

김민석

# 자바스크립트 학습 포트폴리오

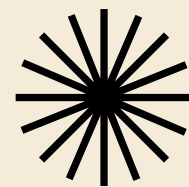
Javascript

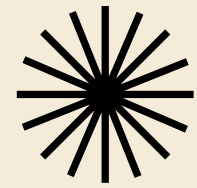
# 자기소개 & 목표

- 목표: 자바스크립트 문법을 실제로 구현하며 익히기
- 결과물: JS만으로 5개 미니 프로젝트 (To-Do, 명언, 슬라이더, 캘린더, 퀴즈)

# 전체 구조 & 역할

- HTML: 페이지 뼈대와 각 영역 구성
- CSS: 디자인/스타일 및 반응형
- JavaScript: 실제 동작(데이터/이벤트/화면 갱신)



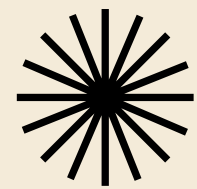


# To-Do 리스트 앱 (할 일 관리)

```
// 로컬스토리지에서 todos(할 일 목록) 불러오기, 없으면 빈 배열  
let todos = JSON.parse(localStorage.getItem('todos') || '[]');  
  
function renderTodos() {  
  todoList.innerHTML = ''; // 화면의 기존 리스트 비우기  
  todos.forEach((todo, i) => {  
    const li = document.createElement('li');  
    li.textContent = todo.text; // 할 일 내용 표시  
    li.className = todo.done ? 'done' : ''; // 완료 여부에 따라 스타일 적용  
    // 할 일 클릭 시 완료/미완료 토글  
    li.onclick = () => { todos[i].done = !todos[i].done; saveTodos(); }  
    const del = document.createElement('button');  
    del.textContent = '삭제';  
    // 삭제 버튼 클릭 시 해당 할 일만 삭제, 이벤트 전파 방지  
    del.onclick = (e) => { e.stopPropagation(); todos.splice(i, 1); saveTodos(); }  
    li.appendChild(del);  
    todoList.appendChild(li);  
  });  
}
```

• 기능:  
할 일  
추가/삭제/완료,  
새로고침에도  
데이터 유지

• 활용 JS:  
배열, 스토리지,  
이벤트 위임,  
함수,  
조건문/반복문

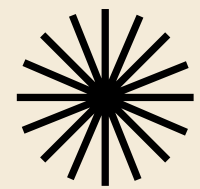


# 랜덤 명언 생성기

```
// 랜덤 명언 출력 함수
function showRandomQuote() {
  // 0~quotes.length-1 사이 랜덤 인덱스
  const idx = Math.floor(Math.random() * quotes.length);
  // id가 'quote-text'인 곳에 명언 넣기
  document.getElementById('quote-text').textContent = quotes[idx];
}
```

- 기능:  
버튼 클릭마다  
랜덤 명언 출력

- 활용 JS:  
배열,  
Math.random(),  
Math.floor()  
DOM 조작

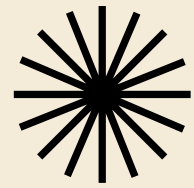


# 이미지 슬라이더

```
let slideIdx = 0; // 현재 이미지 인덱스(0부터 시작)
// 현재 인덱스의 이미지만 보이도록 하는 함수
function showSlide(idx) {
  slides.forEach((img, i) => img.classList.toggle('active', i === idx));
}
// 이전 버튼 클릭시
function prevSlide() {
  slideIdx = (slideIdx - 1 + slides.length) % slides.length;
  showSlide(slideIdx);
}
// 다음 버튼 클릭시
function nextSlide() {
  slideIdx = (slideIdx + 1) % slides.length;
  showSlide(slideIdx);
}
```

- 기능:  
여러 이미지를  
'이전/다음'으로  
넘김
- 활용 JS:  
배열 index,  
classList,  
순환(모듈로 연산)

 배열 index와 classList, 그리고 '모듈로 연산'을 써서 UI 순환 구조를 몸으로 익혔습니다.

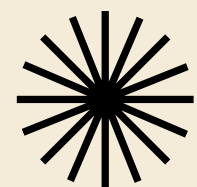


# 간단 캘린더

```
function renderCalendar() {  
  const today = new Date(); // 오늘 날짜  
  const year = today.getFullYear(),  
        month = today.getMonth();  
  const firstDay = new Date(year, month, 1).getDay(); // 1일의 요일(0~6)  
  const lastDate = new Date(year, month + 1, 0).getDate(); // 해당달 마지막일  
  let html = '<div class="calendar">';  
  // ...생략  
  // 날짜 칸 생성  
  for (let d = 1; d <= lastDate; d++) {  
    const cls = d === today.getDate() ? 'calendar-date today' : 'calendar-date';  
    html += `<div class="${cls}">${d}</div>`;  
  }  
  html += '</div>';  
  document.getElementById('calendar-box').innerHTML = html;  
}
```

• 기능:  
이번달 달력  
생성, 오늘 강조

• 활용 JS:  
Date 객체,  
반복문



# 간단 퀴즈

// 답 체크 함수

```
function checkAnswer() {  
  const userAns = document.getElementById('quiz-answer').value.trim().toLowerCase();  
  const correctAns = quizData[currentQuiz].a.toLowerCase();  
  if (userAns === correctAns) {  
    document.getElementById('quiz-result').textContent = '정답입니다!';  
  } else {  
    document.getElementById('quiz-result').textContent = '오답입니다. 정답: ' +  
quizData[currentQuiz].a;  
  }  
  currentQuiz = (currentQuiz + 1) % quizData.length; // 다음 문제로 순환  
  setTimeout(loadQuiz, 2000);  
}
```

• 기능:  
문제/정답 비교,  
실시간 피드백,  
문제 순환

• 활용 JS:  
객체 배열,  
반복문, 조건문



# 결론/느낀점

- 각 프로젝트를 직접 만들며 배열, 객체, DOM, 이벤트, 로컬스토리지, Date 등 JS의 핵심을 손으로 익힘
- 문제 상황에서는 console.log로 흐름을 추적하며 해결법을 스스로 터득함
- 실제로 구현해보며 JS로 '동적' 웹을 만드는 재미와 성장 실감을 느꼈음

