45 graus.

```
a {  
    text-decoration: none; /* Remove o sublinhado dos links */  
    text-transform: uppercase; /* Transforma o texto em maiúsculo */  
    color: white;  
    transition: transform 2s;  
}  
  
/* Define o estilo dos links quando o mouse passa por cima */  
a:hover {  
    color: blue !important;  
    transform: scale(1.3);  
}
```

JavaScript

JavaScript é uma linguagem de programação que permite criar efeitos dinâmicos e interativos em páginas web. Por exemplo, você pode usar JavaScript para fazer um botão mudar de cor quando o usuário clica nele.

JavaScript

Comandos em JavaScript:

`document`: é o que representa a página que você vê no navegador. Ele tem informações sobre os elementos que estão na página, como textos, imagens, botões, etc.

`getElementById`: é uma forma de pegar um elemento da página usando o seu id. O id é um nome único que você pode dar para um elemento. Por exemplo, se você tem um botão com `id="meu-botao"`, você pode usar `document.getElementById("meu-botao")` para pegar esse botão.

`body`: é o elemento que contém tudo o que você vê na página. Você pode usar `document.body` para pegar esse elemento.

`style`: é uma forma de mudar a aparência de um elemento. Você pode mudar coisas como cor, tamanho, posição, etc. Por exemplo, se você quer mudar a cor do texto de um elemento para vermelho, você pode usar `elemento.style.color = "red"`.

JavaScript

Comandos em JavaScript:

`backgroundColor`: é uma forma de mudar a cor de fundo de um elemento. Por exemplo, se você quer mudar a cor de fundo da página para azul, você pode usar `document.body.style.backgroundColor = "blue"`.

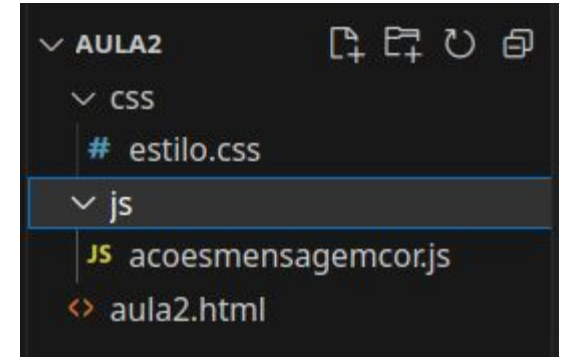
`color`: é uma forma de mudar a cor do texto de um elemento. Por exemplo, se você quer mudar a cor do texto de um botão para branco, você pode usar `botao.style.color = "white"`.

`checked`: é uma forma de saber se um checkbox ou um radio está marcado ou não. Por exemplo, se você quer saber se o usuário marcou um checkbox com `id="aceito"`, você pode usar `document.getElementById("aceito").checked`. Isso vai retornar `true` ou `false`.

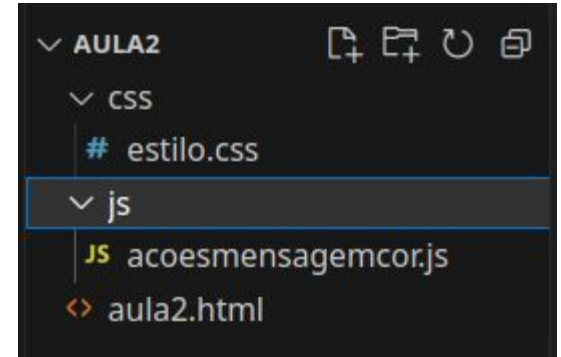
`alert`: é uma forma de mostrar uma mensagem na tela. Por exemplo, se você quer mostrar "Olá mundo!" na tela, você pode usar `alert("Olá mundo!")`. Isso vai abrir uma caixa com a mensagem e um botão de OK.

