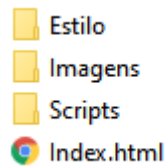


JS - JavaScript

- Criar uma página HTML, conforme Layout, utilizando CSS e JS:

- **Orientações:**

- Organização das Pastas e Arquivos:



- Criar uma página HTML na pasta raiz da atividade (**Index.html**);
- Criar 3 pastas: "**Estilo**", "**Imagens**" e "**Scripts**";
- Criar, dentro da pasta "**Estilo**", um arquivo para o CSS (com extensão .css):
- Descompactar o arquivo **JS_Atividade2.rar** dentro da pasta "**Imagens**";
- Criar, dentro da pasta "**Scripts**", um arquivo para o JS (com extensão .js):
jscript.js
- Usar a META TAG obrigatória de Codificação de Caractere (charset);
- O Título do Navegador deve ser: "Atividade JavaScript";
- Incluir o link para o arquivo de Estilos:

```
<link rel="stylesheet" type="text/css" href="Estilo/??????.css" />
```

- Incluir o ícone para o navegador: "Javascript.png";

```
<link rel="icon" href="Imagens/Javascript.png" />
```

- Adicionar a META TAG de "viewport";

```
<meta name="viewport" content="width=device-width, initial-scale=1" />
```

Index.html

- Layout **responsivo**:



- Layout quando a largura mínima da janela for **768px**:



- Estrutura da página **Index.html**

*** Início Bloco Semântico (**header**) ****

Texto sem tag: Funcionalidades JavaScript

*** Fim Bloco Semântico (**header**) ****

*** Início Bloco Semântico (**main**) ****

- **Bloco** (classes: "**container superior**")
 - **Bloco**
 - **Lista de Seleção**; identificador: "**sel_rede_social**"
Opções:
 - 1º item: "Escolha uma Rede Social..." (selecionado, desabilitado e valor interno vazio)
 - 2º item: "Facebook" (valor interno: "**Facebook**")
 - 3º item: "Instagram" (valor interno: "**Instagram**")
 - 4º item: "Linkedin" (valor interno: "**Linkedin**")
 - 5º item: "Twitter" (valor interno: "**Twitter**")
 - **Bloco**
 - **Imagem** "social-media.png"; identificador: "**img_rede_social**"; texto alternativo: "Redes Sociais"
 - **Parágrafo**; identificador: "**p_rede_social**"; "Redes Sociais"
 - **Bloco**
 - **Texto sem tag**: Clique em uma figura:
 - **Tag para pular uma linha**
 - **Imagem** "Facebook.png"; classe: "**figuras**"; texto alternativo: "Logo Facebook"
 - **Imagem** "Instagram.png"; classe: "**figuras**"; texto alternativo: "Logo Instagram"
 - **Imagem** "Linkedin.png"; classe: "**figuras**"; texto alternativo: "Logo LinkedIn"
 - **Imagem** "Twitter.png"; classe: "**figuras**"; texto alternativo: "Logo Twitter"
 - **Bloco** (identificador: "**div_adiciona_figura**")
*** VAZIO ***
 - **Bloco**
 - **Tabela** identificador: "**tab_principal**"
Caption "Tabela Principal"
 - 1ª linha: Campo/Coluna 1: "Facebook"
 - 2ª linha: Campo/Coluna 1: "Instagram"
 - 3ª linha: Campo/Coluna 1: "LinkedIn"
 - 4ª linha: Campo/Coluna 1: "Twitter"
 - **Bloco**
 - **Tabela** identificador: "**tab_vazia**"
Caption "Tabela Vazia"
 - **Barra de Progressão**; identificador: "**prog_tab_vazia**"; máximo: 5; valor: 0

- Bloco (classes: "container inferior")
 - Bloco
 - Texto sem tag: Selecione uma Linguagem Web:
 - Tag para pular uma linha
 - Campo tipo: "radio"; valor interno: "CSS"
 - Texto sem tag: CSS
 - Campo tipo: "radio"; valor interno: "HTML"
 - Texto sem tag: HTML
 - Campo tipo: "radio"; valor interno: "JavaScript"
 - Texto sem tag: JavaScript
 - Bloco
 - Imagem "code.png"; texto alternativo: "Linguagens de Programação Web"
 - Parágrafo Predefinido: "Programação Web"
 - Bloco
 - Parágrafo: "Lista Principal"
 - Lista Não Ordenada
 - Opções:
 - 1º item: CSS
 - 2º item: HTML
 - 3º item: JavaScript
 - Bloco
 - Parágrafo: "Lista Vazia"
 - Lista Não Ordenada
 - *** VAZIA ***
 - Barra de Progressão; máximo: 6; valor: 0

