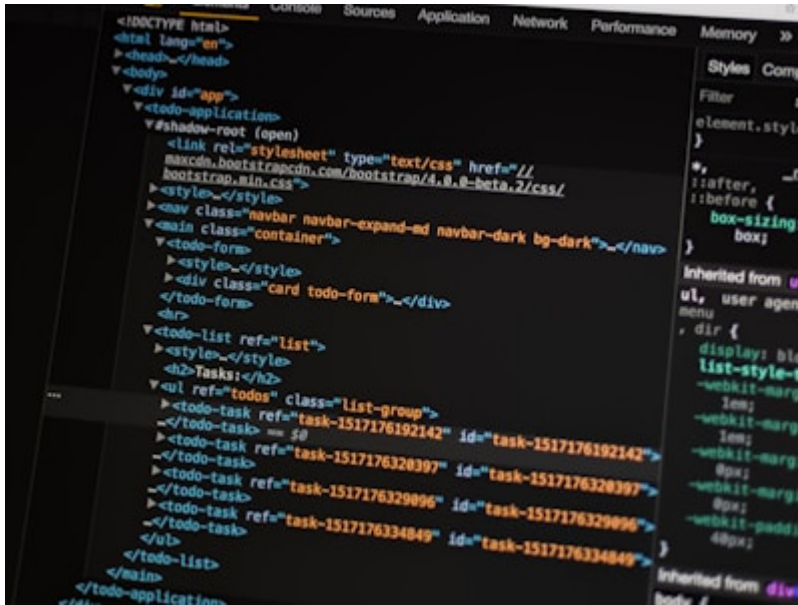


🌀 CSS 스타일링 및 반응형 디자인 (2부)

Flexbox와 반응형 디자인 마스터

강사명: 조성호

소요시간: 30분



📖 3-2교시 학습 목표

- Flexbox를 활용한 모던 레이아웃 구현
- 반응형 디자인으로 모든 기기 대응
- 미디어 쿼리를 통한 브레이크포인트 설정
- 프로필 페이지 완전한 스타일링 완성

🌀 3-1교시 복습

- CSS 기본 문법과 셀렉터
- 박스 모델과 단위 시스템
- CSS 우선순위와 상속
- 기본 스타일 적용 완료



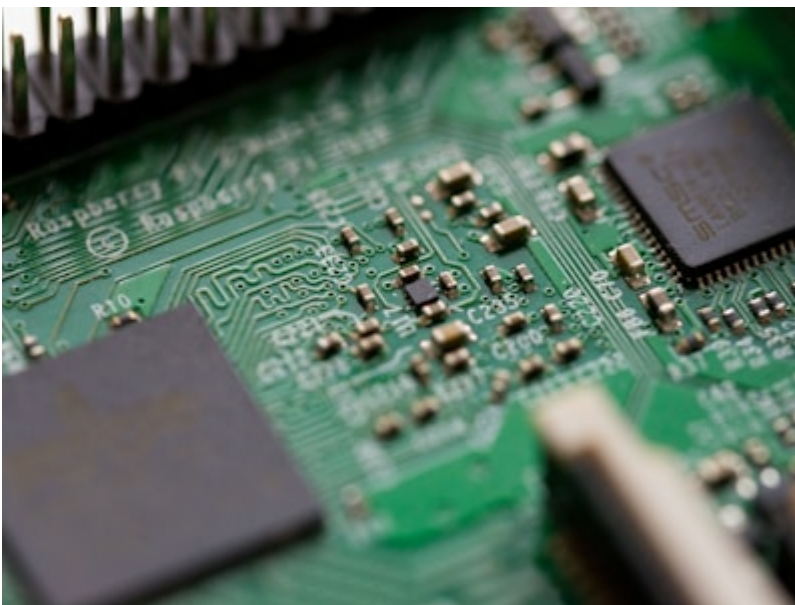
I 모던 레이아웃: Flexbox

Flexbox 기본 개념

Flexbox란?

