

Topic 1: Bootstrap 12-Column Grid System

يعتمد على 12 عمود. هذا النظام يسمح لك بتقسيم الصف إلى أجزاء متساوية Bootstrap نظام الشبكة في

Subtopic 1

Understanding the 12-Column System

```
<!-- Each row has 12 columns total -->
<div class="row">
  <div class="col-6">Half width (6/12)</div>
  <div class="col-6">Half width (6/12)</div>
</div>

<div class="row">
  <div class="col-4">One third (4/12)</div>
  <div class="col-4">One third (4/12)</div>
  <div class="col-4">One third (4/12)</div>
</div>
```

يعني 4 أعمدة من أصل 12 (ثلث col-4). (يعني 6 أعمدة من أصل 12 (نصف العرض col-6). كل صف يحتوي على 12 عمود (العرض).

Subtopic 2

Responsive Breakpoints Explained

Breakpoint	Class Prefix	Screen Size	Example
Extra Small	col-	< 576px	col-12
Small	col-sm-	≥ 576px	col-sm-6
Medium	col-md-	≥ 768px	col-md-4
Large	col-lg-	≥ 992px	col-lg-3
Extra Large	col-xl-	≥ 1200px	col-xl-2

يطبق على الشاشات من الحجم prefix نقاط التوقف المتجاوبة تحدد كيف تظهر الأعمدة في أحجام الشاشات المختلفة. كل المحدد وما فوق.

```
<!-- Example from your demo -->
<div class="col-lg-2 col-md-4 col-sm-6">item1</div>
```

هذا العنصر سيظهر بـ 2 أعمدة في الشاشات الكبيرة، 4 أعمدة في الشاشات المتوسطة، و6 أعمدة في الشاشات الصغيرة.

Topic 2: Static vs Dynamic Grid Layouts

(والشبكات الديناميكية (المنشأة بالجافاسكريبت (HTML تعلم الفرق بين الشبكات الثابتة (في

Subtopic 1

Static Grid in HTML

```
<!-- Static grid with responsive classes -->
<div class="row">
  <div class="col-lg-2 col-md-4 col-sm-6">item1</div>
  <div class="col-lg-2 col-md-4 col-sm-6">item2</div>
  <div class="col-lg-2 col-md-4 col-sm-6">item3</div>
  <div class="col-lg-2 col-md-4 col-sm-6">item4</div>
  <div class="col-lg-2 col-md-4 col-sm-6">item5</div>
  <div class="col-lg-2 col-md-4 col-sm-6">item6</div>
</div>
```

للشاشات col-sm-6، للشاشات المتوسطة col-md-4، للشاشات الكبيرة col-lg-2: متجاوبة classes الشبكة الثابتة تستخدم الصغيرة.

Subtopic 2

Dynamic Grid with JavaScript

```
// Create container and title
const container = document.createElement('div');
container.classList.add('container', 'mt-4');

const title = document.createElement('h1');
title.classList.add('text-center', 'mb-4');
title.textContent = 'Grid Layout Demo (DOM Created)';
container.appendChild(title);
```

Bootstrap من classes إنشاء حاوية وعنوان باستخدام الجافاسكريبت وإضافة.

Topic 3: forEach Loop and Data Arrays

لإنشاء عناصر متعددة من مصفوفة بيانات forEach loop استخدام.

```
// Card data array - contains objects with card information
const cardData = [
  {
    title: 'Card 1',
    content: 'This is the first card created with DOM manipulation.',
    buttonText: 'Learn More',
    buttonClass: 'btn-primary',
    iconClass: 'bi-airplane'
  },
  {
    title: 'Card 2',
    content: 'This is the second card created with DOM manipulation.',
    buttonText: 'Learn More',
    buttonClass: 'btn-primary',
    iconClass: 'bi-airplane'
  },
  {
    title: 'Card 3',
    content: 'This is the third card created with DOM manipulation.',
    buttonText: 'Learn More',
    buttonClass: 'btn-primary',
    iconClass: 'bi-airplane'
  }
];
```

```
{
  title: 'Card 2',
  content: 'Second card built using createElement and classList.',
  buttonText: 'View Details',
  buttonClass: 'btn-success',
  iconClass: 'bi-balloon-fill'
}
// ... more cards
];

// forEach loop processes each object in the array
cardData.forEach(cardInfo => {
  // Create column with Bootstrap grid class
  const col = document.createElement('div');
  col.classList.add('col-3'); // 4 columns per row (12/3 = 4)

  // Create grid card
  const gridCard = document.createElement('div');
  gridCard.classList.add('grid-card');

  // Use object properties to create content
  const cardTitle = document.createElement('h3');
  cardTitle.textContent = cardInfo.title; // Access object property

  const cardParagraph = document.createElement('p');
  cardParagraph.textContent = cardInfo.content;

  const cardButton = document.createElement('button');
  cardButton.classList.add('btn', cardInfo.buttonClass);
  cardButton.textContent = cardInfo.buttonText;

  // Add event listener
  cardButton.addEventListener('click', function() {
    alert(`You clicked ${cardInfo.title}!`);
  });

  // Build the card structure
  gridCard.appendChild(cardTitle);
  gridCard.appendChild(cardParagraph);
  gridCard.appendChild(cardButton);

  // Add to column and row
  col.appendChild(gridCard);
  row.appendChild(col);
});
```

