

A. MỤC TIÊU:

- Thiết kế được giao diện web kết hợp HTML và CSS.
- Giới thiệu Link; Áp dụng Link thiết kế trang web
- Giới thiệu Z-Index; Áp dụng Z-Index thiết kế trang web
- Giới thiệu Grid; Áp dụng Grid thiết kế trang web

B. NỘI DUNG THỰC HÀNH

1. Cơ sở lý thuyết

1.1. Kiến thức cần nhớ

3.3. Định dạng trang web với CSS

3.3.1. Màu sắc và nền

3.3.2. Layout – float và clear

3.3.3. Xử lý văn bản

3.3.4. Viên, Khoảng đệm, Lề

3.3.5. Box Shadow

3.3.6. Transform

3.3.7. Filter, Display

3.3.8. Position

3.3.9. Animation

3.3.10. Link

Trong CSS, có thể sử dụng các thuộc tính để tạo kiểu cho các liên kết (links). Để định dạng liên kết, có thể sử dụng các pseudo-classes như :link, :visited, :hover, và :active. Dưới đây là cách sử dụng:

- :link: Áp dụng kiểu cho các liên kết chưa được truy cập.
- :visited: Áp dụng kiểu cho các liên kết đã được truy cập.
- :hover: Áp dụng kiểu khi con trỏ chuột hover lên liên kết.
- :active: Áp dụng kiểu khi liên kết đang được nhấn.

Ví dụ:

```

<head>
  <meta charset="utf-8" />
  <title>LINKS</title>
  <style>
    /* Định dạng cho liên kết chưa được truy cập */
    a:link {
      color: blue;
      text-decoration: none; /* Xóa gạch chân */
    }
    /* Định dạng cho liên kết đã được truy cập */
    a:visited {
      color: purple;
    }
    /* Định dạng khi hover lên liên kết */
    a:hover {
      color: red;
      text-decoration: underline; /* Gạch chân khi hover */
    }
    /* Định dạng khi liên kết đang được nhấn */
    a:active {
      color: green;
    }
  </style>
</head>
<body>
  <h2>Styling a link with color, text-decoration property</h2>
  <p><b><a href="#" target="_blank">This is a link</a></b></p>
</body>
</html>

```

3.3.11. Z-index

z-index trong CSS được sử dụng để điều khiển thứ tự chồng lấp (stacking order) của các phần tử trong không gian 2D. Nó xác định phần tử nào sẽ nằm trên hay dưới phần tử khác khi các phần tử có thể chồng lên nhau.

Cú pháp: `z-index: <giá trị>;`

Trong đó: Giá trị của z-index nhận một giá trị số, có thể là dương, âm hoặc 0. Giá trị càng cao, phần tử càng nằm trên các phần tử có z-index thấp hơn.

Một số điểm quan trọng:

- Chỉ hoạt động với các phần tử có **position khác static** (tức là có position: relative, absolute, fixed hoặc sticky). Nếu không, z-index sẽ không có tác dụng.
- Giá trị z-index:
 - Giá trị dương: Các phần tử có z-index dương sẽ nằm trên các phần tử có z-index bằng hoặc âm.
 - Giá trị âm: Các phần tử có z-index âm sẽ nằm dưới các phần tử có z-index dương hoặc 0.
 - Giá trị bằng 0: Các phần tử có z-index bằng 0 sẽ nằm ở mức độ "trung bình" so với các phần tử khác.

