

## Topic 1: Introduction to Responsive Design (مقدمة في التصميم المتجاوب)

شرح مفهوم التصميم المتجاوب وأهميته في تصميم المواقع الحديثة.

### What is Responsive Design?

- Definition and importance
- Mobile-first vs. Desktop-first approaches
- Core principles of responsive design

### The Viewport Meta Tag

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- Understanding viewport settings
- Common viewport meta tags

## Topic 2: CSS Media Queries (استعلامات الوسائط في CSS)

شرح كيفية استخدام استعلامات الوسائط لإنشاء تصاميم متجاوبة.

### Media Query Syntax

```
/* Basic syntax */
@media (max-width: 768px) {
  /* CSS rules */
}

/* With media type */
@media screen and (max-width: 768px) {
  /* CSS rules */
}

/* Multiple conditions */
@media (min-width: 576px) and (max-width: 992px) {
  /* CSS rules */
}

/* Common breakpoints */
/* Small devices (landscape phones, 576px and up) */
@media (min-width: 576px) { ... }

/* Medium devices (tablets, 768px and up) */
@media (min-width: 768px) { ... }

/* Large devices (desktops, 992px and up) */
@media (min-width: 992px) { ... }
```

```
/* Extra large devices (large desktops, 1200px and up) */
@media (min-width: 1200px) { ... }
```

## Media Features

- Width and height
- Orientation (portrait/landscape)
- Resolution
- Hover capability
- Prefers-color-scheme (dark/light mode)

## Topic 3: Responsive Units and Values (وحدات القياس المتجاوبة)

شرح وحدات القياس النسبية والمتجاوبة.

### Relative Units

- % - Percentage of parent element
- vw - Viewport width (1vw = 1% of viewport width)
- vh - Viewport height (1vh = 1% of viewport height)
- vmin - 1% of viewport's smaller dimension
- vmax - 1% of viewport's larger dimension
- rem - Relative to root element's font-size
- em - Relative to parent element's font-size

### Responsive Typography

```
html {
  font-size: 16px; /* Base font size */
}

h1 {
  font-size: 2rem; /* 32px */
}

p {
  font-size: 1rem; /* 16px */
  line-height: 1.5;
  max-width: 65ch; /* Optimal line length for readability */
}
```

### Fluid Typography with clamp()

```
h1 {
  font-size: clamp(1.5rem, 4vw + 1rem, 3rem);
}
```

```
}
```

## Topic 4: Responsive Images and Media (الصور والوسائط المتجاوبة)

كيفية جعل الصور والوسائط متجاوبة مع أحجام الشاشات المختلفة.

### Responsive Images

```
<!-- Basic responsive image -->


<!-- Picture element for art direction -->
<picture>
  <source media="(min-width: 1200px)" srcset="large.jpg">
  <source media="(min-width: 768px)" srcset="medium.jpg">
  
</picture>

<!-- Responsive background image -->
<div class="hero" style="background-image: url('hero.jpg');
                        background-size: cover;
                        background-position: center;
                        min-height: 50vh;">

</div>
```

### Responsive Iframes

```
<div class="video-container">
  <iframe width="560" height="315"
    src="https://www.youtube.com/embed/..."
    frameborder="0"
    allowfullscreen
    style="width: 100%; height: auto;">
  </iframe>
</div>

<style>
.video-container {
  position: relative;
  padding-bottom: 56.25%; /* 16:9 aspect ratio */
  height: 0;
  overflow: hidden;
}

.video-container iframe {
  position: absolute;
  top: 0;
  left: 0;
```

```
width: 100%;
height: 100%;
}
</style>
```

## Topic 5: Responsive Layout Patterns (أنماط التنسيق المتجاوب)

أنماط مختلفة لتصميم تنسيقات متجاوبة.

