



VIETNAM
AUSTRALIA
Vocational College

Slides 2.6 - Thiết kế Website theo bố cục Grid

Mentor: Nguyễn Bá Minh Đạo



Nội dung

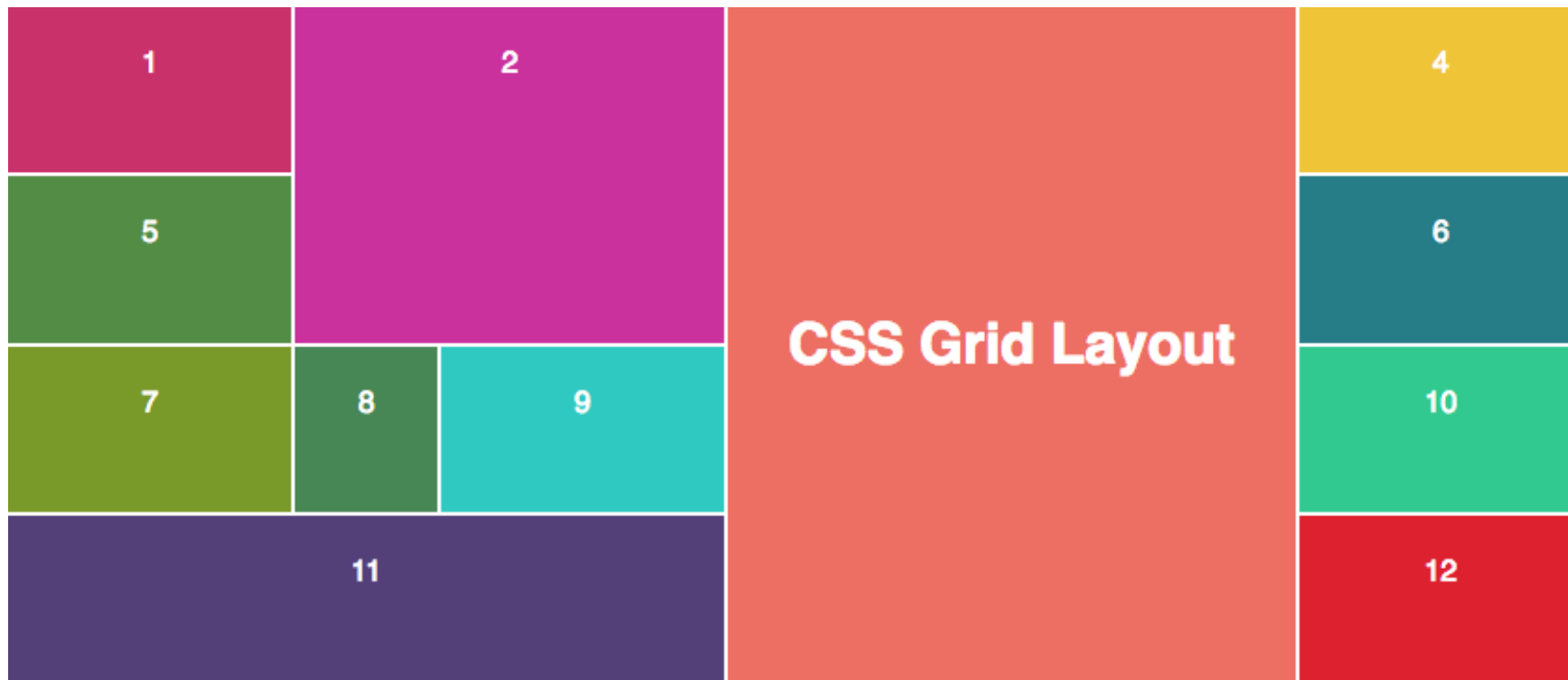
1. Tổng quan về Grid Layout
2. Thành phần của Grid Layout
3. Thuộc tính của Grid Container
4. Thuộc tính của các Grid Items
5. Đặt tên cho các Grid Items
6. Thứ tự của các Grid Items
7. Responsive Web với Grid Layout



Tổng quan về Grid Layout

❑ Grid là gì?

♦ **Bố cục Grid**(gọi ngắn gọn là **Grid**): Là **một hệ thống bố cục grid-based** (bố cục dạng dưới) **với các hàng và cột**, giúp thiết kế các trang web dễ dàng hơn mà không cần phải sử dụng float và các thuộc tính position.