Exercício prático

Organização da pasta e arquivos



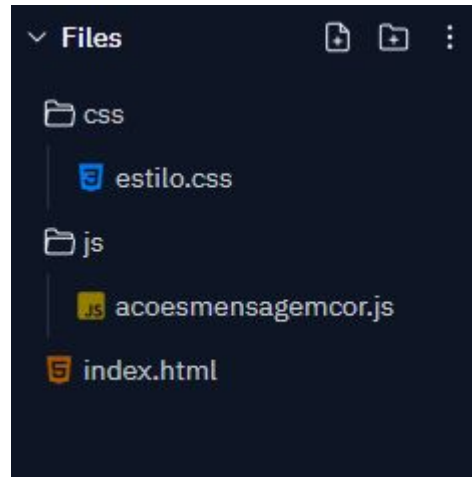
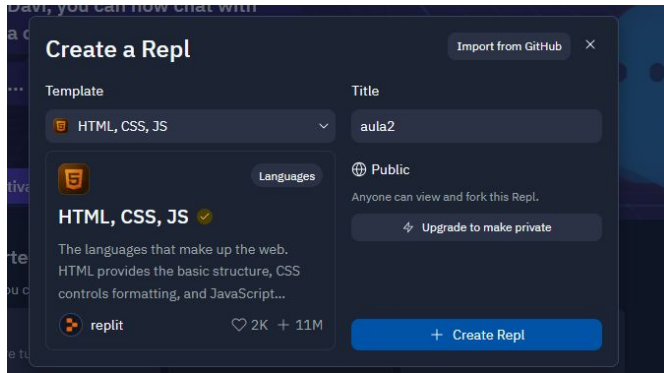
**Como chamamos os
arquivos css e
javascript em nosso
arquivo html?**



Vamos codar os arquivos da aula 2?

Acesse o replit.com para fazer.

1. Crie um projeto chamado **aula2** usando HTML, CSS E JS.
2. Crie a estrutura de arquivos ao lado.
3. Digite os códigos.
4. Clique no botão RUN e teste o projeto no painel da Webview.



Tabelas, integrações, Responsividade – Aula 3

Tabelas

Para inserir tabelas na página nos utilizamos essencialmente das tags table, th, tr e td.

```
<table>
  <tr>
    <th>Nome</th>
    <th>Sobrenome</th>
    <th>Idade</th>
  </tr>
  <tr>
    <td>João</td>
    <td>Silva</td>
    <td>25</td>
  </tr>
  <tr>
    <td>Maria</td>
    <td>Santos</td>
    <td>32</td>
  </tr>
</table>
```

Novas entradas - forms

Em Html 5 existem inputs para arquivos, números, datas, paleta para aumento e diminuição de valores de algum elemento.

```
<h2>Inputs</h2>
<form action="">
  Data: <input type="date"><br><br>
  Número: <input type="number"><br><br>
  Arquivo: <input type="file"><br><br>
  Range: <input type="range"><br><br>
</form>
```


Downloads

Podemos configurar links para download.

```
<a href="https://www.gov.br/ana/pt-br/todos-os-documentos-do-portal/doc-modelo.pdf/@download/file/doc-modelo.pdf"
download>Baixar arquivo PDF</a><br><br>
```

Incorporação

Usando a tag iframe podemos incorporar recursos da Internet como mapas do google e vídeos do youtube.

```
<iframe
  src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3950.004123283576!2d-34.88125638522507!3d-8.
  054279994697031!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.
  1!3m3!1m2!1s0x7ab18c5a4c2e01d%3A0x6f32462f33a5e25b!2sRecife%20-%20PE!5e0!3m2!1spt-BR!2sbr!4v1629132233677!5m2!1spt
  -BR!2sbr"
  width="600" height="450" style="border:0;" allowfullscreen="" loading="lazy"></iframe><br><br>

<!-- Link para um vídeo no YouTube -->
<h2>Vídeo do YouTube</h2>
<iframe width="560" height="315" src="https://www.youtube.com/embed/n_Pf1BgHS7c?si=zk97zgrv-BwwbpdY"
  title="YouTube video player" frameborder="0"
  allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"
  allowfullscreen></iframe>
```