يحتوي على معلومات البطاقة. نستخدم object كل عنصر هو cardData. يمر على كل عنصر في مصفوفة forEach loop. ديناميكياً HTML هذه المعلومات لإنشاء عناصر

Subtopic 1

Understanding forEach Loop

```
// Basic forEach syntax
array.forEach(item => {
  // Code to run for each item
});

// In your code:
cardData.forEach(cardInfo => {
  // cardInfo is each object in the array
  // cardInfo.title, cardInfo.content, etc.
});
```

هو طريقة لتمرير على كل عنصر في مصفوفة. بدلاً من كتابة كود منفصل لكل بطاقة، نكتب كود واحد يعمل foreach لجميع البطاقات.

Topic 4: Advanced Bootstrap Grid System

متجاوبة متعددة classes مع Bootstrap استخدام نظام الشبكة المتقدم في.

```
// Create responsive columns dynamically
const col = document.createElement('div');
col.classList.add('col-3'); // 4 columns per row

// Create grid card with custom styling
const gridCard = document.createElement('div');
gridCard.classList.add('grid-card');
```

مخصص styling أعمدة في كل صف) وإضافة 4) class col-3 إنشاء أعمدة ديناميكية مع.

Subtopic 1

Responsive Breakpoints

- `col-lg-*`: Large screens (≥992px)
- `col-md-*`: Medium screens (≥768px)
- `col-sm-*`: Small screens (≥576px)
- `col-*`: Extra small screens (<576px)

نقاط التوقف المتجاوبة تحدد كيف تظهر الأعمدة في أحجام الشاشات المختلفة.

Topic 5: Interactive Grid Components

إنشاء مكونات تفاعلية في الشبكة مع أزرار وأيقونات.

```
// Card data array
const cardData = [
  {
```

```

        title: 'Card 1',
        content: 'This is the first card created with DOM manipulation.',
        buttonText: 'Learn More',
        buttonClass: 'btn-primary',
        iconClass: 'bi-airplane'
    }
    // ... more cards
];

// Create interactive button
const cardButton = document.createElement('button');
cardButton.classList.add('btn', cardInfo.buttonClass);
cardButton.textContent = cardInfo.buttonText;

// Add click event
cardButton.addEventListener('click', function() {
    alert(`You clicked ${cardInfo.title}!`);
});

```

للتفاعل event listeners وإضافة Bootstrap مختلفة من classes إنشاء أزرار تفاعلية مع.

Topic 6: Custom CSS Styling

وتصميم متجاوب hover مخصص للشبكة مع تأثيرات styling إضافة.

```

.grid-card {
    background: white;
    border: 2px solid #dee2e6;
    border-radius: 8px;
    padding: 20px;
    margin-bottom: 20px;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
    transition: all 0.3s ease;
    height: 200px;
    display: flex;
    flex-direction: column;
    justify-content: space-between;
}

.grid-card:hover {
    transform: translateY(-5px);
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.15);
    border-color: #007bff;
}

```

للترتيب الداخلي flexbox وتصميم hover مخصص للبطاقات مع تأثيرات CSS.

Practical Exercise

event متجاوبة مختلفة وأضف أزرار مع classes تمرين عملي: أنشئ شبكة ديناميكية تحتوي على 5 بطاقات تفاعلية. استخدم listeners.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Grid Exercise</title>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.css"
rel="stylesheet">
</head>
<body>
  <script>
    // Your dynamic grid code here
    // Create container, row, and 5 interactive cards
  </script>
</body>
</html>
```

Key Concepts Covered

المفاهيم الرئيسية التي تم تغطيتها في هذه الجلسة:

1. **12-Column Grid System:** Understanding Bootstrap's grid foundation
2. **Responsive Breakpoints:** col, col-sm, col-md, col-lg, col-xl
3. **Screen Sizes:** How breakpoints work across different devices
4. **Static vs Dynamic Grids:** HTML vs JavaScript creation
5. **forEach Loop:** Processing arrays of objects
6. **DOM Manipulation:** Creating elements with createElement and classList
7. **Interactive Components:** Buttons with event listeners
8. **Data-Driven UI:** Using arrays to generate multiple components