Tổng quan về Grid Layout

❑ Để sử dụng Grid:

♦ Để bắt đầu sử dụng bố cục Grid, bạn cần **định nghĩa** một **phần tử thùng chứa dạng lưới** (**grid-container**) và thiết lập thuộc tính **display:grid** hoặc **display:inline-grid** cho phần tử **grid-container** này.

```
<body>
  <div class="grid-container">
    <div class="grid-item">1</div>
    <div class="grid-item">2</div>
    <div class="grid-item">3</div>
    <div class="grid-item">4</div>
    <div class="grid-item">5</div>
    <div class="grid-item">6</div>
    <div class="grid-item">7</div>
    <div class="grid-item">8</div>
    <div class="grid-item">9</div>
  </div>
</body>
```

```
<style>
  .grid-container {
    display: grid;
    grid-template-columns: auto auto auto;
    background-color: #2196F3;
    padding: 10px;
  }

  .grid-item {
    background-color: rgba(255, 255, 255, 0.8);
    border: 1px solid rgba(0, 0, 0, 0.8);
    padding: 20px;
    font-size: 30px;
    text-align: center;
  }
</style>
```

Hoặc display:
inline-grid;



Tổng quan về Grid Layout

❑ Để sử dụng Grid:

♦ Để bắt đầu sử dụng bố cục Grid, bạn cần **định nghĩa** một **phần tử thùng chứa dạng lưới** (**grid-container**) và thiết lập thuộc tính **display:grid** hoặc **display:inline-grid** cho phần tử **grid-container** này.

