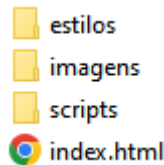


JS - JavaScript

- Criar uma página HTML, conforme Layout, utilizando CSS e JS:
- **Orientações:**
 - Criar uma pasta para esta atividade. Exemplo: "js03"
 - Dentro da pasta criada no passo anterior, criar a página principal **index.html**
 - Descompactar, na mesma pasta, o conteúdo do arquivo **imagens.zip**. Ao descompactar, será criada a pasta "imagens" com todas as figuras dentro;
 - Criar uma pasta, dentro da pasta desta atividade, chamada "**estilos**";
 - Criar, dentro da pasta "**estilos**", um arquivo para o CSS. A extensão do arquivo deve ser .css
 - Criar uma pasta, dentro da pasta desta atividade, chamada "**scripts**";
 - Criar, dentro da pasta "**scripts**", um arquivo para o JS. A extensão do arquivo deve ser .js
jscript.js
 - Desta forma, deverá ser utilizada uma estrutura de pastas, conforme apresentado abaixo:



- Esta estrutura de pastas não pode ser modificada.
- O Título do Navegador deve ser: "Atividade JavaScript";
- Incluir o link para o arquivo de Estilos:

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

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

```
<link rel="icon" href="imagens/javascript.png" />
```

- Adicionar a META TAG de "viewport" (se não houver);

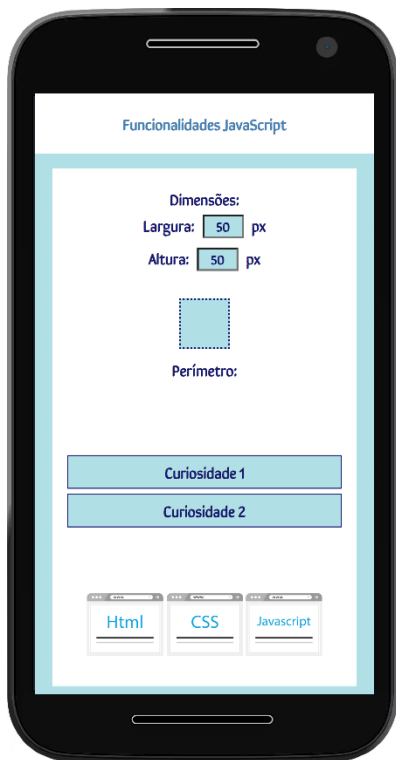
```
<meta name="viewport" content="width=device-width, initial-scale=1" />
```

- Incluir a referência para o arquivo JavaScript:

```
<script src="scripts/jscript.js" defer></script>
```

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** (classe: "**container**")
 - **Bloco**
 - Texto sem tag: Dimensões:
 - **Tag para pular uma linha**
 - Texto sem tag: Largura:
 - **Campo** tipo: "**number**"; identificador: "**num_largura**"; mínimo: 50; máximo: 100; valor: 50
 - Texto sem tag: px
 - **Tag para pular uma linha**
 - Texto sem tag: Altura:
 - **Campo** tipo: "**number**"; identificador: "**num_altura**"; mínimo: 50; máximo: 100; valor: 50
 - Texto sem tag: px
 - **Tag para pular uma linha**
 - **Tag para pular uma linha**
 - **Bloco**; identificador: "**div_poligono**"
 - *** VAZIO ***
 - **Parágrafo**; identificador: "**p_perimetro**"

Perímetro:
 - **Bloco** identificador: "**curiosidades**"
 - **Bloco** (classe "**titulo**")
 - Texto sem tag: Curiosidade 1
 - **Bloco** (classe "**texto**")
 - Texto sem tag: Em 2023, o JavaScript se destacou, pelo décimo primeiro ano consecutivo, como a linguagem de programação mais utilizada no mundo.
 - **Bloco** (classe "**titulo**")
 - Texto sem tag: Curiosidade 2
 - **Bloco** (classe "**texto**")
 - Texto sem tag: JavaScript é uma linguagem interpretada, o que significa que o código é executado diretamente pelo navegador, dispensando um compilador.
 - **Bloco** identificador: "**linguagens**"
 - **Imagem** "arq1.png"; texto alternativo: "HTML"
 - **Imagem** "arq2.png"; texto alternativo: "CSS"
 - **Imagem** "arq3.png"; texto alternativo: "JavaScript"

