

# Laboratoriya 8

## Mavzu: CSS Flexbox va Grid bilan sahifa tartibini yaratish

### Ishdan maqsad:

Ushbu ishda CSS Flexbox va CSS Grid texnologiyalarining veb-tartib yaratishdagi rolini, ularning bir o'lchovli (Flexbox) va ikki o'lchovli (Grid) imkoniyatlarini, afzalliklarini hamda ular birgalikda qanday samarali qo'llanilishini o'rganish

### Nazariy qism:

CSS Flexbox va CSS Grid veb-sahifalar uchun qulay va moslashuvchan tartib (layout) yaratishda muhim ahamiyatga ega. Bu ikki texnologiya zamonaviy veb-dizaynning asosiy qismlaridan biridir. Ularning har biri o'ziga xos afzalliklarga ega va turli vaziyatlarda qo'llaniladi.

CSS Flexbox (Flexible Box Layout) - Flexbox bir o'lchovli (bir qatorda yoki bir ustunda) elementlarni tartibga solish uchun mo'ljallangan. U qatorda joylashgan elementlarning o'lchamlarini moslashuvchan tarzda boshqarish imkonini beradi.

### Asosiy tushunchalar:

1. **Flex Container:** Elementlar guruhi joylashgan asosiy element. **display: flex;** yoki **display: inline-flex;** CSS xususiyati bilan belgilangan.
2. **Flex Items:** Flex Container ichidagi har bir bola elementi.
3. **Main Axis:** Flex Container ichida elementlar joylashadigan asosiy yo'nalish (odatda gorizonta).
4. **Cross Axis:** Main Axis ga perpendikulyar bo'lgan yo'nalish (odatda vertika).

### Flex Container Xususiyatlari:

#### 1. **display: flex | inline-flex;**

- 1) **flex:** Block-level flex container.
- 2) **inline-flex:** Inline-level flex container.

#### 2. **flex-direction: row | row-reverse | column | column-reverse;**

- 1) **row:** Elementlar qatorda chapdan o'ngga joylashadi (standart).
- 2) **row-reverse:** Elementlar qatorda o'ngdan chapga joylashadi.
- 3) **column:** Elementlar ustunda yuqoridan pastga joylashadi.
- 4) **column-reverse:** Elementlar ustunda pastdan yuqoriga joylashadi.

### **3. flex-wrap: nowrap | wrap | wrap-reverse;**

1) **nowrap:** Elementlar bir qatorda (yoki ustunda) joylashadi va sig'mayotganlar kesiladi (standart).

2) **wrap:** Elementlar sig'mayotganda keyingi qatorga (yoki ustunga) o'tadi.

3) **wrap-reverse:** Elementlar sig'mayotganda keyingi qatorda teskari tartibda joylashadi.

**4. justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly;** - elementlarni main axis bo'yicha joylashtiradi.

1) **flex-start:** Elementlar boshlanish nuqtasiga suriladi (standart).

2) **flex-end:** Elementlar oxirgi nuqtasiga suriladi.

3) **center:** Elementlar o'rta joylashtiriladi.

4) **space-between:** Birinchi element boshida, oxirgi element oxirida, qolganlari orasida teng masofada joylashadi.

5) **space-around:** Har bir element atrofida teng masofa qoldiriladi. Elementlar orasidagi masofa chetlardagi masofadan ikki barobar katta bo'ladi.

6) **space-evenly:** Barcha elementlar orasida va chetlarda teng masofa qoldiriladi.

**5. align-items: flex-start | flex-end | center | baseline | stretch;** - elementlarni cross axis bo'yicha joylashtiradi.

1) **flex-start:** Elementlar cross axis boshlanish nuqtasiga suriladi.

2) **flex-end:** Elementlar cross axis oxirgi nuqtasiga suriladi.

3) **center:** Elementlar cross axis o'rtasiga joylashtiriladi.

4) **baseline:** Elementlar o'zlarining matn asos chizig'i (baseline) bo'yicha tekislanadi.

5) **stretch:** Elementlar container sig'dirmaguncha cho'ziladi (standart).