Ví dụ:

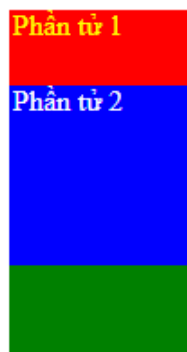
```
<head>
  <meta charset="utf-8" />
  <title>LINKS</title>
  <style>
    /* Phần tử đầu tiên với z-index 1 */
    .div1 {
      position: absolute;
      z-index: 1;
      background-color: red;
      width: 100px;
      height: 100px;
      color: yellow;
      padding-left: 2px;
    }
    /* Phần tử thứ hai với z-index 2 (nằm trên phần tử
đầu tiên) */
    .div2 {
      position: absolute;
      z-index: 2;
      background-color: blue;
      width: 100px;
      height: 100px;
      top: 50px;
      color: white;
      padding-left: 2px;
```

```

    }
    /* Phần tử thứ ba với z-index -1 (nằm dưới cả hai
    phần tử trên) */
    .div3 {
        position: absolute;
        z-index: -1;
        background-color: green;
        width: 100px;
        height: 100px;
        top: 100px;
        padding-left: 2px;
    }
</style>
</head>
<body>
    <div class="div1">Phần tử 1</div>
    <div class="div2">Phần tử 2</div>
    <div class="div3">Phần tử 3</div>
</body>

```

Kết quả:



3.3.12. Grid

CSS Grid là một hệ thống bố cục mạnh mẽ giúp sắp xếp các phần tử trên trang web theo dạng lưới (grid), cho phép dễ dàng tạo các bố cục phức tạp mà không cần dùng quá nhiều float hay flexbox.

Để tạo một Grid, đặt thuộc tính `display: grid;` hoặc `display: inline-grid;` cho phần tử cha.

Các khái niệm cơ bản trong CSS Grid:

- Grid Container: Là phần tử cha chứa các phần tử con (grid items).
- Grid Items: Là các phần tử con trực tiếp của grid container.
- Grid Lines: Là các đường ngang và dọc chia lưới thành các hàng và cột.

- Grid Tracks: Là khoảng cách giữa hai grid lines liên tiếp, tạo thành các hàng (row tracks) và cột (column tracks).
- Grid Cell: Là một đơn vị nhỏ nhất trong lưới, được tạo bởi giao điểm của một hàng và một cột.
- Grid Area: Là một vùng trong lưới, có thể bao gồm một hoặc nhiều grid cells.

Cú pháp cơ bản:

```
.container {
  display: grid;
  grid-template-columns: 100px 200px auto;
  grid-template-rows: 50px 100px;
  gap: 10px;
}
```

Trong đó:

- grid-template-columns: Xác định số cột và kích thước của mỗi cột.
 - 100px, 200px: Định kích thước cột cố định theo pixel.
 - auto: Cột tự điều chỉnh kích thước theo nội dung.

Hoặc:

```
grid-template-columns: 1fr 2fr 1fr;
/* fr là đơn vị tỷ lệ, giúp chia kích thước cột theo phần
trăm của tổng không gian */
```

- grid-template-rows: Xác định số hàng và kích thước của mỗi hàng.
- gap: Xác định khoảng cách giữa các hàng và cột (có thể sử dụng row-gap và column-gap riêng biệt).

🔖 Lưu ý:

- Định vị phần tử con:

```
.item {
  grid-column: 1 / 3; /* Chiếm từ cột 1 đến cột 3 */
  grid-row: 2 / 4; /* Chiếm từ hàng 2 đến hàng 4 */
}
```

Hoặc sử dụng span để kéo dài qua nhiều cột/hàng:

```
.item {
  grid-column: span 2; /* Chiếm 2 cột */
}
```

- Canh chỉnh nội dung trong Grid: dùng justify-items, align-items, place-items để căn chỉnh phần tử con:

```
.container {
  justify-items: center; /* Căn giữa theo chiều ngang */
}
```

- ```

 align-items: center; /* Căn giữa theo chiều dọc */
 }

```
- Canh chỉnh toàn bộ Grid
 

```

.container {
 /*Căn chỉnh toàn bộ lưới theo chiều ngang*/
 justify-content: center;
 /*Căn chỉnh toàn bộ lưới theo chiều dọc*/
 align-content: center;
}