Grid Container - Display Grid

127.0.0.1:5500/grid-container-01.html

1	2	3
4	5	6
7	8	9

Grid Container - Display Inline G

127.0.0.1:5500/grid-container-02.html

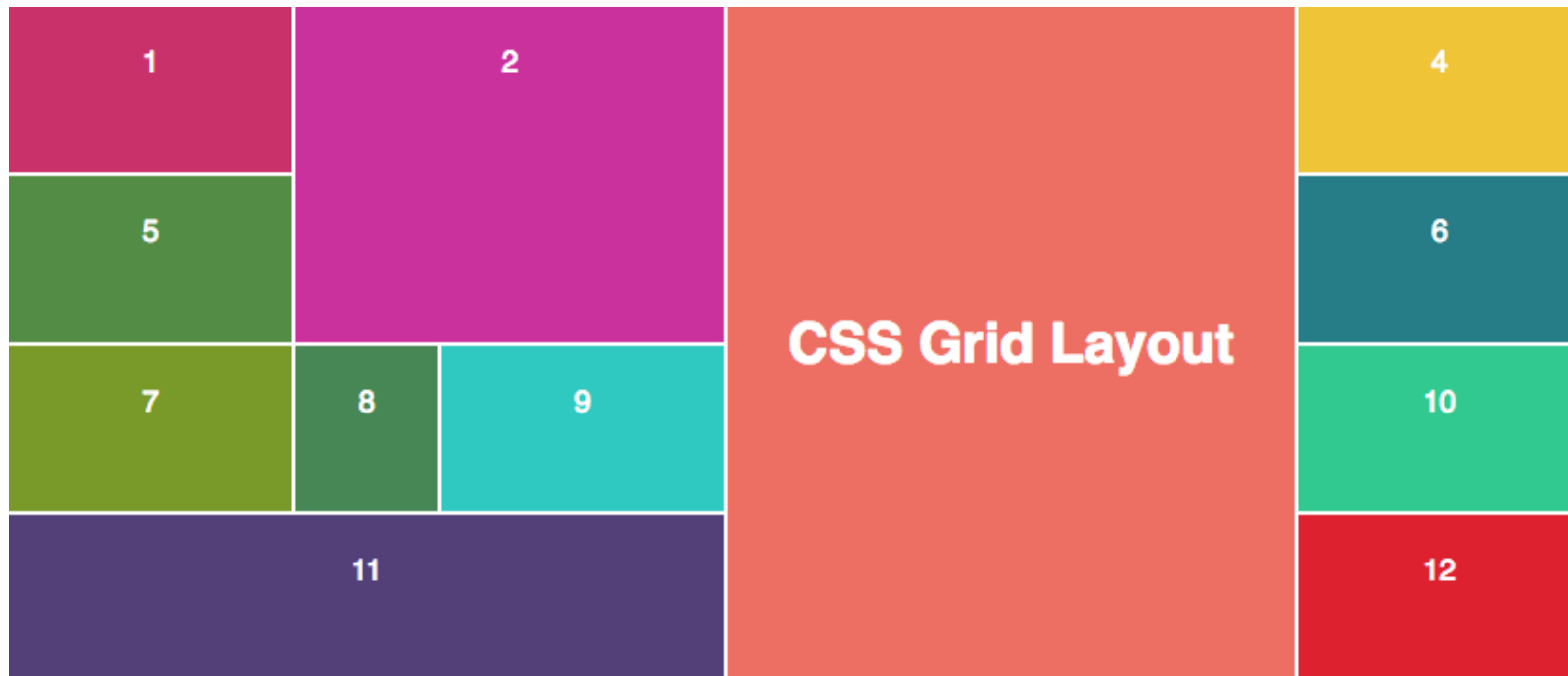
1	2	3
4	5	6
7	8	9



Thành phần của Grid Layout

❑ Các thành phần của Grid Layout:

- ♦ Một bố cục grid bao gồm **1 phần tử cha** và **1 hoặc nhiều phần tử con**.
 - Phần tử cha (hay còn gọi 1 **grid container**) khi phần tử này được thiết lập thuộc tính **display** là **grid** hoặc **inline-grid**.
 - Các phần tử con (hay còn gọi là các **grid item**), đặt bên trong các cột và hàng.

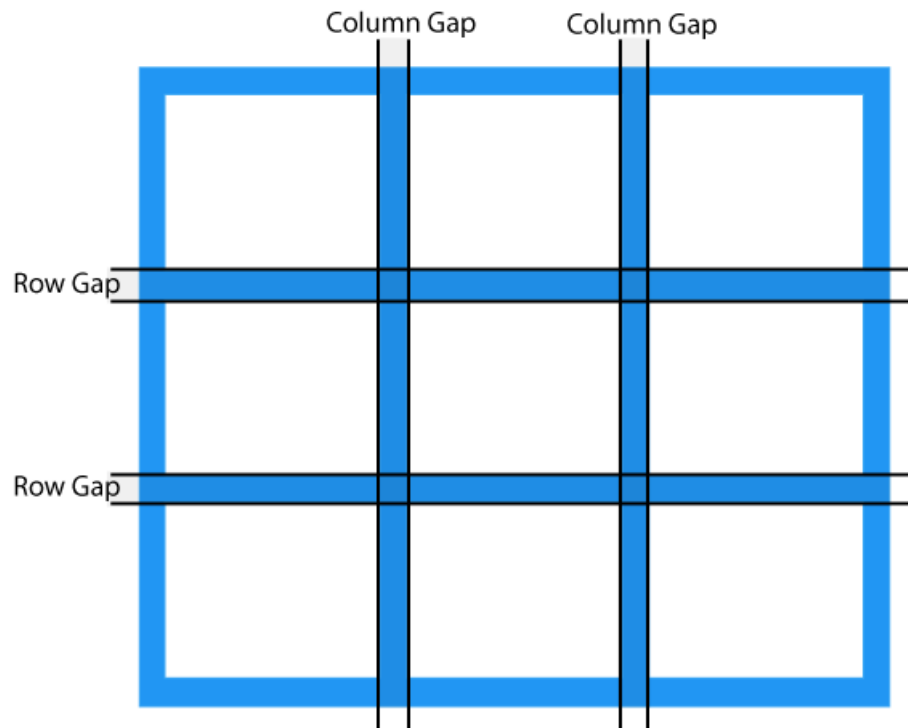




Thành phần của Grid Layout

❑ Khoảng cách lưới (Grid Gaps):

- ♦ **Khoảng cách giữa mỗi cột/hàng** được gọi là **khoảng cách lưới** (grid gaps).
- ♦ Chúng ta **có thể canh chỉnh khoảng cách này** bằng cách dùng **các thuộc tính** sau:
 - **grid-column-gap**
 - **grid-row-gap**
 - **grid-gap**



```
<body>
  <div class="grid-container">
    <div class="grid-item">1</div>
    <div class="grid-item">2</div>
    <div class="grid-item">3</div>
    <div class="grid-item">4</div>
    <div class="grid-item">5</div>
    <div class="grid-item">6</div>
    <div class="grid-item">7</div>
    <div class="grid-item">8</div>
    <div class="grid-item">9</div>
  </div>
</body>
```



