

برنامه لیست انجام کار

نام و نام خانوادگی: مهدی تجار
شماره دانشجویی: ۹۶۰۲۴۹۵۴۷
نام درس: اصول طراحی نرم افزار
نام استاد: دکتر رضا کارگر
زمان: پنجشنبه از ۱۷:۳۵ تا ۲۰:۲۰

زبان های استفاده شده

CSS - HTML - Java Script

توضیحات

برنامه لیست کارهای ساده برای ذخیره و مدیریت کارها و ذخیره آنها در حافظه محلی مرورگر وب است و این برنامه دارای یک طراحی رابط کاربری زیبا است که اضافه کردن کارها را انجام می دهد.

بررسی و توضیحاتی کوتاه در مورد کد های جاوا اسکریپت استفاده در برنامه

در ابتدا ۵ متغیر با `const` برای فیلد ورودی - دکمه اضافه کردن - لیست کارها - پیام های خطا و دکمه پاک کردن با استفاده از متد `querySelector` تعریف میکنیم

نکته: با استفاده از متد `querySelector()` میتوانیم به تگ های موجود در صفحه `html` دسترسی پیدا نماییم، در واقع همانند یک انتخابگر عمل میکند و همیشه به راحتی تگ ها، کلاس ها، آی دی ها، صفت ها و.. رو از صفحه `html` انتخاب کرد.

```
const task_input = document.querySelector('input');
const add_btn = document.querySelector('.add-task-button');
const todos_list = document.querySelector('.todos-list');
const alert_message = document.querySelector('.alert-message');
const delete_all_btn = document.querySelector('.delete-all-btn');
```

در خط بعدی سپس با `let` متغیری برای ذخیره لیست کارها ایجاد میکنیم که بتوان مقدار های لیست کارها را با متد `localStorage` و `JSON` در مرورگر ذخیره کرد.

```
let todos = JSON.parse(localStorage.getItem('todos')) || [];

window.addEventListener('DOMContentLoaded', showAllTodos);
```

در خط بعدی تابعی به نام `getRandomId` ایجاد میکنیم که کار این تابع ایجاد یک شناسه یکتا برای هر کار یا `todo` که در برنامه اضافه میکنیم به صورت تصادفی میدهد

```
function getRandomId() {
  return Math.random().toString(36).substring(2, 15) +
  Math.random().toString(36).substring(2, 15);
}
```

در خط بعدی تابعی به نام `addToDo` ایجاد میکنیم که کار این تابع ایجاد یک کار یا `todo` است.

```
function addToDo(task_input) {
  let task = {
    id: getRandomId(),
    task: task_input.value,
    completed: false
  }
  todos.push(task);
}
```

در خط بعدی با استفاده از `task_input` مقدار فیلد `todo` یا کار که کاربر وارد میکند را ذخیره میکنیم و در صفحه مقدار ها را به صورت لیست نمایش میدهیم

```
task_input.addEventListener('keyup', (e) => {
  if (e.keyCode === 13 && task_input.value.length > 0) {
    addToDo(task_input);
    saveToLocalStorage();
    task_input.value = '';
    showAllTodos();
  }
});
```

در خط بعدی با استفاده از `add_btn` چک میکنیم که اگر دکمه اضافه کردن کلیک شد و کاربر مقداری وارد نکرده بود خطا نمایش دهیم و در صورتی که کاربر مقدار وارد کرده بود با استفاده از `addToDo` مقدار را ذخیره میکنیم.

```
add_btn.addEventListener('click', () => {
  if (task_input.value === '') {
    showAlertMessage('لطفاً یک کار وارد کنید', 'error');
  } else {
    addToDo(task_input);
    saveToLocalStorage();
    showAllTodos();
    task_input.value = '';
    showAlertMessage('کار با موفقیت اضافه شد', 'success');
  }
});
```

در خط بعدی با استفاده از `delete_all_btn` تمامی مقدار کارها یا `todo` هایی که کاربر وارد کرده بود را حذف میکند که تابع `clearAllTodos` را فراخوانی می کند.

```
delete_all_btn.addEventListener('click', clearAllTodos);
```