**6. align-content: flex-start | flex-end | center | space-between | space-around | stretch;** - bu xususiyat faqat **flex-wrap** qiymati **wrap** yoki **wrap-reverse** bo'lgan holatlarda ishlaydi. U qatorlar (yoki ustunlar) orasidagi bo'sh joyni boshqaradi. ( **justify-content** ga o'xshash, lekin qatorlar uchun).

#### **Flex Item Xususiyatlari:**

**1. order: <integer>;** - elementlarning ko'rinish tartibini o'zgartirish imkonini beradi. Agar berilmasa, qiymati 0 bo'ladi.

**2. flex-grow: <number>;** - agar containerda bo'sh joy bo'lsa, element qancha kengayishini belgilaydi (standart 0). 1 dan katta qiymatlar elementni boshqalarga nisbatan ko'proq kengayishiga olib keladi.

**3. flex-shrink: <number>;** - agar containerda joy yetmasa, element qancha kichrayishini belgilaydi (standart 1). 0 qiymati elementni kichraytirmaydi.

**4. flex-basis: <length> | auto;** - elementning asosiy o'lchamini belgilaydi (width yoki height qarab). Agar auto bo'lsa, elementning o'z o'lchami ishlatiladi.

**5. flex: <flex-grow> <flex-shrink> <flex-basis>;** - flex-grow, flex-shrink, va flex-basis xususiyatlarini birlashtiruvchi qisqa yozuv. Masalan: flex: 1; (flex-grow: 1; flex-shrink: 1; flex-basis: 0%;)

flex: 1 0 auto; (flex-grow: 1; flex-shrink: 0; flex-basis: auto;)

**6. align-self: auto | flex-start | flex-end | center | baseline | stretch;** - individual flex itemning align-items xususiyatini o'zgartirish imkonini beradi. auto bo'lsa, containerning align-items qiymati ishlatiladi.

Flexbox quyidagi hollarda ishlatiladi:

- Bir o'lchovli tartiblar (navigatsiya panellari, ro'yxatlar, tugmalar qatori).
- Elementlarni bir qatorda yoki bir ustunda teng taqsimlash.
- Elementlarni vertikal yoki gorizontal o'rta joylashtirish.
- Elementlarni moslashuvchan tarzda saralash va tartibini o'zgartirish.

#### **CSS Grid (Grid Layout)**

CSS Grid ikki o'lchovli (qatorlar va ustunlar) tartibni boshqarish uchun mo'ljallangan. U murakkab va kattaroq hajmdagi sahifa tartiblarini yaratish uchun juda mos keladi.

### **Asosiy Tushunchalar:**

**1. Grid Container:** Elementlar guruhi joylashgan asosiy element. `display: grid;` yoki `display: inline-grid;` CSS xususiyati bilan belgilangan.

**2. Grid Items:** Grid Container ichidagi har bir bola elementi.

**3. Grid Lines:** Qatorlar va ustunlar chegaralarini belgilovchi gorizontal va vertikal chiziqlar.

**4. Grid Tracks:** Qatorlar (rows) va ustunlar (columns) orasidagi joy.

**5. Grid Cells:** Bir qator va bir ustun kesishgan joy.

**6. Grid Areas:** Bir nechta hujayralarni egallagan aniqlangan hudud.

Grid Container Xususiyatlari:

**1. display: grid | inline-grid;**

1) grid: Block-level grid container.

2) inline-grid: Inline-level grid container.

**2. grid-template-columns: <track-list> | auto | max-content | min-content | 1fr;**

1) Ustunlar tuzilmasini belgilaydi. <track-list> bo'sh joy bilan ajratilgan qiymatlardan iborat (masalan, 100px 1fr 50%).

2) fr birligi "fraction" (ulush) degan ma'noni bildiradi va containerda qolgan bo'sh joyni ulushlar bo'yicha taqsimlaydi.

**3. grid-template-rows: <track-list> | auto | max-content | min-content | 1fr; -** qatorlar tuzilmasini belgilaydi.

**4. grid-template-areas: <string-list>; -** grid hududlarini nomlash va ularni vizual tarzda tuzish imkonini beradi. Har bir string bir qatorni, vergullar esa hujayralarni bildiradi.