Thành phần của Grid Layout

❑ Khoảng cách lưới (Grid Gaps):

- ♦ **Khoảng cách giữa mỗi cột/hàng** được gọi là **khoảng cách lưới** (grid gaps).
- ♦ Chúng ta có thể canh chỉnh khoảng cách này bằng cách dùng các thuộc tính sau:
 - **grid-column-gap**
 - **grid-row-gap**
 - **grid-gap**

```
<style>
.grid-container {
  display: grid;
  grid-column-gap: 50px;
  grid-template-columns: auto auto auto;
  background-color: #2196F3;
  padding: 10px;
}

.grid-item {
  background-color: rgba(255, 255, 255, 0.8);
  border: 1px solid rgba(0, 0, 0, 0.8);
  padding: 20px;
  font-size: 30px;
  text-align: center;
}
</style>
```

Hoặc grid-row-gap: 50px;

Hoặc grid-gap: 50px;



Thành phần của Grid Layout

❑ Khoảng cách lưới (Grid Gaps):

- ♦ **Khoảng cách giữa mỗi cột/hàng** được gọi là **khoảng cách lưới** (grid gaps).
- ♦ Chúng ta **có thể canh chỉnh khoảng cách này** bằng cách dùng các **thuộc tính** sau:
 - grid-column-gap
 - grid-row-gap
 - **grid-gap**

```
<body>
  <div class="grid-container">
    <div class="grid-item">1</div>
    <div class="grid-item">2</div>
    <div class="grid-item">3</div>
    <div class="grid-item">4</div>
    <div class="grid-item">5</div>
    <div class="grid-item">6</div>
    <div class="grid-item">7</div>
    <div class="grid-item">8</div>
    <div class="grid-item">9</div>
  </div>
</body>
```

```
<style>
.grid-container {
  display: grid;
  grid-gap: 50px 100px;
  grid-template-columns: auto auto auto;
  background-color: #2196F3;
  padding: 10px;
}

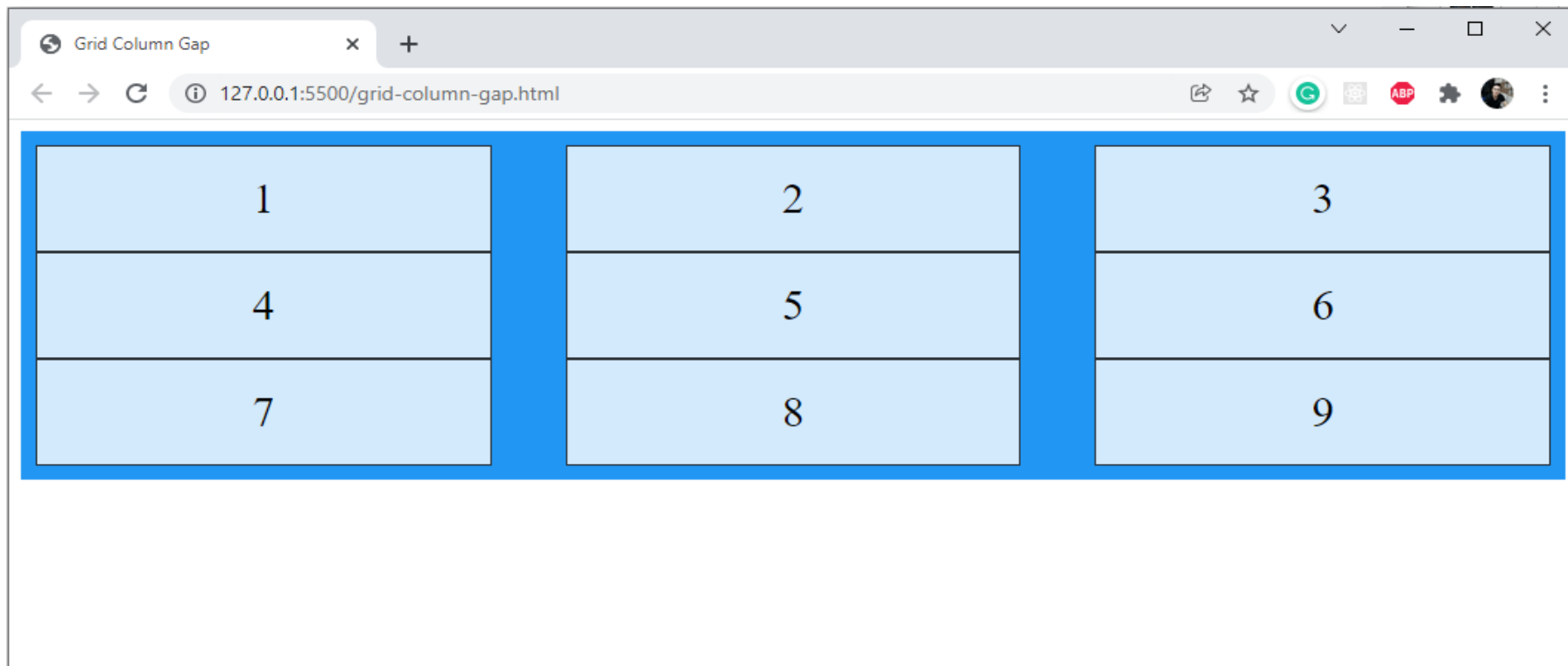
.grid-item {
  background-color: rgba(255, 255, 255, 0.8);
  border: 1px solid rgba(0, 0, 0, 0.8);
  padding: 20px;
  font-size: 30px;
  text-align: center;
}
</style>
```



