Estas anotações se referem as vídeo aulas oferecidas pela digital innovation one do curso de introdução a java script  
( https://web.digitalinnovation.one/course/programacao-para-internet-com-javascript/learning/33e81edf-6f9b-4b97-b3c6-12d930cf0831/ )

Lembrando que por se tratar de um curso que fiz após começar meus estudos em JavaScript, não irei anotar todo o conteúdo que já foi aprendido em outro lugar, apenas tentarei referenciar onde já existe para evitar redundâncias.

Primeiro passo de tudo se tratando de JavaScript é criar uma pagina HTML para poder rodar os scripts, no caso utilizei a mesma pagina ‘home.html’ que criei para usar os scripts do livro get programming with JS, apenas alterando o script que a pagina carregava, caso haja alguma duvida, consultar o outro .docx que fala sobre as anotações daquele livro

ALERT:

O ‘alert’ é um comando que cria uma janela de interação com o usuário que faz o display de alguma informação, no exemplo que segue ele mostra um texto:

var mensagem = 'Testando mensagem em alert';

alert(mensagem);

LISTA:

A maior parte do conhecimento sobre listas já está disponível no outro .docx sobre o tema.

Métodos:

lista.reverse():

Retorna a lista ao contrário;

Lista.toString():

Retorna a lista em forma de string com seus elementos separados por virgula, basicamente um .join(“,”);

// métodos de listas

var lista = ["gato", "sapato", "cadarço", "periquito", "salsicha"];

alert(lista.toString())

console.log(lista.reverse().toString())

DICIONARIO:

Dicionarios também já foram abordados no outro .docx.

LAÇOS:

If:

No exemplo que segue também mostramos o coneito de prompt, que é como se fosse um alert mas exige uma interação com o usuário em que ele precisa digitar algum valor.

//Exemplo de laço IF e comando Prompt() :

var idade = prompt("Qual sua idade?");

if (idade >= 18){

    alert("Maior de idade!");

}else{

    alert("Menor de idade!");

}

Loops:

Existem dois tipos de loops em JavaScript: loops for e loops while.

O ‘for’ como de costume repete o loop uma determinada quantidade de vezes e o ‘while’ até que sua condição seja falsa.

Aqui estão representadas as sintaxes:

//Loops

var countW = 0;

while (countW <= 3){

    console.log("contagem do loop 'while' : " + countW);

    countW ++;

}

for (var i = 0; i <=3; i++){

    console.log("contagem do loop 'for' : " + i)

}

DATE:

A função que trabalha com data no JavaScript é a que segue:

//Trabalhando com data

var d = new Date;

alert(d);

console.log(d.getMonth());

console.log(d.getDate());

console.log(d.getDay());

console.log(d.getHours());

console.log(d.getMilliseconds());

FUNÇÕES MANIPULANDO ELEMENTOS HTML:

Como já sabemos, é possível criar funções em JavaScript, algumas dela spodem alterar ou interagir com elementos da pagina HTML a qual está atrelada. No exemplo a seguir ao clicar no botão criado na pagina HTML é executada a função em JS ‘botao()’, veja:

Primeiro cria-se a função botão() em js:

//Funções

function botao(){

    alert("Obrigado por clicar!");

}

Então atrela-se o código na pagina HTML:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Aprendendo a programar.</title>

</head>

<script src="./introducaoAJS.js"></script>

<script></script>

<body>

    <h1>Olá Mundo!</h1>

    </br>

    Os codigos são atualizados todos os dias depois dos estudos no meu repositório do GitHub. ( https://github.com/leomac00/learningToCode )

    </br>

    <button type="button" onclick="botao()">Clique aqui!</button> <!-- Cria o botao -->

</body>

</html>

Entao ao clicar no botão na pagina de HTML o alert é executado.

Também podemos criar um botão com uma função que redireciona o usuário para uma pagina especifica, veja o código em js:

function redirecionar(){

window.open("https://github.com/leomac00/learningToCode")

}

Aqui está um exemplo de como uma função em JS altera um elemento de texto na pagina HTML, perceba que nesse exemplo usamos o ‘tihs’ como argumento da função trocaTexto() e da função voltaTexto() pois dessa maneira ele usa o id do objeto em questão para fazer as alterações necessárias, caso contrario, se passássemos o id do elemento na função em JS, apenas aquele elemento toda vez seria alterado.

Primeiro veremos como ficaria o exemplo com o ‘this’:

HTML:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Aprendendo a programar.</title>

</head>

<script src="./introducaoAJS.js"></script>

<script></script>

<body>

    <h1>Olá Mundo!</h1>

    </br>

    Os codigos são atualizados todos os dias depois dos estudos no meu repositório do GitHub. ( https://github.com/leomac00/learningToCode )

    </br>

    <button type="button" onclick="redirecionar()">GitHub</button>

    </br>

    </br>

    <button type="button" onclick="botao()">->Teste<-</button> <!-- Cria o botao -->

    <h3 id="agradecimento"></h3>

    </br>

    <h3 id="mouseMove" onmouseover="trocaTexto(this)" onmouseout="voltaTexto(this)">Passe o mouse aqui.</h3>

    <h3 id="mouseMove2" onmouseover="trocaTexto(this)" onmouseout="voltaTexto(this)">Passe o mouse aqui.</h3>

</body>

</html>

JS:

function trocaTexto(elemento){

    elemento.innerHTML = "Obrigado por passar o mouse aqui!";

}

function voltaTexto(elemento){

    elemento.innerHTML = "Passe o mouse aqui.";

}

Agora sem o ‘this’:

HTML:

    <h3 id="mouseMove" onmouseover="trocaTexto()" onmouseout="voltaTexto()">Passe o mouse aqui.</h3>

    <h3 id="mouseMove2" onmouseover="trocaTexto()" onmouseout="voltaTexto()">Passe o mouse aqui.</h3>

JS:

function trocaTexto(elemento){

    document.getElementById("mouseMove").innerHTML = "Obrigado por passar o mouse aqui!";

}

function voltaTexto(elemento){

    document.getElementById("mouseMove").innerHTML = "Passe o mouse aqui.";

}

Ai neste caso teríamos que criar mais duas funções (trocaTexto2() e voltaTExto2()) alerando o element ID para passar a referencia correta.

Em resumo, o ‘this’ como argumento da função a ser executada em JS no elemento HTML diz “use este elemento como argumento da função tal a ser executada”.

Também é possível alterar elementos de acordo com um menu dropdown:

Cria-se o menu na pagina HTML:

    <select onchange="change(this)">

        <option value="1">Valor 1</option>

        <option value="2">Valor 2</option>

        <option value="3">Valor 3</option>

    </select>

E então a função a ser executada em JS:

function change(elemento){

    console.log(elemento.value)

}

No caso a função faz algo bem simples mas poderia ser inserido algo mais complexo ali.