- **Flexible Box Layout:** 유연한 박스 레이아웃
- **1차원 레이아웃 시스템** (가로 또는 세로)
- **정렬과 분배**가 쉬운 현대적 레이아웃

```
.flex-container {  
  display: flex; /* Flexbox 활성화 */  
}
```



Flexbox 주요 속성

컨테이너 속성

```
.flex-container {
  display: flex;

  /* 주축 방향 */
  flex-direction: row;      /* 가로 (기본값) */
  flex-direction: column;   /* 세로 */

  /* 주축 정렬 */
  justify-content: center;  /* 가운데 */
  justify-content: space-between; /* 양끝 정렬 */
  justify-content: space-around; /* 균등 분배 */

  /* 교차축 정렬 */
  align-items: center;      /* 가운데 */
  align-items: flex-start;  /* 시작점 */
  align-items: stretch;    /* 늘림 (기본값) */

  /* 줄바꿈 */
  flex-wrap: wrap;          /* 줄바꿈 허용 */
}
```

Flexbox 아이템 속성

```
.flex-item {
  flex: 1;      /* 균등 분할 */
  flex: 2;      /* 2배 크기 */
  flex-grow: 1; /* 늘어남 허용 */
  flex-shrink: 0; /* 줄어듦 방지 */
  flex-basis: 200px; /* 기본 크기 */

  align-self: center; /* 개별 정렬 */
}
```

실용적인 Flexbox 패턴

```
/* 완전 중앙 정렬 */
.center-both {
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
}

/* 네비게이션 메뉴 */
.nav-menu {
```

```

display: flex;
justify-content: space-between;
align-items: center;
}

```

☑ 반응형 디자인 기초

미디어 쿼리 기본 구조

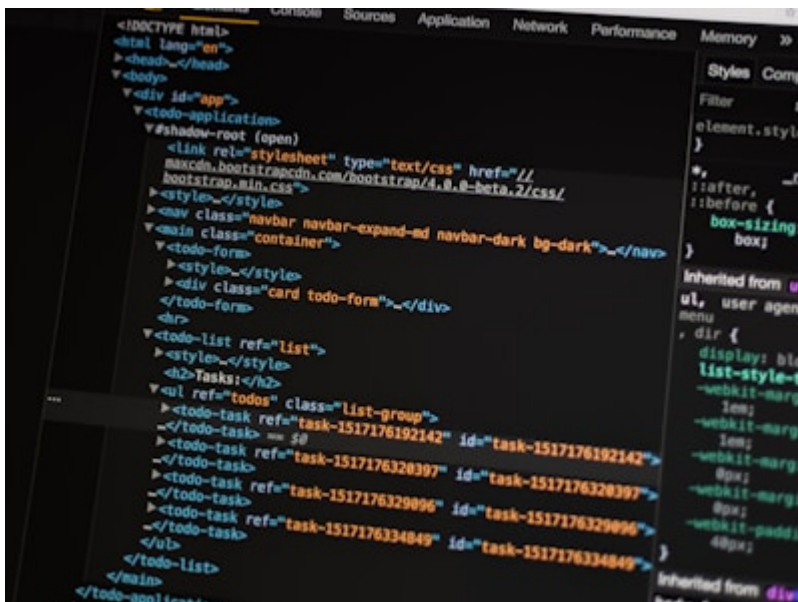
```

/* 기본 스타일 (모바일 퍼스트) */
.container {
  width: 100%;
  padding: 10px;
}

/* 태블릿 (768px 이상) */
@media (min-width: 768px) {
  .container {
    width: 750px;
    margin: 0 auto;
    padding: 20px;
  }
}

/* 데스크톱 (1024px 이상) */
@media (min-width: 1024px) {
  .container {
    width: 1000px;
    padding: 30px;
  }
}

```



반응형 브레이크포인트

일반적인 브레이크포인트

```
/* 모바일 (기본) */
@media (max-width: 576px) {
  .container { padding: 15px; }
}

/* 큰 모바일 */
@media (min-width: 577px) and (max-width: 768px) {
  .container { padding: 20px; }
}

/* 태블릿 */
@media (min-width: 769px) and (max-width: 1024px) {
  .container {
    width: 90%;
    margin: 0 auto;
  }
}

/* 데스크톱 */
@media (min-width: 1025px) {
  .container {
    width: 1000px;
    margin: 0 auto;
  }
}
```

프로필 페이지 완성 실습

Step 1: 네비게이션 Flexbox 적용

```
/* 네비게이션 스타일링 */
nav ul {
  display: flex;
  justify-content: center;
  align-items: center;
  list-style: none;
  gap: 30px;
  margin: 20px 0;
}

nav a {
  text-decoration: none;
}
```

```
    color: #2c3e50;
    font-weight: 500;
    padding: 10px 15px;
    border-radius: 5px;
    transition: all 0.3s ease;
}

nav a:hover {
    background-color: #3498db;
    color: white;
    transform: translateY(-2px);
}
```

