Muito melhor acompanhar hábitos do que acompanhar METAS.

Hard Skils – Habilidades técnicas e conceitos da área  
Soft Skils – Habilidades comportamentais (comunicação, resolução de problemas, gerenciamento de tempo e tarefas, trabalho em equipe...)

Variaveis

Uma caixinha onde guardamos um tipo de dado para usar mais tarde. E essa caixinha recebe todos os tipos de dados, que podemser e, textos, número, booleanos (valores lógicos: verdeiro ou falso) ou dados mais estuturados.

//Declaro e atribuo valor = let   
let boasVindas = “Fala, Dev!’;

//Reatribuo valor   
boasVindas = `Fala, Deve! Tudo beleza!?`;

// Constante não podemos mudar o valor   
const serHumano = true  
serHumano = false //Erro!

\*\* Funções

Agrupamento de código;  
 Reuso de código;  
 Mini programas dentro do programa maior;  
 Toda linguagem oferece muitas opções.

// usando uma função   
alert(‘Fala, Dev!’)

// criando uma função   
function alert(text) {  
 return text  
 }

\*\* Tudo são objetos

\*Atributos – são as propriedades de um objeto

\*Métodos – São as funcionalidades de um objeto

// Criando um objeto

const celular = {  
 cor: ‘preto’;  
 ligar: function () {}  
{

// Usando um objeto

celular.cor // preto  
celular.ligar() // executa a função

\*\* DOM – Document Object Model

É a modelagem dos elementos HTML em objeto JS.   
 Posso controlar minha página, meu documento HTML pelo JS através do objeto document.

Document // objeto que existe em todo navegador   
.querySelector(‘a’) // seleciona a tag a   
.click() // dá a ordem de clicar na tag a

#Aceleracao

// Uma variavel, não pode começar com números e não pode ter espaço, e estamos usando uma regra chamada camel case, ou seja, começa com a letra minúscula e quando temos um espaço, a próxima letra eu coloco maiúscula e não o espaço. Ex.: nlwSetup

Não posso utilizar caracteres especiais e também não posso utilizar palavras reservadas. O que são palavras reservadas?? São palavras que fazem parte da linguagem de programação, como: let, const, function...

\*\* Array é uma estrutura de dados, sempre ao termino da instrução dentro dos colchetes eu preciso colocar uma virgula para ir para a próxima linha.

button.addEventlistener(‘click’) – Estou adicionando um ouvidor de evento, essa função precisa de 2 argumentos. O primeiro, qual evento ele vai ouvir? Ele vai ouvir o click e o segundo argumento é uma função que iremos criar, chamada add que vai ouvir toda vez que ele fizer o click.

Lembrando que função serve para agrupar códigos para ser utilizada em qualquer momento.

Condicional = IF()  
A condicional é a logica de programação

Toda função que encontra a palavra return, para de rodar a aplicação

const form = document.querySelector('#form-habits')

const nlwSetup = new NLWSetup(form)

const button = document.querySelector('header button')

button.addEventListener('click',add)

function add() {

const today = "01/01"

const dayExists = nlwSetup.dayExists (today)

if(dayExists){

alert("Dia já incluso")

return

}

nlwSetup.addDay ("01/01")

}

Explicando o meu script.

Quando eu clicar no botão (button.addEventListener('click',add)) ele vai disparar a função que criamos.

A primeira linha da função, vai colocar uma data do dia 01/01 para uma variavel chamada today;

Na segunda linha, ele vai pegar o nlwSetup e verificar se o today, que é o dia 01/01 já existe, se não existe ele vai retornar um false e o valor vai ser false, a primeira vez que eu rodei a minha aplicação. Se ele é false, ele não entra no código da condicional IF. Porque o IF precisa de um valor true para rodar esse pedaço de código.

Por fim, se ele for false, ele pula o IF e vai para o addDay, adiciona o dia, retorna com o alerta “adicionado com sucesso” Clicando novamente no botão, ele vai rodar a aplicação toda até , entra na condicional IF por ser um true, encontra o alerta “dia já incluso”encontra o return e para de rodar o código.

Transformando uma data americanizada, para brasileiro pt-br

new Date().toLocaleDateString(‘pt-br’).slice(0, -5)  
19/01 – esse vai ser o retorno no meu console.log

Salvando alterações

form.addEventListener(“change”, save)

localStorage – É um objeto que guarda na memoria do borwser informações da página. Pra eu guardar essa informação eu preciso usar uma opção setItem que é uma funcionalidade e eu preciso declarar uma chave e um valor. A chave eu vou dar nome de NLWSetup@habits e o meu valor não pode ser um objeto como o nlwSetup.data é, eu preciso transformar esse nlwSetup.data em uma string.

Para transformar em uma string, vou acessar o meu devtools, usar o JSON.stringify(nlwSetup.data) e a impressão será uma string.

Carregando meus dados salvos

localStorage.getItem(‘key’) key é a chave que eu defini ali em cima **NLWSetup@habits** elas precisam ser exatamente iguais pois se houver algo errado ela não encontra a chave que eu quero. Após isso, ela retorna com uma string ou seja, um texto e eu preciso transformar ela em um objeto para colocar no setData.

Para isso eu vou usar o JSON.parse. A função parse me diz, me dá um texto que eu vou transformar ele em objeto.

Quando eu for rodar essa aplicação em outro ambiente, ele não vai ficar salvo. Pois o local storage é diferente do de origem e a aplicação vai estar limpa, sem nada preenchido e no devtools vai dar erro. Quando ele chega nessa linha de código const data = JSON.parse(localStorage.getItem("NLWSetup@habits")) ele não encontra dentro do local storage. Dentro do JS temos o “||” que é o “OU” do JS, ex.:  
Procura isso aqui const data = JSON.parse(localStorage.getItem("NLWSetup@habits")) || {} ele é vazio