**5. grid-gap: <row-gap> <column-gap>; (Yoki gap, row-gap, column-gap) -** grid Tracks orasidagi bo'sh joyni belgilaydi.

**6. justify-items: start | end | center | stretch;** - grid itemlarni ularning grid hujayralarida gorizontal yo'nalishda (column axis) joylashtiradi.

**7. align-items: start | end | center | stretch;** - grid itemlarni ularning grid hujayralarida vertikal yo'nalishda (row axis) joylashtiradi.

**8. justify-content: start | end | center | space-between | space-around | space-evenly;** - gridning o'zini container ichida gorizontal yo'nalishda (column axis) joylashtiradi, agar grid containerdan kichik bo'lsa.

**9. align-content: start | end | center | space-between | space-around | space-evenly;** - gridning o'zini container ichida vertikal yo'nalishda (row axis) joylashtiradi, agar grid containerdan kichik bo'lsa.

### **Grid Item Xususiyatlari:**

**1. grid-column-start: <line> | span <count>;** - grid item qaysi ustun chiziqdan boshlanishini belgilaydi. span <count> bilan u necha ustunni egallashini ham belgilash mumkin.

**2. grid-column-end: <line> | span <count>;** - grid item qaysi ustun chiziqda tugashini belgilaydi.

**3. grid-row-start: <line> | span <count>;** - grid item qaysi qator chiziqdan boshlanishini belgilaydi.

**4. grid-row-end: <line> | span <count>;** - grid item qaysi qator chiziqda tugashini belgilaydi.

**5. grid-column: <start-line> / <end-line>;** - grid-column-start va grid-column-end ni birlashtiruvchi qisqa yozuv.

**6. grid-row: <start-line> / <end-line>;** - grid-row-start va grid-row-end ni birlashtiruvchi qisqa yozuv.

**7. grid-area: <row-start> / <column-start> / <row-end> / <column-end>;** - grid itemni grid area orqali joylashtirish. Shuningdek, grid-template-areas da nomlangan hududga ham o'rnatish mumkin.

**8. justify-self: start | end | center | stretch;** - individual grid itemni uning grid hujayrasida gorizontal yo'nalishda joylashtiradi.

**9. align-self: start | end | center | stretch;** - individual grid itemni uning grid hujayrasida vertikal yo'nalishda joylashtiradi.

Grid quyidagi hollarda ishlatiladi:

- Murakkab va kattaroq sahifa tartiblari (header, footer, sidebar, main content).

- Ikki o'lchovli (qatorlar va ustunlar) elementlar joylashuvini aniq nazorat qilish.

- Elementlarni aniq qator va ustunlar bo'yicha tekislash.

- Nisbatan kenglik va balandliklarni fr birligi bilan samarali boshqarish.

**Flexbox:** Bir o'lchovli tartiblar uchun ajoyib. Agar siz elementlarni bir qatorda yoki bir ustunda joylashtirmoqchi bo'lsangiz va ularning moslashuvchanligini boshqarmoqchi bo'lsangiz, Flexbox yaxshi tanlovdir. U elementlar orasidagi bo'sh joyni taqsimlashda va ularni tekislashda juda kuchli.

**Grid:** Ikki o'lchovli tartiblar uchun mos. Agar siz qatorlar va ustunlardan iborat murakkab sahifa tuzilishini yaratmoqchi bo'lsangiz, Grid sizga katta nazoratni taqdim etadi. U sahifani aniq bo'laklarga bo'lish va elementlarni bu bo'laklarga joylashtirish uchun juda mos keladi.

Ko'pincha, eng yaxshi natijalarga erishish uchun ikkalasini birgalikda ishlatish tavsiya etiladi. Masalan, butun sahifaning umumiy tuzilmasini Grid bilan yaratib, keyin har bir qism (masalan, navigatsiya paneli yoki tugmalar guruhi) ichidagi elementlarni tartibga solish uchun Flexboxdan foydalanish mumkin.

Qisqacha qilib aytganda, flexbox- bir o'lchovli (qator yoki ustun); grid esa Ikki o'lchovli (qator va ustun) larda ishlatiladi.