Thành phần của Grid Layout

❑ Khoảng cách lưới (Grid Gaps):

- ♦ **Khoảng cách giữa mỗi cột/hàng** được gọi là **khoảng cách lưới** (grid gaps).
- ♦ Chúng ta **có thể canh chỉnh khoảng cách này** bằng cách dùng **các thuộc tính** sau:
 - **grid-column-gap**
 - grid-row-gap
 - grid-gap

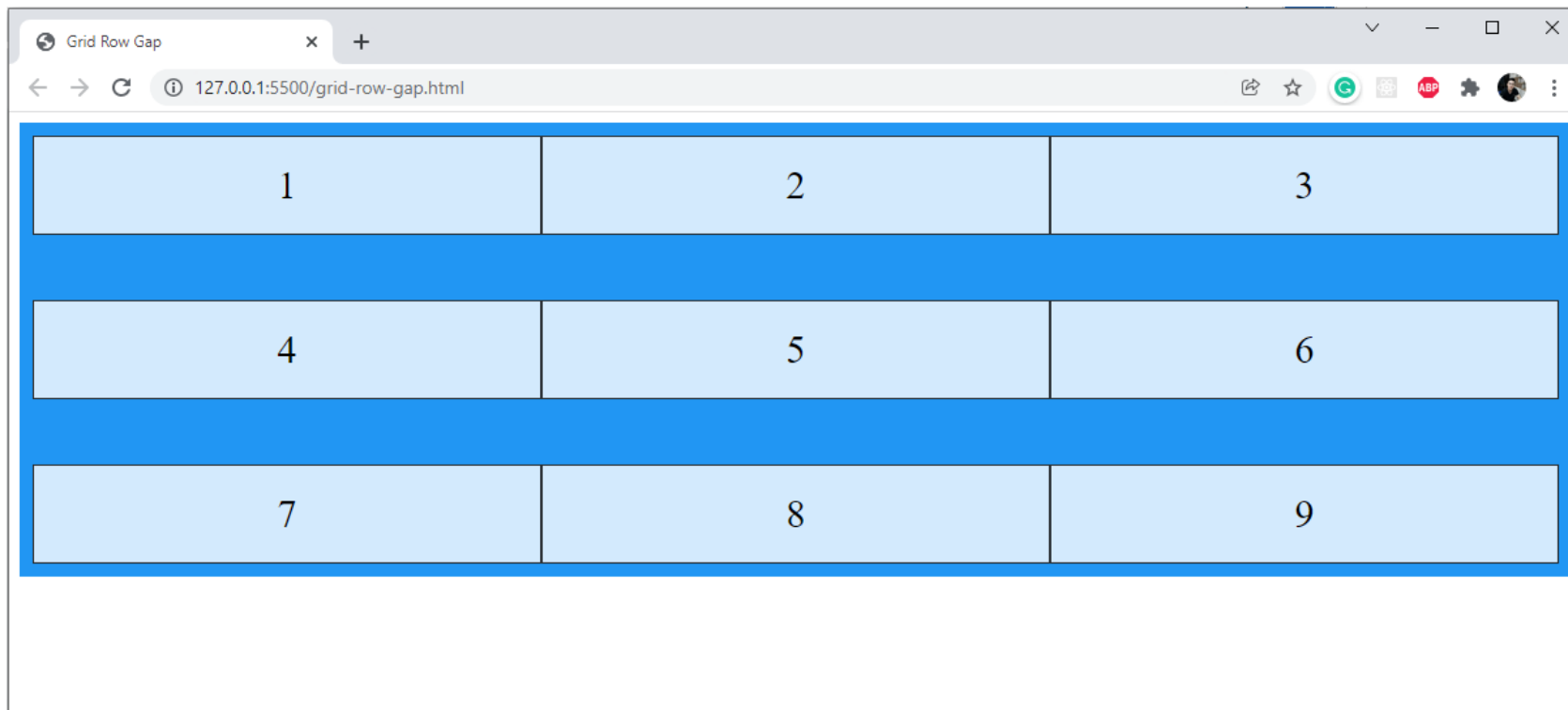




Thành phần của Grid Layout

❑ Khoảng cách lưới (Grid Gaps):

- ♦ **Khoảng cách giữa mỗi cột/hàng** được gọi là **khoảng cách lưới** (grid gaps).
- ♦ Chúng ta **có thể canh chỉnh khoảng cách này** bằng cách dùng **các thuộc tính** sau:
 - grid-column-gap
 - **grid-row-gap**
 - grid-gap