```
  - Thuộc tính grid-auto-flow xác định cách các ô được xếp khi không đủ chỗ
 

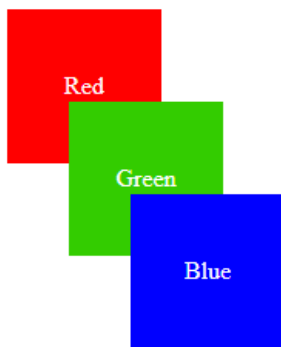
```

.container {
 grid-auto-flow: row; /* Mặc định, sắp xếp theo hàng */
}

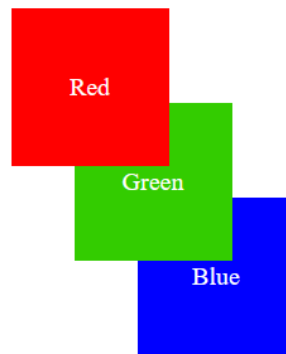
```

## 1.2. Giới thiệu bài tập mẫu

**Bài 1.** Tạo trang BaiTapMau1.html:



Hình (a)



Hình (b)

### Yêu cầu:

- Tạo 3 block có kích thước 100 x 100 và vị trí như hình (a)
- Thêm thuộc tính position và Z-index tương ứng các lớp để thay đổi thứ tự xếp chồng của các Block như hình (b)

### Hướng dẫn:

- Tạo Element
 

```

<body>
 <div class="red">Red</div>
 <div class="green">Green</div>
 <div class="blue">Blue</div>
</body>

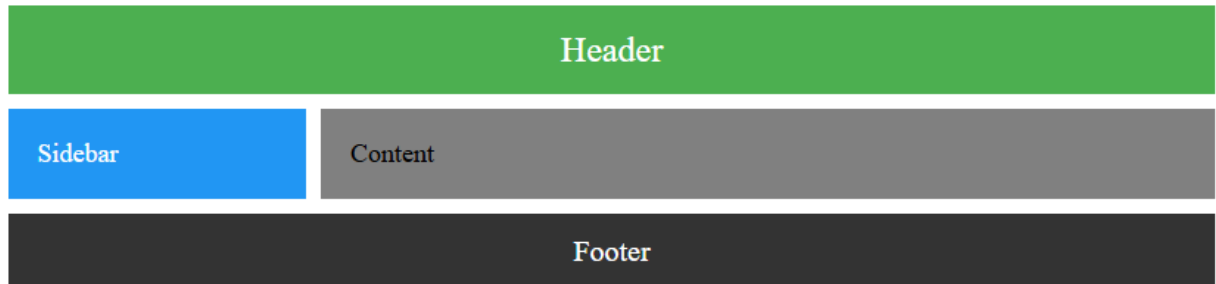
```
- Định dạng CSS cho Element và thêm thuộc tính Animation:

```
<style>
 .red{
 margin-left:50px;
 margin-top:50px;
 width:100px;
 height:100px;
 color:#FFF;
 background-color:#F00;
 line-height:100px;
 text-align:center;
 position:relative;
 z-index:1;
 }
 .green{
 margin-left:90px;
 margin-top:-40px;
 width:100px;
 height:100px;
 color:#FFF;
 background-color:#3C0;
 line-height:100px;
 text-align:center;
 position:relative;
 z-index:2;
 }
 .blue{
 margin-left:130px;
 margin-top:-40px;
 width:100px;
 height:100px;
 color:#FFF;
 background-color:#00F;
 line-height:100px;
 text-align:center;
 position:relative;
 z-index:3;
 }
}
```

</style>

- Thay đổi thứ tự z-index của 3 phần tử

**Bài 2.** Tạo BaiTapMau2.html thiết kế trang web theo layout như sau sử dụng grid:



**Hướng dẫn:**

**Bước 1.** Tạo các element như sau:

```
<div class="container">
 <header class="header">Header</header>
 <aside class="sidebar">Sidebar</aside>
 <main class="content">Content</main>
 <footer class="footer">Footer</footer>
</div>
```