Step 2: 섹션 카드 스타일링

```
/* 섹션 공통 스타일 */
section {
    background: white;
    margin: 30px 0;
    padding: 30px;
    border-radius: 15px;
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);
    transition: transform 0.3s ease;
}

section:hover {
    transform: translateY(-5px);
}

/* 제목 스타일 */
section h2 {
    color: #2c3e50;
    border-bottom: 3px solid #3498db;
    padding-bottom: 10px;
    margin-bottom: 20px;
}

/* 목록 스타일 */
section ul, section ol {
    padding-left: 20px;
}

section li {
    margin: 10px 0;
    line-height: 1.8;
}
```

Step 3: 반응형 레이아웃 구현

```
/* 모바일 기본 스타일 */
.container {
  width: 95%;
  margin: 0 auto;
  padding: 10px;
}

header h1 {
  font-size: 1.8rem;
}

/* 태블릿 스타일 */
@media (min-width: 768px) {
  .container {
    width: 90%;
    padding: 20px;
  }

  header h1 {
    font-size: 2.5rem;
  }

  /* 관심사와 목표를 2열로 배치 */
  .content-grid {
    display: flex;
    gap: 20px;
  }

  .content-grid section {
    flex: 1;
  }
}

/* 데스크톱 스타일 */
@media (min-width: 1024px) {
  .container {
    max-width: 1000px;
    padding: 40px;
  }

  header h1 {
    font-size: 3rem;
  }
}
```

Step 4: 고급 효과 추가

```
/* 그라데이션 배경 */
body {
```

```

    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    min-height: 100vh;
}

/* 카드 호버 효과 */
section {
    position: relative;
    overflow: hidden;
}

section::before {
    content: '';
    position: absolute;
    top: 0;
    left: -100%;
    width: 100%;
    height: 100%;
    background: linear-gradient(90deg, transparent, rgba(255,255,255,0.1),
transparent);
    transition: left 0.5s;
}

section:hover::before {
    left: 100%;
}

/* 버튼 스타일 */
.btn {
    display: inline-block;
    padding: 12px 24px;
    background: #3498db;
    color: white;
    border: none;
    border-radius: 25px;
    cursor: pointer;
    transition: all 0.3s ease;
}

.btn:hover {
    background: #2980b9;
    transform: scale(1.05);
}

```

Step 5: 완성된 CSS 구조

```

/* 1. 리셋 및 전역 설정 */
* { box-sizing: border-box; margin: 0; padding: 0; }

/* 2. 기본 타이포그래피 */
body { font-family: 'Arial', sans-serif; line-height: 1.6; }

```



```
/* 3. 레이아웃 컨테이너 */
.container { max-width: 1000px; margin: 0 auto; }

/* 4. 컴포넌트 스타일 */
header { /* 헤더 스타일 */ }
nav { /* 네비게이션 스타일 */ }
section { /* 섹션 스타일 */ }

/* 5. 반응형 미디어 쿼리 */
@media (min-width: 768px) { /* 태블릿 */ }
@media (min-width: 1024px) { /* 데스크톱 */ }
```

도전과제 미리보기

CSS 스타일링 마스터 챌린지

요구사항

1. 완전한 반응형 디자인 (모바일, 태블릿, 데스크톱)
2. 고급 레이아웃 기법 활용 (Flexbox, Grid)
3. 시각적 계층 구조 (색상, 타이포그래피, 간격)
4. 인터랙티브 요소 (호버, 애니메이션)
5. 최적화된 폼 디자인

추천 컬러 팔레트

```
:root {
  --primary: #667eea;
  --secondary: #764ba2;
  --accent: #f093fb;
  --light: #f8f9fa;
  --dark: #2c3e50;
}
```



개발자 도구 활용 팁

반응형 테스트

1. **F12** → **Device Mode** (Ctrl+Shift+M)
2. 다양한 기기 시뮬레이션
3. 실시간 브레이크포인트 확인

CSS 디버깅

1. **Elements** 탭에서 실시간 스타일 수정
2. **Computed** 탭에서 최종 적용 스타일 확인
3. **Console**에서 CSS 오류 확인

성능 최적화

1. **Lighthouse** 성능 측정
2. **Network** 탭에서 CSS 로딩 시간 확인

🙏 수고하셨습니다!

오늘 배운 내용 정리

- **Flexbox** 완전 마스터
- 반응형 디자인 구현 방법
- 미디어 쿼리 활용법
- 프로필 페이지 완성

다음 시간 예고: JavaScript 기초와 상호작용! 🖥️

과제: 개인 프로필 페이지에 창의적인 CSS 효과 추가하기