- Bloco identificador: "div_modal"
- Bloco (classe "conteudo_modal")
 - Parágrafo; identificador: "p_fechar":
 - Span: ×
 - Imagem identificador: "img_modal"; texto alternativo: ***VAZIO***
 - Parágrafo; identificador: "p_modal": ***VAZIO***

*** Fim Bloco Semântico (main) ****

- **Estilos/Propriedades CSS:**

- Incluir a origem da Fonte:

```
@import url('https://fonts.googleapis.com/css2?family=Sirin+Stencil&display=swap');
```

- **:root**
 - cor-fonte-principal: steelblue;
 - cor-fonte-secundaria: midnightblue;
 - cor-fundo-principal: powderblue;
 - cor-fundo-secundaria: white;
- **HTML e Corpo da Página (body):**
 - Margens: 0
 - Efeito de Preenchimento: 0
 - Nome da Fonte: 'Sirin Stencil', cursive
 - Espessura da Fonte: bold
 - Tamanho da Fonte: 1rem
 - Alinhamento do Texto: center
 - Cor: var(--cor-fonte-principal)
 - Cor de Fundo: var(--cor-fundo-principal)
- **Bloco de Cabeçalho (header):**
 - 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
 - Largura: 100%
 - Posição: absolute
- **Classe "container":**
 - Largura: 90%
 - Margens: 1rem auto
 - Disposição do Elemento: grid
 - Quantidade e Largura das Colunas: 1fr
 - Lacuna entre Linhas e Colunas: 0.5rem
 - Efeito de Preenchimento: 20px 0px
 - Alinhamento dos Itens: center
 - Cor: var(--cor-fonte-secundaria)
 - Cor de Fundo: var(--cor-fundo-secundaria)
- **Campo (input):**
 - Nome da Fonte: inherit (herdar)
 - Espessura da Fonte: inherit (herdar)
 - Alinhamento do Texto: inherit (herdar)
 - Margens: 5px
 - Cor de Fundo: var(--cor-fundo-principal)
 - Cor: var(--cor-fonte-secundaria)
- **Identificador "div_poligono":**
 - Altura: 50px
 - Largura: 50px
 - Disposição do Elemento: flex
 - Espaçamento entre os Itens: center
 - Alinhamento dos Itens: center

Margens: auto
Cor de Fundo: `var(--cor-fundo-principal)`
Borda: 2px dotted `var(--cor-fonte-secundaria)`
Tamanho da Fonte: 0.75rem

- Identificador "p_perimetro":
Margem Superior: 10px
- Identificador "curiosidades":
Largura: 90%
Margens: 50px auto
- Classe "titulo":
Largura: 100%
Efeito de Preenchimento: 5px
Margens: 5px auto
Borda: 1px solid `var(--cor-fonte-secundaria)`
Dimensionamento: border-box
Cor de Fundo: `var(--cor-fundo-principal)`
Cursor: pointer
- Classe "texto":
Disposição do Elemento: flex
Largura: 100%
Efeito de Preenchimento: 5px
Borda: 1px dashed `var(--cor-fonte-secundaria)`
Dimensionamento: border-box
Cor: `var(--cor-fonte-principal)`
Cor de Fundo: `var(--cor-fundo-secundaria)`
- Imagem dentro do Identificador "linguagens":
Largura: 25%
Cursor: pointer
- Identificador "div_modal":
Coordenada Superior: 0
Coordenada Esquerda: 0
Altura: 100%
Largura: 100%
Posição: fixed
Disposição do Elemento: none
Cor de Fundo: `rgba(0, 0, 0, 0.4)`
Ordem de Exibição: 1
- Classe "conteudo_modal":
Largura: 50%
Efeito de Preenchimento: 20px
Margens: 50% auto
Borda: 1px solid `var(--cor-fonte-principal)`
Cor de Fundo: `var(--cor-fundo-secundaria)`
- Imagem dentro da Classe "conteudo_modal":
Largura: 50%
- Identificador "p_fechar":
Margens: 0
Alinhamento do Texto: right
- Span dentro do Identificador "p_fechar":
Cor: `var(--cor-fonte-principal)`

- Ao passar o mouse no elemento Span dentro do Identificador "p_fechar":
Cor: `var(--cor-fonte-secundaria)`
Cursor: pointer

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

- Bloco de Cabeçalho (header):
Tamanho da Fonte: 2rem
- Classe "container":
Margens: 2rem auto
Quantidade e Largura das Colunas: 1fr 2fr 1fr
- Classe "conteudo_modal":
Largura: 20%
Margens: 10% auto