**Bước 2.** Tạo các style

```
/* Thiết lập Grid */
.container {
 display: grid;
 grid-template-areas: /*đặt tên cho các khu vực trong Grid*/
 "header header"
 "sidebar content"
 "footer footer";
 grid-template-columns: 1fr 3fr; /* Sidebar chiếm 1 phần,
 Content chiếm 3 phần */
 grid-template-rows: 60px auto 50px; /* Header cao 60px,
 Footer cao 50px */
 gap: 10px; /* Khoảng cách giữa các ô */
 width: 90%;
 margin: 20px auto;
}
/* Định dạng các phần tử */
.header {
 grid-area: header;
 background-color: #4CAF50;
```



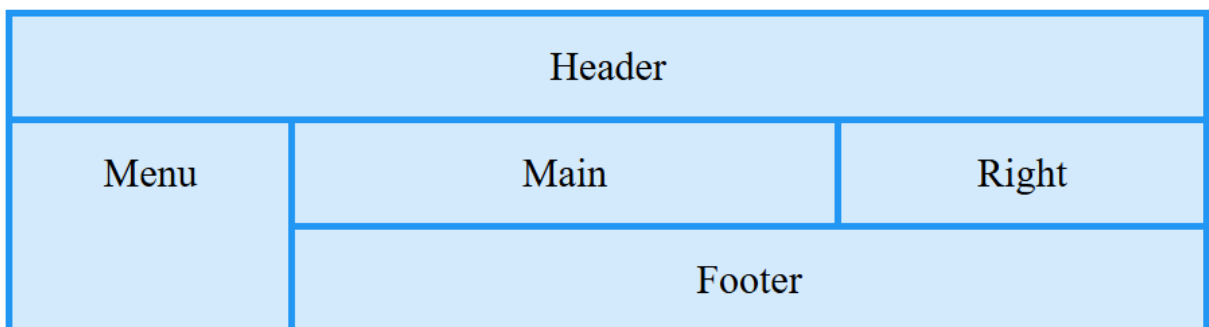
```

 color: white;
 text-align: center;
 line-height: 60px;
 font-size: 24px;
 }
 .sidebar {
 grid-area: sidebar;
 background-color: #2196F3;
 color: white;
 padding: 20px;
 font-size: 18px;
 }
 .content {
 grid-area: content;
 background-color: #808080;
 padding: 20px;
 font-size: 18px;
 }
 .footer {
 grid-area: footer;
 background-color: #333;
 color: white;
 text-align: center;
 line-height: 50px;
 font-size: 20px;
 }
}

```

**Bài 3.** Tạo BaiTapMau2.html thiết kế trang web theo layout như sau sử dụng grid

#### **KHUNG TRANG SỬ DỤNG CSS GRID**



**Yêu cầu:**

- Dùng class selector

- Chia khung trang thành 10 cột
  - Header: từ cột 1 đến cột 10
  - Menu: từ cột 1 và chiếm 2 cột
  - Main: từ cột 3 và chiếm 5 cột
  - Right: từ cột 8 và chiếm 3 cột
  - Footer: từ cột 3 và chiếm 8 cột
- Sử dụng display: grid
- Grid-column, grid-row

**Hướng dẫn:** Tạo style như sau:

```
.grid-container{
 display:grid;
 grid-template-columns: auto auto auto auto auto auto auto auto auto auto auto;
 grid-gap: 5px;
 background-color:#2196F3;
 padding: 5px;
}
.grid-container > div {
 background-color:rgba(255,255,255,0.8);
 text-align:center;
 padding:20px 0;
 font-size:30px;
}
.header{
 grid-column:1 / span 10;
 grid-row:1;
}
```

Tương tự cho các phần tử còn lại.

## 2. Bài tập tại lớp

**Bài 1.** Tạo Layout Gallery ảnh:



### Yêu cầu:

- Tạo một gallery hình ảnh với Grid.
- Ảnh đầu tiên rộng gấp đôi các ảnh khác.
- Sử dụng grid-column để điều chỉnh kích thước..