*** Fim Bloco Semântico (main) ****

- Incluir a referência para o arquivo JavaScript no fim da tag <body>:

```
<script src="Scripts/jscript.js"></script>
```

- **Estilos/Propriedades CSS:**

- Incluir a origem da Fonte:

```
@import url('https://fonts.googleapis.com/css2?family=Kanit:wght@300&display=swap');
```

- :root
 - cor-fonte-principal: darkslateblue;
 - cor-fonte-secundaria: dimgray;
 - cor-fundo-principal: gainsboro;
 - cor-fundo-secundaria: aliceblue;
- HTML e Corpo da Página (body):
 - Margens: 0
 - Efeito de Preenchimento: 0
 - Nome da Fonte: 'Kanit', sans-serif
 - Espessura da Fonte: bold
 - Tamanho da Fonte: 1rem
 - Alinhamento do Texto: center
 - Cor: var(--cor-fonte-principal)
 - Cor de Fundo: var(--cor-fundo-principal)
- Imagem (img):
 - Largura: 40%
- Bloco de Cabeçalho (header):
 - Coordenada Superior: 0
 - Altura: 4rem
 - Largura: 100%
 - Posição: fixed
 - Disposição do Elemento: flex
 - Espaçamento entre os Itens: center
 - Alinhamento dos Itens: center
 - Cor de Fundo: var(--cor-fundo-secundaria)
 - Ordem de Exibição: 1
- Bloco Principal (main):
 - Coordenada Superior: 4rem
 - Altura: 100%
 - Largura: 100%
 - Posição: absolute
- Classe "container":
 - Altura Máxima: 200%
 - Largura: 100%
 - Margens: 1rem auto
 - Disposição do Elemento: grid
 - Lacuna entre Linhas e Colunas: 1rem
 - Efeito de Preenchimento: 0.25rem
 - Alinhamento dos Itens: center
 - Cor: var(--cor-fonte-secundaria)
 - Cor de Fundo: var(--cor-fundo-secundaria)
- Classes "superior" e "inferior":
 - Quantidade e Largura das Colunas: 1fr
- Campo (input), Lista de Seleção (select) e Opções (option):
 - Cor: inherit (herdar)
 - Nome da Fonte: inherit (herdar)
 - Espessura da Fonte: inherit (herdar)
 - Alinhamento do Texto: inherit (herdar)
 - Margens: 8px 5px

/* PARTE NOVA: */

- Classe "figuras":
Largura: 15%
Cursor: pointer
- Identificador "div_adiciona_figura":
Disposição do Elemento: flex
Espaçamento entre os Itens: center
Alinhamento dos Itens: center
Quebrar para a próxima linha: wrap
- Classe "novo_input":
Largura: 60%
Margem Superior: 0.5rem
- Tabela:
Margens: auto
Tamanho da Fonte: 0.75rem
Largura: 90%
- Campo da Tabela (td):
Borda: 1px dashed var(--cor-fonte-principal)
- Legenda da Tabela (caption):
Cor: var(--cor-fonte-principal)
- Campos dentro da tabela com Identificador "tab_principal" e adicionar os itens que estão dentro da Lista Principal:
Cursor: pointer
- Adicionar Identificadores da Lista Principal e da Lista Vazia:
Posicionamento dos Itens da Lista: inside
Tamanho da Fonte: 0.8rem
Efeito de Preenchimento: 0

Se a largura mínima da janela for **768px (MEDIA QUERY)**:

- Imagem (img):
Largura: 50%
- Bloco de Cabeçalho (header):
Tamanho da Fonte: 2rem
- Bloco Principal (main):
Altura: calc(100% - 4rem)
- Classe "container":
Altura: 38%
Largura: 90%
Margens: 2rem auto
- Classe "superior":
Quantidade e Largura das Colunas: repeat(6, 1fr)
- Classe "inferior":
Quantidade e Largura das Colunas: repeat(4, 1fr)