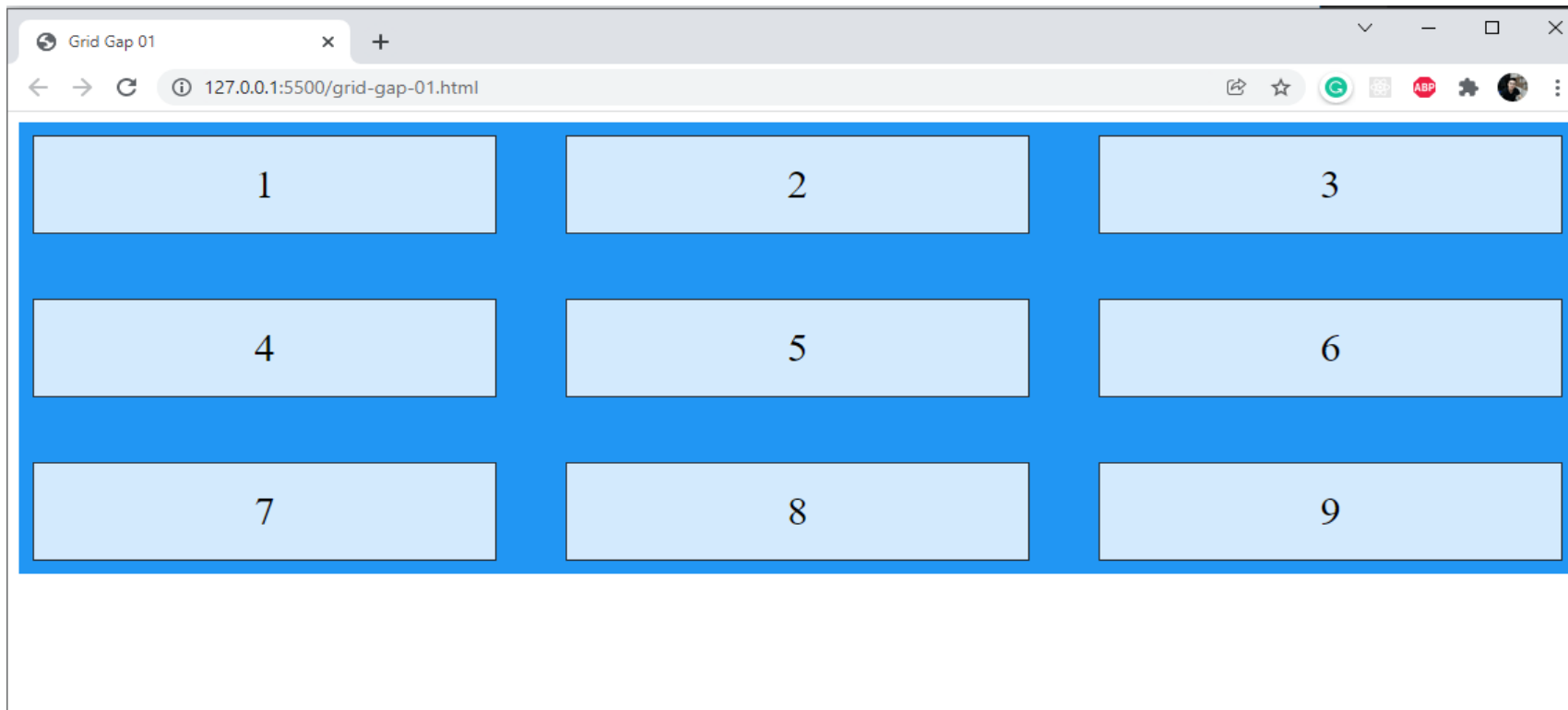




Thành phần của Grid Layout

❑ Khoảng cách lưới (Grid Gaps):

- ♦ **Khoảng cách giữa mỗi cột/hàng** được gọi là **khoảng cách lưới** (grid gaps).
- ♦ Chúng ta **có thể canh chỉnh khoảng cách này** bằng cách dùng **các thuộc tính** sau:
 - grid-column-gap
 - grid-row-gap
 - **grid-gap**