**Amaliy qism:** Bu mavzu orqali kartochka ko'rinishi (Grid va Flexbox orqali): Kitoblar javonini shakillantiramiz yoki manbani o'ziga xos kartochka sifatida ko'rsatamiz. Bu kartochkada muqova rasmi, sarlavha, muallif, qisqacha tavsif va "O'qish" tugmasi bo'ladi.

**Index.html Kod:**

```

<!DOCTYPE html>
<html lang="uz">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Kitoblar</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1> Kitoblar Javoni </h1>
  <!-- GRID KONTEYNER -->
  <div class="books-grid">
    <!-- Kitob 1 -->
    <div class="card">
      
      <h2>O'tgan Kunlar</h2>
      <p class="author">Abdulla Qodiriy</p>
      <p class="description">XX asr boshlaridagi O'zbekiston hayoti haqida
roman.</p>
      <button>📖 O'qish</button>
    </div>
    <!-- Kitob 2 -->
    <div class="card">
      
      <h2>Sinchalak</h2>
      <p class="author">Abdulla Qahhor</p>
      <p class="description">Bolalar uchun ibratli hikoya. Mehnat va halollik
haqida.</p>
      <button>📖 O'qish</button>
    </div>
    <!-- Kitob 3 -->
    <div class="card">
      
      <h2>Kecha va Kunduz</h2>
      <p class="author">Cho'lpon</p>
      <p class="description">Yoshlarning hayoti va orzulari haqida ijtimoiy
roman.</p>
      <button>📖 O'qish</button>
    </div>
  </div>
</body>
</html>

```

## Style.css Kod:

```

/* Umumiy */
* {

```

```

    margin: 0;
    padding: 0;
    box-sizing: border-box;
}
body {
    font-family: Arial, sans-serif;
    background: linear-gradient(135deg, #a2d3ad 0%, #aee6d0 100%);
    min-height: 100vh;
    padding: 40px 20px;
}
/* Sarlavha */
h1 {
    text-align: center;
    color: white;
    margin-bottom: 40px;
    font-size: 2.5em;
}
/* ===== GRID LAYOUT ===== */
.books-grid {
    /* Grid yoqildi */
    display: grid;
    /* 3 ta ustun, teng kenglikda */
    grid-template-columns: repeat(3, 1fr);
    /* Kartochkalar orasidagi bo'shliq */
    gap: 30px;
    /* Markazda */
    max-width: 1200px;
    margin: 0 auto;
}
/* ===== KARTOCHKA (FLEXBOX) ===== */
.card {
    /* Flexbox yoqildi */
    display: flex;
    /* Vertikal yo'nalish */
    flex-direction: column;
    /* Dizayn */
    background: white;
    border-radius: 15px;
    padding: 20px;
    box-shadow: 0 5px 20px rgba(0, 0, 0, 0.2);
    transition: transform 0.3s;
}
/* Hover */
.card:hover {
    transform: translateY(-10px);
}
/* Rasm */
.card img {
    width: 100%;
    height: 280px;
    object-fit: cover;
}