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

Redimensionar o Polígono

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

Campo (input) Numérico com identificador `num_largura`

Campo (input) Numérico com identificador `num_altura`

Bloco com identificador `div_poligono`

Parágrafo com identificador `p_perimetro`

2º Adicionar os eventos para chamar a função:

Objeto: Campo (input) Numérico com identificador `num_largura`

Evento: Alterar (change)

Função: `funRedimensionaPoligono`

Parâmetros: sem parâmetros

Objeto: Campo (input) Numérico com identificador `num_altura`

Evento: Alterar (change)

Função: `funRedimensionaPoligono`

Parâmetros: sem parâmetros

3º Declarar a função para executar as ações:

Função: `funRedimensionaPoligono`

Parâmetros: sem parâmetros

Codificação:

- Verificar se os valores dos Campos (inputs) Numéricos são válidos (entre 50 e 100);
- Se não for válido:
 - * Exibir mensagem de alerta;
- Se for válido:
 - * Atualizar a largura do Bloco com identificador `div_poligono`, usando o valor do objeto do Campo Numérico com identificador `num_largura`;
 - * Atualizar a altura do Bloco com identificador `div_poligono`, usando o valor do objeto do Campo Numérico com identificador `num_altura`;
 - * Calcular a área do Polígono, usando os valores dos objetos dos Campos Numéricos com identificadores `num_largura` e `num_altura`. Atribuir o resultado em uma variável;
 - * Preencher o resultado do cálculo da área (variável utilizada) no conteúdo (texto) do Bloco com identificador `div_poligono`;
 - * Calcular o perímetro do Polígono, usando os valores dos objetos dos Campos Numéricos com identificadores `num_largura` e `num_altura`. Atribuir o resultado em uma variável;
 - * Preencher o resultado do cálculo do perímetro (variável utilizada) no conteúdo (texto) do Parágrafo com identificador `p_perimetro`.

Curiosidades

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

Array com os Blocos com a Classe `titulo`

2º Adicionar o evento para chamar a função:

Objetos do Array com os Blocos com a Classe `titulo`

Evento: Clicar (click)

Função: `funExibeTexto`

Parâmetros: o próprio Bloco/Objeto que foi clicado (this)

3º Declarar a função para executar as ações:

Função: `funExibeTexto`

Parâmetros: o Bloco/Objeto que foi clicado (this)

Codificação:

- Declarar um objeto para armazenar o elemento "filho" do objeto recebido como parâmetro (nesse contexto, o único filho do bloco com a Classe `titulo` é o bloco com a Classe `texto`);
- Verificar se o objeto obtido no passo anterior está visível ou não, comparando o valor do atributo CSS "display". Se o valor do atributo for "flex", alterar para "none" (ocultar). Caso contrário, alterar para "flex" (exibir).

Modal com as Linguagens de Programação

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

Bloco com identificador `div_modal`

Tag Span dentro do Parágrafo com identificador `p_fechar`

Array com as Imagens que estão dentro do Bloco com identificador `linguagens`

Imagem com identificador `img_modal`

Parágrafo com identificador `p_modal`

2º Adicionar os eventos para chamar as funções:

Objetos do Array com as Imagens que estão dentro do Bloco com identificador `linguagens`

Evento: Clicar (click)

Função: `funExibeModal`

Parâmetros: atributos src e alt em cada Imagem

Tag Span dentro do Parágrafo com identificador `p_fechar`

Evento: Clicar (click)

Função: `funFechaModal`

Parâmetros:

3º Declarar a função para executar as ações:

Função: `funExibeModal`

Parâmetros: src e alt

Codificação:

- Adicionar ao objeto da Imagem com identificador `img_modal` o atributo "src", preenchendo neste atributo o respectivo parâmetro recebido;

- Adicionar ao objeto da Imagem com identificador `img_modal` o atributo "alt", preenchendo neste atributo o respectivo parâmetro recebido;

- Adicionar ao objeto da Imagem com identificador `img_modal` o atributo "title", preenchendo neste atributo o parâmetro alt recebido;

- Preencher o Parágrafo com identificador `p_modal` com o parâmetro alt recebido.

- Alterar o valor do atributo CSS "display" do Bloco com identificador `div_modal` para "block"(exibir).

Função: `funFechaModal`

Parâmetros: sem parâmetros

Codificação:

- Alterar o valor do atributo CSS "display" do Bloco com identificador `div_modal` para "none"(ocultar).