در خط بعدی با استفاده از تابع `showAllTodos` تمامی مقدار کارها یا `todo` هایی که کاربر وارد کرده بود را به صورت لیست نمایش میدهد. همچنین در این تابع ۲ عدد دکمه اضافه کردیم تا کاربر بتواند کار یا `todo` مورد نظر را حذف یا ویرایش کند

```
function showAllTodos() {
  todos_list.innerHTML = '';
  todos.forEach((todo) => {
    todos_list.innerHTML += `
      <li class="todo-item" data-id="${todo.id}">
        <p class="task-body">
          ${todo.task}
        </p>
        <div class="todo-actions">
          <button class="btn btn-success"
onclick="editTodo('${todo.id}')">
            <i class="bx bx-edit-alt bx-sm"></i>
          </button>
          <button class="btn btn-error"
onclick="deleteTodo('${todo.id}')">
            <i class="bx bx-trash bx-sm"></i>
          </button>
        </div>
      </li>
    `;
  });
}
```

```

        </div>
      </li>
    `;
  });
}

```

در خط بعدی با استفاده از تابع `saveToLocalStorage` می‌توانیم مقدار هایی که کاربر وارد میکند را در مرورگر کاربر ذخیره کنیم، از این تابع قبل تر هم در کد `add_btn` استفاده شده.

```

function saveToLocalStorage() {
  localStorage.setItem('todos', JSON.stringify(todos));
}

```

در خط بعدی با استفاده از تابع `showAlertMessage` می‌توانیم پیغام های خطا یا موفقیت آمیز را در یک بلوک `Div` به صورت `HTML` به کاربر نمایش دهیم

```

function showAlertMessage(message, type) {
  let alert_box = `
    <div class="alert alert-${type} shadow-lg mb-5 w-full">
      <div>
        <span>
          ${message}
        </span>
      </div>
    </div>
  `;

  alert_message.innerHTML = alert_box;
  alert_message.classList.remove('hide');
  alert_message.classList.add('show');
  setTimeout(() => {
    alert_message.classList.remove('show');
    alert_message.classList.add('hide');
  }, 3000);
}

```

در خط بعدی با استفاده از تابع `deleteToDo` می‌توانیم مقدار هایی که کاربر وارد کرده را حذف کنیم، از این تابع در دکمه حذف لیست کارها استفاده شده و فراخوانی می‌شود.

```

function deleteToDo(id) {
  todos = todos.filter(todo => todo.id !== id);
  saveToLocalStorage();
  showAlertMessage('کار با موفقیت حذف شد', 'success');
  showAllTodos();
}

```

در خط بعدی با استفاده از تابع `editTodo` می‌توانیم مقدار هایی که کاربر وارد کرده را ویرایش کند، از این تابع در دکمه ویرایش لیست کارها استفاده شده و فراخوانی می‌شود.

```
function editTodo(id) {
  let todo = todos.find(todo => todo.id === id);
  task_input.value = todo.task;
  todos = todos.filter(todo => todo.id !== id);
  add_btn.innerHTML = "<i class='bx bx-check bx-sm'></i>";
  saveToLocalStorage();
  add_btn.addEventListener('click', () => {
    add_btn.innerHTML = "<i class='bx bx-plus bx-sm'></i>";
    showAlertMessage('کار مورد نظر با موفقیت به روز شد', 'success');
  });
}
```

در خط بعدی با استفاده از تابع `clearAllTodos` می‌توانیم تمام مقدار هایی که کاربر وارد کرده را حذف کند، از این تابع در متغیر `delete_all_btn` استفاده شده و فراخوانی می‌شود.

```
function clearAllTodos() {
  if(todos.length > 0) {
    todos = [];
    saveToLocalStorage();
    showAlertMessage('همه کارها با موفقیت پاک شد', 'success');
    showAllTodos();
  } else {
    showAlertMessage('هیچ کاری برای پاک کردن وجود ندارد', 'error');
  }
}
```

با تشکر