```



```

        border-radius: 10px;
        margin-bottom: 15px;
    }
    /* Sarlavha */
    .card h2 {
        font-size: 1.3em;
        color: #2c3e50;
        margin-bottom: 8px;
    }
    /* Muallif */
    .author {
        color: #667eea;
        font-weight: bold;
        margin-bottom: 10px;
    }
    /* Tavsif */
    .description {
        color: #7f8c8d;
        line-height: 1.5;
        margin-bottom: 15px;
        /* Bo'sh joyni egallaydi */
        flex-grow: 1;
    }
    /* Tugma */
    .card button {
        width: 100%;
        padding: 12px;
        background: #4CAF50;
        color: white;
        border: none;
        border-radius: 8px;
        font-size: 1em;
        cursor: pointer;
        transition: background 0.3s;
    }
    .card button:hover {
        background: #45a049;
    }
    /* ===== RESPONSIVE ===== */
    /* Planshet */
    @media (max-width: 768px) {
        .books-grid {
            grid-template-columns: repeat(2, 1fr);
            gap: 20px;
        }

        h1 {
            font-size: 2em;
        }
    }
}

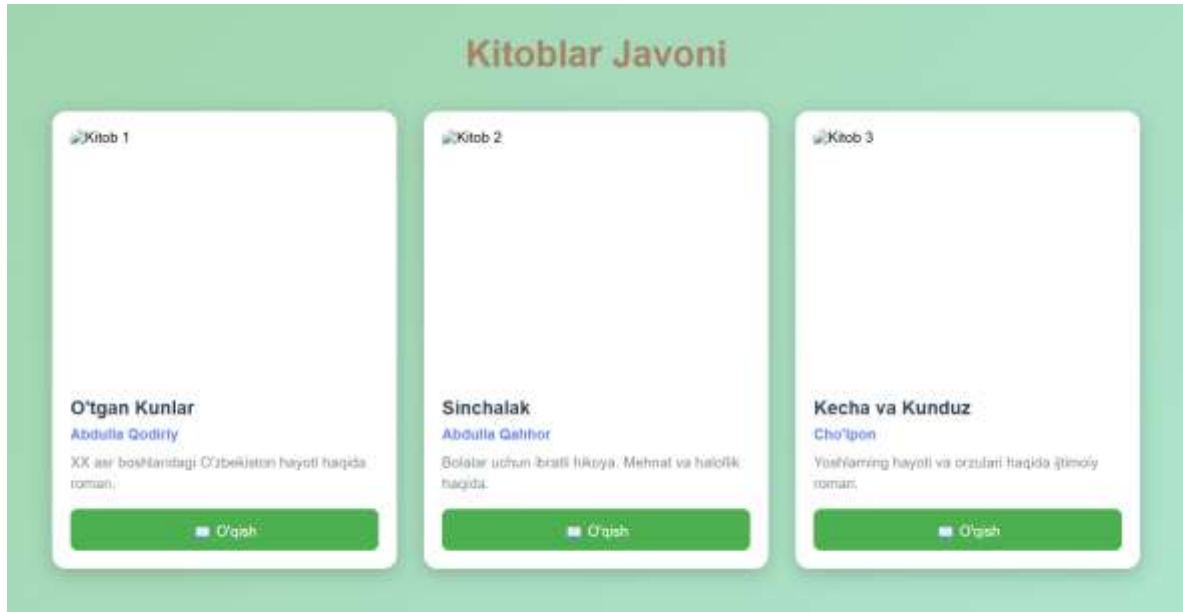
```

```

/* Mobil */
@media (max-width: 480px) {
  .books-grid {
    grid-template-columns: 1fr;
  }
  h1 {
    font-size: 1.5em;
  }
}

```

## Natija:



## Amaliy foyda:

1. Grid Layout - **Elementlarni panjara shaklida joylashtirish**
2. Flexbox Layout - **Moslashuvchan strukturalar**
3. Flex-grow - **Bo'shliqni boshqarish**
4. Responsive Design - **Har xil ekranlarda ishlash**
5. Grid va Flexbox ni birgalikda ishlatish - **Zamonaviy layoutlar.**

## TEXNIK XULOSA

### Ishlatilgan texnologiyalar:

#### HTML:

- Semantik struktura
- Class lardan foydalanish

#### CSS Grid:

- Ko'p ustunli layout
- Responsive dizayn

#### CSS Flexbox:

- Vertikal joylashtirish
- Flex-grow bilan bo'shliq boshqarish

#### CSS Animations:

- Hover effektlari

- Transform

**Media Queries:**

- Responsive dizayn
- Mobil moslashuvchanlik

**Yakuniy xulosa:**

**Proyekt maqsadi:** 3 ta kitobni chiroyli va professional ko'rinishda ko'rsatish

**Qanday amalga oshirildi:** HTML - strukturani yaratish, CSS Grid - 3 ustunli layout, CSS Flexbox - kartochka ichidagi tartib, Flex-grow - tugmalarni bir xil joyda ushlab turish, Responsive - mobilda ham ishlaydi.



yanada sifatli va zamonaviy qilishga yordam beradi.