- **Codificação – Arquivo `jscript.js`:**

Mudar a Figura (Redes Sociais)

- 1º Criar os objetos para os elementos que serão manipulados:
 - Lista de Seleção com identificador `sel_rede_social`
 - Imagem com identificador `img_rede_social`
 - Parágrafo com identificador `p_rede_social`
- 2º Adicionar o evento para chamar a função:
 - Objeto: Lista de Seleção com identificador `sel_rede_social`
 - Evento: Alterar (change)
 - Função: `FunMudaFigura`
 - Parâmetros: sem parâmetros
- 3º Declarar a função para executar as ações:
 - Função: `FunMudaFigura`
 - Parâmetros: sem parâmetros
 - Codificação:
 - Adicionar ao objeto da Imagem com identificador `img_rede_social` o atributo "src", preenchendo neste atributo o caminho para o arquivo da imagem. Usar a opção selecionada na Lista de Seleção com identificador `sel_rede_social`;
 - Adicionar ao objeto da Imagem com identificador `img_rede_social` o atributo "alt", preenchendo neste atributo a opção selecionada na Lista de Seleção com identificador `sel_rede_social`;
 - Adicionar ao objeto da Imagem com identificador `img_rede_social` o atributo "title", preenchendo neste atributo a opção selecionada na Lista de Seleção com identificador `sel_rede_social`;
 - Preencher no objeto do parágrafo com identificador `p_rede_social` a opção selecionada na Lista de Seleção com identificador `sel_rede_social`.

Adicionar Figuras

- 1º Criar os objetos/arrays para os elementos que serão manipulados:
 - Array para as Imagens com a classe `figuras`
 - Objeto para o Bloco com identificador `div_adiciona_figura`
- 2º Criar novos objetos para serem manipulados:
 - Objeto para Imagem
 - Objeto para Campo (input)
- 3º Adicionar os eventos para chamar a função:
 - Objetos do Array com as Imagens com a classe `figuras`
 - Evento: Clicar (click)
 - Função: `FunAdicionaFigura`
 - Parâmetros: atributos src e alt em cada Imagem
- 4º Declarar a função para executar as ações:
 - Função: `FunAdicionaFigura`
 - Parâmetros: src e alt
 - Codificação:
 - Adicionar ao objeto da nova Imagem criada o atributo "src", preenchendo neste atributo o respectivo parâmetro recebido;
 - Adicionar ao objeto da nova Imagem criada o atributo "alt", preenchendo neste atributo o respectivo parâmetro recebido;
 - Adicionar ao objeto da nova Imagem criada o atributo "title", preenchendo neste atributo o parâmetro alt recebido;
 - Adicionar o objeto da nova Imagem criada ao objeto do Bloco com identificador `div_adiciona_figura`;
 - Adicionar ao objeto do novo Campo (input) criado o atributo "type", preenchendo neste atributo o valor "text";

- Adicionar ao objeto do novo Campo (input) criado o atributo "disabled", preenchendo neste atributo o valor "true";
- Adicionar ao objeto do novo Campo (input) criado o atributo "value", preenchendo neste atributo o parâmetro alt recebido;
- Adicionar ao objeto do novo Campo (input) criado a classe referente aos estilos do novo Campo (classe "novo_input");
- Adicionar o objeto do novo Campo (input) criado ao objeto do Bloco com identificador `div_adiciona_figura`.

Preencher Tabela

1º Criar os objetos/arrays para os elementos que serão manipulados:

Array para os campos/colunas da Tabela com o identificador `tab_principal`

Tabela com identificador `tab_vazia`

Caption da Tabela com identificador `tab_vazia`

Barra de Progressão com identificador `prog_tab_vazia`

2º Declarar variável para controlar a quantidade de linhas na tabela nova. Iniciar com ZERO.