### Bài 2. Tạo trang sử dụng Grid CSS như sau:

## Welcome to My Homepage

Use the menu to select difference style sheets

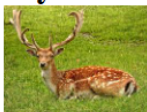
Trang chủ

Dịch vụ

Liên hệ

Về chúng tôi


### Same page Different StyleSheets



This is a Demonstration of how different stylesheet can change the layout of your html Page. You can change the layout of this page try selecting the different style sheet in the menu, or by selecting one of the following links  
[StyleSheet1](#), [StyleSheet2](#), [StyleSheet3](#), [StyleSheet4](#)

### No Style

This is a Demonstration of how different stylesheet can change the layout of your html Page. You can change the layout of this page try selecting the different style sheet in the menu, or by selecting one of the following links



#### Side-Bar

Từ ngày 18/5 đến 14/6, độc giả chia sẻ khoảnh khắc đầu tiên đáng nhớ của bé như ăn dặm, bi bô tập nói, chụp chứng những bước đi đầu tiên; cảm xúc hồi hộp, lòng ngóng khi cha mẹ lần đầu thay tã, tắm, massage, hát ru, mặc đồ... cho con, có cơ hội nhận giải thưởng tiền mặt và quà tặng hấp dẫn. Ba mẹ trúng giải sẽ nhận gói tiết kiệm bím tã nhập khẩu 2 triệu đồng hoặc 5 triệu đồng, áp dụng khi mua tã Genki - Whito - Whoopee tại Con Cưng - hệ thống siêu thị mẹ bầu và em bé toàn quốc trong một năm, không giới hạn số lần mua.

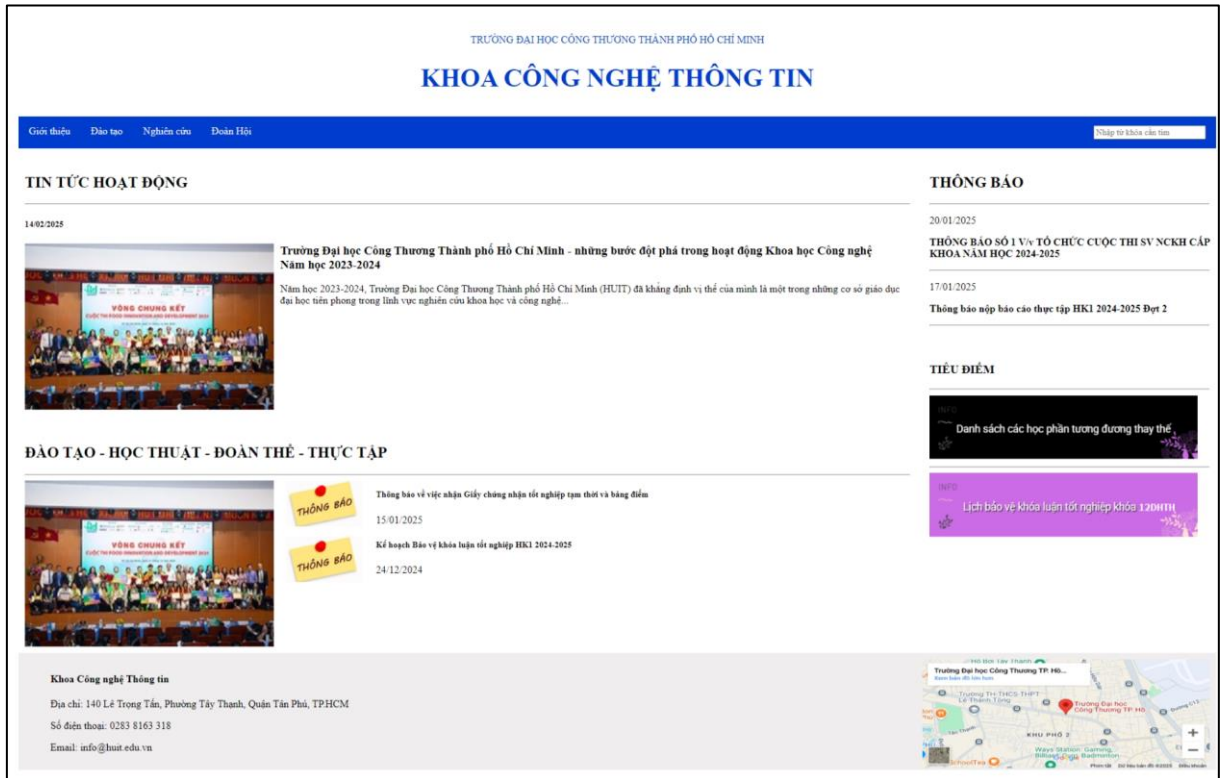
Chuyên mục văn hoá giải trí của VnExpress. © 2024 Ngoisao.net  
Tầng 5, Tòa nhà FPT Cầu Giấy, phố Duy Tân, Phường Dịch Vọng Hậu, Quận Cầu Giấy, Hà Nội  
Email: [lienhe@ngoisao.net](mailto:lienhe@ngoisao.net) | Điện thoại: 024 7300 9999 - Ext 4546