Validação

Podemos usar validações para verificar dados válidos em formulários: e-mail preenchido corretamente, mínimo de caracteres utilizado, se o campo obrigatório foi preenchido.

```
<!-- Validação em HTML5 -->
<h2>Validação em HTML5</h2>
<form action="">
  Email: <input type="email" required><br><br>
  Senha: <input type="password" minlength="8" required><br><br>
  <input type="submit">
</form>
```

Media Types

Media types em CSS são usados para especificar a mídia ou o dispositivo para o qual uma folha de estilo CSS se aplica. Isso permite que você crie folhas de estilo específicas para diferentes tipos de dispositivos, como impressoras ou telas, e adapte a apresentação do conteúdo de acordo.

Aqui está um exemplo que mostra como usar media types em CSS para aplicar estilos diferentes para telas e impressoras:

Media Types

```
/* Estilos para telas */  
@media screen {  
  body {  
    background-color: lightblue;  
  }  
}
```

```
/* Estilos para impressoras */  
@media print {  
  body {  
    background-color: white;  
  }  
}
```

Media Types

Neste exemplo, usamos duas regras `@media` para especificar estilos diferentes para telas e impressoras. A primeira regra `@media screen` aplica os estilos dentro das chaves `{}` apenas em dispositivos com o media type `screen`, como monitores de computador ou telas de celular. A segunda regra `@media print` aplica os estilos dentro das chaves `{}` apenas em dispositivos com o media type `print`, como impressoras.

Alguns media types comuns usados em CSS incluem:

- `all`: adequado para todos os dispositivos
- `print`: adequado para impressoras
- `screen`: adequado para telas de computador, celular e tablet
- `speech`: adequado para leitores de tela

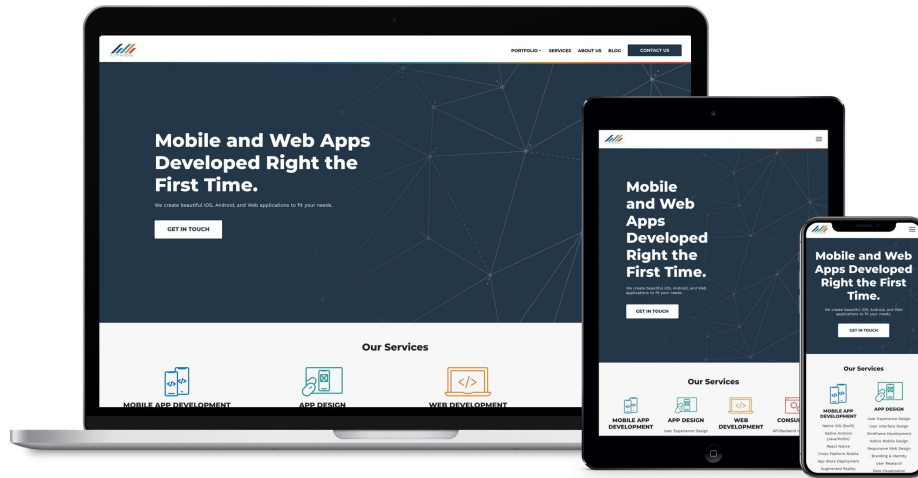
Media Queries

Media queries são uma técnica em CSS que permite aplicar estilos condicionalmente com base em características do dispositivo do usuário, como a largura da tela ou tipo de tela. Isso é útil para criar designs responsivos que se adaptam automaticamente a diferentes tamanhos de tela e dispositivos.

