CSS Layout Basics

أساسيات تخطيط CSS

Table of Contents / جدول المحتويات

Introduction to CSS Layout

مقدمة في تخطيط CSS

CSS Layout is about controlling the position and size of elements on a web page. It's essential for creating well-structured, responsive designs.

Key Concepts / المفاهيم الرئيسية

- **Normal Flow**: The default layout mode where elements are displayed in the order they appear in the HTML.
- **Positioning Schemes**: Different ways to position elements (static, relative, absolute, fixed, sticky).
- **Display Types**: How elements are displayed (block, inline, inline-block, flex, grid, etc.).

Display Property

خاصية العرض

The `display` property specifies how an element is displayed.

Common Values / القيم الشائعة

```css
/* Block elements take full width */
.block {
  display: block;
}

/* Inline elements take only necessary width */
.inline {
  display: inline;
}
```

```css
/* Inline-block: inline with width/height */
.inline-block {
  display: inline-block;
}

/* Flex container */
.flex-container {
  display: flex;
}

/* Grid container */
.grid-container {
  display: grid;
}
```

Positioning

التحديد الموضعي

CSS provides several methods for positioning elements.

Position Values / قيم التحديد

```css
/* Default positioning */
.static {
  position: static;
}

/* Relative to normal position */
.relative {
  position: relative;
  top: 10px;
  left: 20px;
}

/* Absolute to nearest positioned ancestor */
.absolute {
  position: absolute;
  top: 0;
  right: 0;
}

/* Fixed viewport position */
.fixed {
  position: fixed;
  bottom: 20px;
  right: 20px;
}

/* Sticky positioning */
```

```css
.sticky {
  position: sticky;
  top: 0;
}
```

Floats

العوامات

Floats were traditionally used for layouts but are now primarily used for text wrapping around images.

Float Examples / أمثلة على العوامات

```css
.float-left {
  float: left;
  margin-right: 20px;
}

.float-right {
  float: right;
  margin-left: 20px;
}

.clearfix::after {
  content: "";
  display: table;
  clear: both;
}
```

Flexbox

فليكس بوكس

Flexbox is a one-dimensional layout method for laying out items in rows or columns.

Flex Container Properties / خصائص حاوية الفليكس

```css
.flex-container {
  display: flex;
  flex-direction: row; /* or column */
  justify-content: center; /* main axis */
  align-items: center; /* cross axis */
  flex-wrap: wrap;
  gap: 10px;
}
```

Flex Item Properties / خصائص عناصر الفليكس

```
.flex-item {
  flex: 1; /* grow/shrink/basis */
  order: 1; /* reorder items */
  align-self: flex-start; /* override container's align-items */
}
```

Grid

الشبكة

CSS Grid is a two-dimensional layout system for the web.

Grid Container / حاوية الشبكة

```
.grid-container {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-template-rows: 100px 200px;
  gap: 10px;
  grid-template-areas:
    "header header header"
    "sidebar main main";
}

.header { grid-area: header; }
.sidebar { grid-area: sidebar; }
.main { grid-area: main; }
```

Grid Items / عناصر الشبكة

```
.grid-item {
  grid-column: 1 / 3; /* start / end */
  grid-row: 1;
  grid-area: header; /* named area */
}
```

Practical Examples

أمثلة عملية

Holy Grail Layout / تخطيط الكأس المقدس

```
<div class="holy-grail">
  <header>Header</header>
  <div class="content">
```

```
    <nav>Navigation</nav>
    <main>Main Content</main>
    <aside>Sidebar</aside>
  </div>
  <footer>Footer</footer>
</div>
```

```
.holy-grail {
  display: grid;
  grid-template-rows: auto 1fr auto;
  min-height: 100vh;
}

.content {
  display: grid;
  grid-template-columns: 200px 1fr 200px;
  gap: 1rem;
}

/* Responsive */
@media (max-width: 768px) {
  .content {
    grid-template-columns: 1fr;
  }

  nav, aside {
    order: 2;
  }
}
```

Homework

الواجب المنزلي

1. Create a responsive navigation bar that switches between horizontal and mobile layouts.
2. Build a photo gallery using CSS Grid with at least 9 images.
3. Create a pricing table using Flexbox with 3 columns that stack on mobile.
4. Implement a sticky header that stays at the top when scrolling.

Resources

الموارد

- MDN CSS Layout
- CSS-Tricks Guide to Flexbox
- CSS Grid Garden
- Flexbox Froggy

Next Session Preview

معاينة الجلسة القادمة

In the next session, we'll dive into **Responsive Design and Media Queries**, learning how to create websites that work beautifully on all devices.

Note: This document follows the template structure from Session 1 and includes both English and Arabic content as per the documentation standards.