<https://localhost:44391/ViduMau1.html#>

### Yêu cầu:

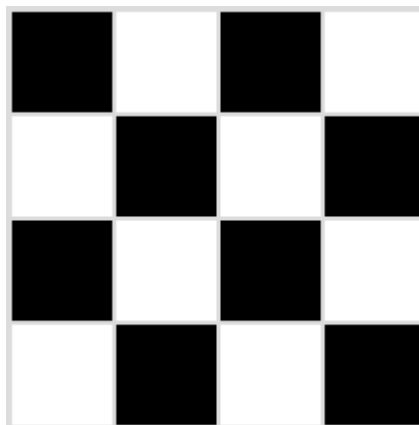
- Dùng class selector
- Sử dụng display Grid
- Grid-column, grid-row.

**Bài 3.** Thiết kế trang web với giao diện sau với các thuộc tính CSS:



### 3. Bài tập về nhà

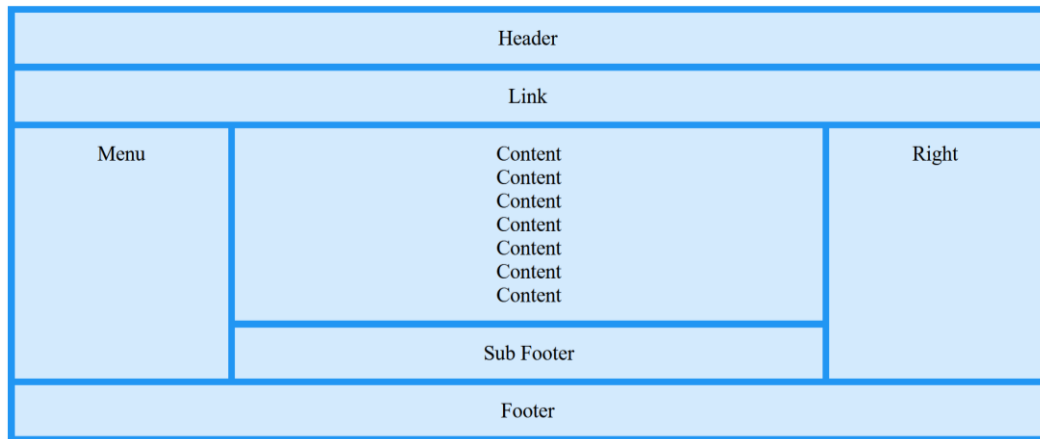
**Bài 1.** Tạo bàn cờ vua bằng Grid:



**Yêu cầu:**

- Tạo bàn cờ vua 4x4.
- Các ô xen kẽ trắng - đen.

**Bài 2.** Tạo trang theo layout như sau:



**Yêu cầu:**

- Dùng class selector
- Chia khung trang thành 10 cột
- Sử dụng display Grid
- Grid-column, grid-row

**Bài 3.** Sử dụng CSS và HTML để thiết kế các mẫu layout sau đây (Lưu ý: tạo file CSS riêng từ bên ngoài). Độ rộng layout web là 900px, các thành phần còn lại bên trong tự cho theo tỉ lệ như trong hình.