Exemplo: https://www.w3schools.com/css/tryit.asp?filename=trycss3_media_queries2

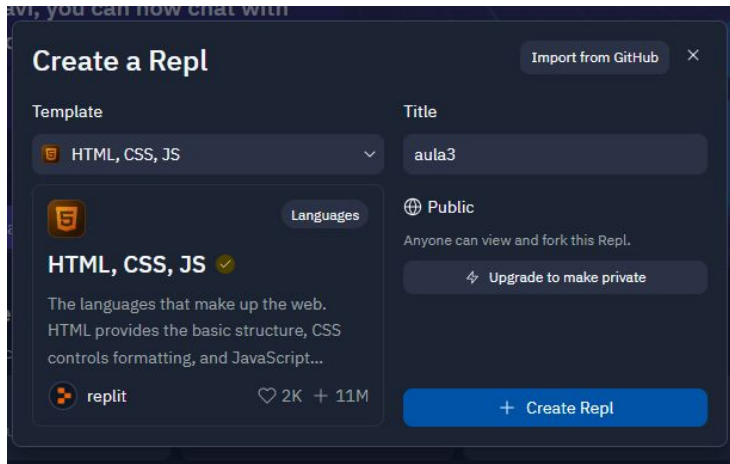
Viewport

O viewport é a área visível de uma página da web para o usuário. O tamanho do viewport varia de acordo com o dispositivo, sendo menor em um celular do que em uma tela de computador.



Vamos codar a aula 3?

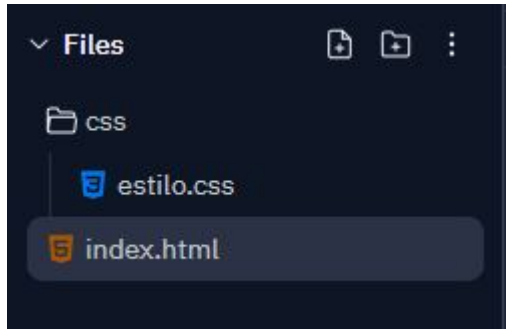
Vamos codar a aula 3?



Acesse o replit.com.

Crie um projeto chamado aula3, e crie a estrutura de arquivos e pastas abaixo.

Digite o código.



Execute o projeto e visualize o resultado no painel lateral.

**Vamos utilizar o material
de apoio?**

**Atividades
complementares
obrigatórias**

Mídias Sociais

Integre sua página com mídias sociais como: botão do twitter para seguir perfil, botão curtir do facebook, etc.

Zend Framework e Doctrine – Criar um mini projeto

<https://www.devmedia.com.br/criando-um-crud-com-zend-framework-2-e-doctrine-2/32100>

Mobile First e Bootstrap,
Ajax e JQuery, PHP e
SQLite – Mini-projetos
finais

Ajax

AJAX é uma técnica de desenvolvimento web que permite que aplicações trabalhem de modo assíncrono, ou seja, sem precisar recarregar a página inteira a cada requisição ao servidor.

JQuery

jQuery é uma biblioteca JavaScript que simplifica a manipulação de elementos HTML, a criação de animações e o desenvolvimento de aplicações AJAX.

O document é um objeto que representa o documento HTML carregado na janela do navegador. O click é um método que vincula uma função a ser executada quando um elemento é clicado. O fadeIn e o fadeOut são métodos que fazem um elemento aparecer ou desaparecer gradualmente, alterando sua opacidade.

Quando você utiliza o \$ seguido de um seletor CSS ou um elemento HTML, está usando a funcionalidade do jQuery para interagir com o DOM (Documento Object Model) da página.

PHP

É uma linguagem de programação para desenvolvimento web.

Bootstrap

O Bootstrap é um framework front-end amplamente usado para o desenvolvimento de páginas da web responsivas e a criação de interfaces visualmente atraentes. Ele deixa a página responsiva se baseando em um grid de 12 colunas.

Mobile First

Mobile first é um conceito que coloca como prioridade o desenvolvimento de sites, páginas e conteúdos para dispositivos móveis. Isso significa que o design e a funcionalidade do site são pensados primeiro para telas pequenas, e depois adaptados para telas maiores.