LISTA DE ESTUDO (TO-DO-LIST)

Fez-se necessário o uso da biblioteca Day.js para formatação de datas e implementa uma lista de atividades interativa. As principais funcionalidades incluem a adição, atualização, remoção e marcação de atividades como concluídas, proporcionando uma interface com cores confortáveis ao usuário sem muita poluição visual dando ao usuário uma experiencia dinâmica e fácil de usar.

Day.js: Uma biblioteca de manipulação de datas usada para formatar as datas no código. Day.js é uma alternativa leve ao Moment.js.

- Esta função formata uma data fornecida em diferentes formatos (dia, semana, mês e hora) usando a biblioteca Day.js.

```
// bibliotecas e códigos de terceiros
const formatador = (data) => {
  return {
    dia: {
      numerico: dayjs(data).format('DD'),
      semana: {
        curto: dayjs(data).format('ddd'),
        longo: dayjs(data).format('dddd'),
      }
    },
    mes: dayjs(data).format('MMMM'),
    hora: dayjs(data).format('HH:mm')
  }
}
```

Lista de atividades:

Inicializa uma lista vazia de atividades.

```
// lista, array, vetor []
let atividades = []
```

Criação de Itens de Atividade:

Esta função cria um item de atividade em formato HTML, incluindo um checkbox para marcar a atividade como concluída, um botão de remoção e informações formatadas sobre a atividade.

```
// arrow function
const criarItemDeAtividade = (atividade) => {

  let input = `
  <input
  onchange="concluirAtividade(event)"
  value="${atividade.data}"
  type="checkbox"
  `

  if (atividade.finalizada) {
    input += 'checked'
  }

  input += '>'
```

Atualização da Lista de Atividades:

Atualiza o conteúdo da seção de atividades, re-renderizando todos os itens de atividades, ou exibindo uma mensagem se a lista estiver vazia.

```
const atualizarListaDeAtividades = () => {
  const section = document.querySelector('section')
  section.innerHTML = ''

  // verificar se a minha lista está vazia
  if (atividades.length == 0) {
    section.innerHTML = `<p>Nenhuma atividade cadastrada.</p>`
    return
  }

  for (let atividade of atividades) {
    section.innerHTML += criarItemDeAtividade(atividade)
  }
}
```

Remoção de Atividade:

Filtra a lista de atividades para remover a atividade correspondente à data fornecida e atualiza a lista exibida.

```
const removerAtividade = (data) => {
  atividades = atividades.filter((atividade) => atividade.data !== data)
  atualizarListaDeAtividades()
}

atualizarListaDeAtividades()
```

Salvar Nova Atividade:

Obtém os dados do formulário, cria uma nova atividade e verifica se a data e a hora estão disponíveis antes de adicionar a nova atividade à lista.

```
const salvarAtividade = (event) => {
  event.preventDefault()
  const dadosDoFormulario = new FormData(event.target)

  const nome = dadosDoFormulario.get('atividade')
  const dia = dadosDoFormulario.get('dia')
  const hora = dadosDoFormulario.get('hora')
  const data = `${dia} ${hora}`

  const novaAtividade = {
    nome,
    data,
    finalizada: false
  }

  const atividadeExiste = atividades.find((atividade) => {
    return atividade.data == novaAtividade.data
  })

  if (atividadeExiste) {
    return alert('Dia/Hora não disponível')
  }

  atividades = [novaAtividade, ...atividades]
  atualizarListaDeAtividades()
}
```

Criação de Seleção de Dias e Horas:

Gera uma lista de opções de dias e horas formatados e as adiciona ao elemento <select> correspondente no formulário.

```
const criarDiasSelecao = () => {
  const dias = [
    '2024-02-28',
    '2024-02-29',
    '2024-03-01',
    '2024-03-02',
    '2024-03-03',
  ]

  let diasSelecao = ''

  for (let dia of dias) {
    const formatar = formatador(dia)
    const diaFormatado = `
    ${formatar.dia.numerico} de
    ${formatar.mes}
    `
    diasSelecao += `
    <option value="${dia}">${diaFormatado}</option>
    `
  }

  document
    .querySelector('select[name="dia"]')
    .innerHTML = diasSelecao

}
criarDiasSelecao()


const criarHorasSelecao = () => {
  let horasDisponiveis = ''

  for (let i = 6; i < 23; i++) {
    const hora = String(i).padStart(2, '0')
    horasDisponiveis += `
    <option value="${hora}:00">${hora}:00</option>`
    horasDisponiveis += `
    <option value="${hora}:30">${hora}:30</option>`
  }

  document
    .querySelector('select[name="hora"]')
    .innerHTML = horasDisponiveis
}
criarHorasSelecao()
```

Concluir Atividade:

Altera o estado de conclusão de uma atividade quando o checkbox correspondente é marcado ou desmarcado, e atualiza a lista de atividades exibida.

```
const concluirAtividade = (event) => {
  const input = event.target
  const dataDesteInput = input.value

  const atividade = atividades.find((atividade) => {
    return atividade.data == dataDesteInput
  })

  if (!atividade) {
    return
  }

  atividade.finalizada = !atividade.finalizada
  atualizarListaDeAtividades()
}
```

Interação com o Usuário:

- O usuário pode adicionar novas atividades preenchendo um formulário com o nome da atividade, dia e hora.

- A lista de atividades é exibida na tela com informações detalhadas e opções de controle (marcar como concluída e remover).

- As atividades são formatadas de forma clara, com datas e horas apresentadas de maneira legível.

ORDENAÇÃO DE PALAVRAS POR FREQUENCIAS DE CARACTERES

Recebe os parâmetros necessários:

- Uma lista de palavras.

- Um caractere específico.

```
script.py > ...
1
2   def ordenar_por_frequencia_de_caractere(palavras, caractere):
```

Ordena a lista de palavras:

Utilizamos a função sorted para ordenar a lista de palavras. Para determinar a ordem, usamos uma função lambda como chave (key) que conta a frequência do caractere em cada palavra usando o método count. Adicionamos o argumento reverse=True para ordenar em ordem decrescente.

```
    return sorted(palavras, key=lambda palavra: palavra.count(caractere), reverse=True)
```

Funcionamento:

Para verificar se a função funciona corretamente, utilizamos um exemplo com uma lista de palavras e um caractere. A função é chamada com esses parâmetros, e a lista de palavras é ordenada e exibida conforme esperado.

```
script.py > ...
1
2   def ordenar_por_frequencia_de_caractere(palavras, caractere):
3       return sorted(palavras, key=lambda palavra: palavra.count(caractere), reverse=True)
4
5   palavras = ["Gama", "Matematica", "Vestibular", "IA"]
6   caractere = 'a'
7   palavras_ordenadas = ordenar_por_frequencia_de_caractere(palavras, caractere)
8   print(palavras_ordenadas)
9
```

Resultado:

Executando o arquivo no terminal por meio do comando python nomedoarquivo.py (python script.py) temos o resultado esperado: [‘Matemática’, ‘Gama’, Vestibular’, ‘IA’]

```
PS C:\Users\Moisés\workspace\ordenação-palavra> python script.py
['Matematica', 'Gama', 'Vestibular', 'IA']
PS C:\Users\Moisés\workspace\ordenação-palavra>
```