### Common Responsive Patterns

#### 1. Mostly Fluid

- Single column on mobile
- Multi-column on larger screens
- Uses max-width to prevent stretching

#### 2. Column Drop

- Stacks columns vertically on small screens
- Displays side by side on larger screens

#### 3. Layout Shifter

- Completely different layouts for different screen sizes
- Uses multiple breakpoints

#### 4. Off Canvas

- Navigation or sidebar hidden off-screen on mobile
- Toggleable with a menu button

### CSS Grid for Responsive Layouts

```
.grid-container {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: 1rem;
  padding: 1rem;
}

/* Responsive grid with media queries */
@media (min-width: 768px) {
  .grid-container {
    grid-template-columns: repeat(3, 1fr);
  }
}
```

### Flexbox for Responsive Navigation

```
.nav {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
  align-items: center;
}

.menu-toggle {
  display: none;
}

@media (max-width: 768px) {
  .nav {
    flex-direction: column;
    align-items: flex-start;
  }

  .menu {
    display: none;
    width: 100%;
  }

  .menu.active {
    display: flex;
    flex-direction: column;
  }

  .menu-toggle {
    display: block;
    align-self: flex-end;
  }
}
```

## Practical Exercises (تمارين عملية)

### Exercise 1: Create a Responsive Layout

1. Create a webpage with a header, navigation, main content, sidebar, and footer
2. Make it responsive using media queries
3. Implement a mobile menu that toggles on small screens

### Exercise 2: Responsive Typography

1. Create a blog post with headings and paragraphs
2. Use relative units (rem, em) for font sizes
3. Implement fluid typography using clamp()
4. Adjust line height and max-width for better readability

### Exercise 3: Responsive Image Gallery

1. Create a grid of images
2. Make the grid responsive using CSS Grid
3. Implement the picture element for art direction
4. Add hover effects that work on touch devices

---

## Homework (الواجب المنزلي)

### Project: Build a Responsive Portfolio Website

1. Create a responsive portfolio website with the following pages:
  - Home/About
  - Projects/Portfolio
  - Contact
2. Requirements:
  - Must be fully responsive (mobile, tablet, desktop)
  - Use CSS Grid and/or Flexbox for layouts
  - Implement responsive navigation
  - Include responsive images and typography
  - Use at least 3 media query breakpoints
  - Ensure good performance (optimize images, etc.)
  - Test on multiple devices and screen sizes

### Additional Tasks:

1. Implement a dark/light mode toggle using CSS custom properties and prefers-color-scheme
  2. Add print styles to make the content printer-friendly
  3. Implement a "skip to main content" link for better accessibility
- 

## Resources (الموارد)

### Articles & Tutorials

- [A Complete Guide to CSS Media Queries](#)
- [Responsive Design - MDN Web Docs](#)
- [The 100% Correct Way to Do CSS Breakpoints](#)
- [Responsive Images in CSS](#)

### Tools

- [Google Mobile-Friendly Test](#)
- [Responsive Design Checker](#)
- [Viewport Resizer](#)
- [Flexbox Froggy](#) (for practicing Flexbox)
- [CSS Grid Garden](#) (for practicing CSS Grid)

### Frameworks

- [Bootstrap](#)

- [Tailwind CSS](#)
  - [Bulma](#)
  - [Foundation](#)
- 

## Next Session Preview (معاينة الجلسة القادمة)

### CSS Transitions & Animations

- CSS transitions
- CSS animations
- Transformations
- Performance considerations
- Practical use cases

### Preparation

1. Review the basics of CSS selectors and properties
  2. Think about where animations could enhance user experience
  3. Explore existing CSS animation libraries
- 

## Summary (ملخص)

### Key Takeaways

1. Responsive design ensures your website works well on all devices
2. Media queries allow you to apply CSS rules based on device characteristics
3. Use relative units and fluid typography for flexible layouts
4. Optimize images and media for different screen sizes
5. Test your designs on multiple devices and screen sizes

### Common Pitfalls to Avoid

- Using fixed units (px) for everything
- Not testing on real devices
- Overcomplicating breakpoints
- Forgetting about touch targets on mobile
- Ignoring performance implications

### Best Practices

1. Mobile-first approach
2. Use relative units (rem, em, %, vw, vh)
3. Implement responsive images
4. Test early and often
5. Consider accessibility and performance