3º Adicionar os eventos para chamar as funções:

Objetos do Array com os campos/colunas da Tabela com o identificador `tab_principal`

Evento: Clicar (click)

Função: `FunPreencheTabela`

Parâmetros: atributo `innerText` de cada campo/coluna

4º Declarar a função para executar as ações:

Função: `FunPreencheTabela`

Parâmetros: `innerText`

Codificação:

- Verificar, utilizando a variável que controla a quantidade de linhas na tabela, se a linha pode ser inserida na tabela (limite de 5 linhas);
- Se puder inserir:
 - * Criar novos objetos do tipo "tr" (linha) e "td" (campo/coluna);
 - * Preencher o texto do novo campo/coluna, usando o parâmetro recebido;
 - * Adicionar o objeto do novo campo/coluna ao objeto da nova linha;
 - * Adicionar o objeto da nova linha ao objeto com identificador `tab_vazia`;
 - * Incrementar a variável que controla a quantidade de linhas na tabela;
 - * Atualizar o objeto caption da tabela com identificador `tab_vazia` de acordo com a contagem de linhas na tabela;
 - * Atualizar o atributo value da barra de progressão com identificador `prog_tab_vazia` de acordo com a contagem de linhas na tabela.
- Se não puder inserir:
 - * Exibir mensagem de alerta.

Mudar a Figura (Linguagens de Programação Web)

- 1º Criar os objetos/arrays para os elementos que serão manipulados:
 - Array para os Radio Buttons com opções para selecionar a Linguagem (com mesmo name)
 - Objeto para a Imagem que exibirá o logo da Linguagem
 - Objeto para o Parágrafo Predefinido que exibirá a Linguagem selecionada
- 2º Adicionar os eventos para chamar a função:
 - Objetos do Array com os Radio Buttons com opções para selecionar a Linguagem
 - Evento: Clicar (click)
 - Função: **FunMudaImgLingProg**
 - Parâmetros: sem parâmetros
- 3º Declarar a função para executar as ações:
 - Função: **FunMudaImgLingProg**
 - Parâmetros: sem parâmetros
 - Codificação:
 - Percorrer o Array com os Radio Buttons com opções para selecionar a Linguagem e verificar qual foi selecionado;
 - Adicionar ao objeto da Imagem o atributo "src", preenchendo neste atributo o caminho para o arquivo da imagem. Usar a opção selecionada no Radio Button (atributo value, com o nome que identifica o arquivo da Imagem da Linguagem de Programação);
 - Adicionar ao objeto da Imagem o atributo "alt", preenchendo neste atributo a opção selecionada no Radio Button (atributo value, com o nome da Linguagem de Programação);
 - Adicionar ao objeto da Imagem o atributo "title", preenchendo neste atributo a opção selecionada no Radio Button (atributo value, com o nome da Linguagem de Programação);
 - Preencher no objeto do Parágrafo Predefinido a opção selecionada no Radio Button (atributo value, com o nome da Linguagem de Programação).

Preencher Lista

- 1º Criar os objetos/arrays para os elementos que serão manipulados:
 - Array para os itens da Lista Principal com os nomes das Linguagens de Programação
 - Objeto para a Lista Vazia
 - Objeto para o Parágrafo referente à Lista Vazia
 - Barra de Progressão
- 2º Declarar variável para controlar a quantidade de itens na lista nova. Iniciar com ZERO.
- 3º Adicionar os eventos para chamar as funções:
 - Objetos do Array com os itens da Lista Principal com os nomes das Linguagens de Programação
 - Evento: Clicar (click)
 - Função: **FunPreencheLista**
 - Parâmetros: atributo innerText de cada item
- 4º Declarar a função para executar as ações:
 - Função: **FunPreencheLista**
 - Parâmetros: innerText
 - Codificação:
 - Verificar, utilizando a variável que controla a quantidade de linhas na lista, se o item pode ser inserido na lista (limite de 6 itens);
 - Se puder inserir:
 - * Criar novo objeto do tipo "li" (item da lista);
 - * Preencher o texto do novo item, usando o parâmetro recebido;
 - * Adicionar o objeto do novo item ao objeto da Lista Vazia;
 - * Incrementar a variável que controla a quantidade de linhas na lista;
 - * Atualizar o objeto Parágrafo referente à Lista Vazia de acordo com a contagem de itens na lista.
 - * Atualizar o atributo value da barra de progressão de acordo com a contagem de linhas na lista.

- Se não puder inserir:
 - * Exibir mensagem de alerta.