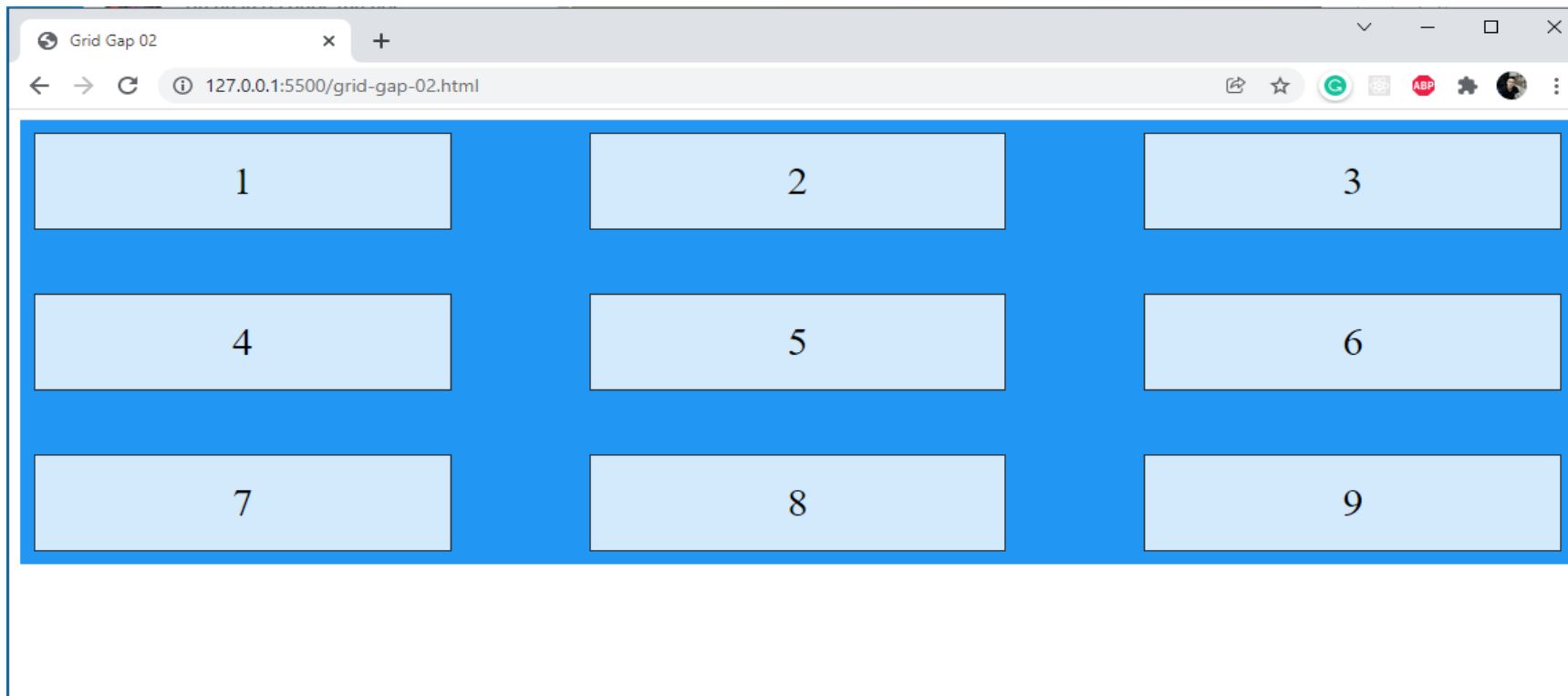




Thành phần của Grid Layout

❑ Khoảng cách lưới (Grid Gaps):

- ♦ **Khoảng cách giữa mỗi cột/hàng** được gọi là **khoảng cách lưới** (grid gaps).
- ♦ Chúng ta **có thể canh chỉnh khoảng cách này** bằng cách dùng **các thuộc tính** sau:
 - grid-column-gap
 - grid-row-gap
 - **grid-gap**

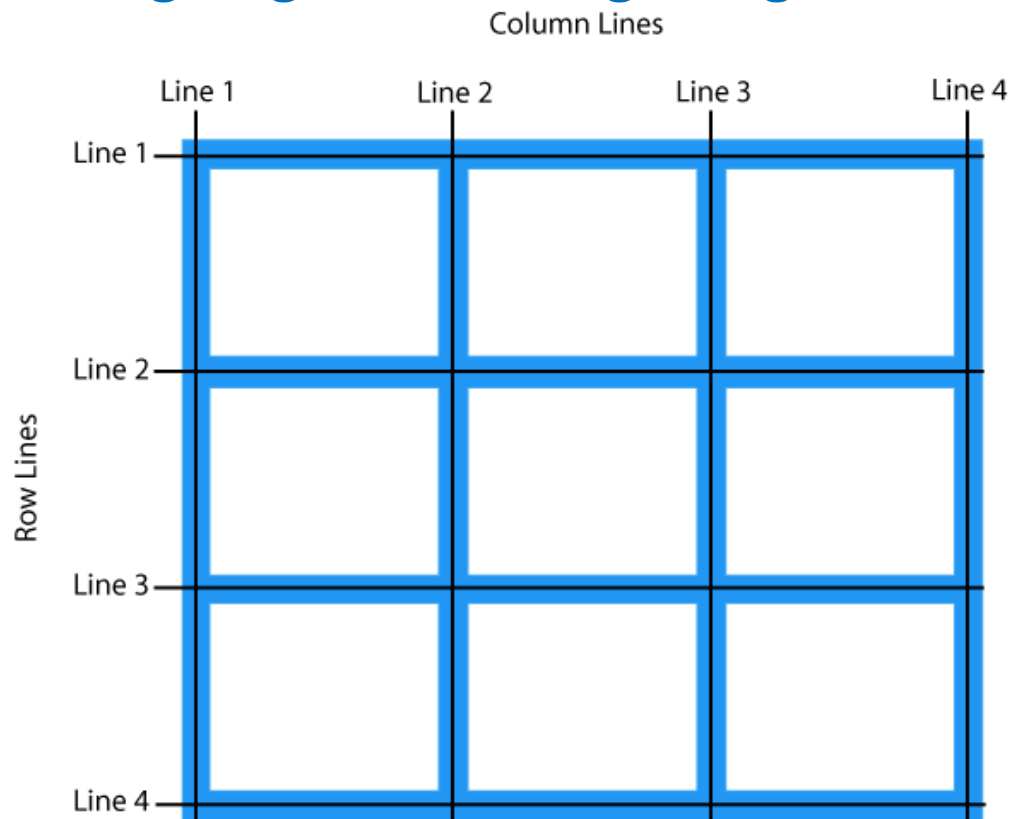




Thành phần của Grid Layout

❑ Đường kẻ lưới (Grid Lines):

- ◆ **Đường kẻ giữa các cột trong lưới** được gọi là **đường kẻ cột** (column lines).
- ◆ **Đường kẻ giữa các dòng trong lưới** được gọi là **đường kẻ dòng** (row lines).



```
<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
  <div class="item7">7</div>
  <div class="item8">8</div>
</div>
```



Thành phần của Grid Layout

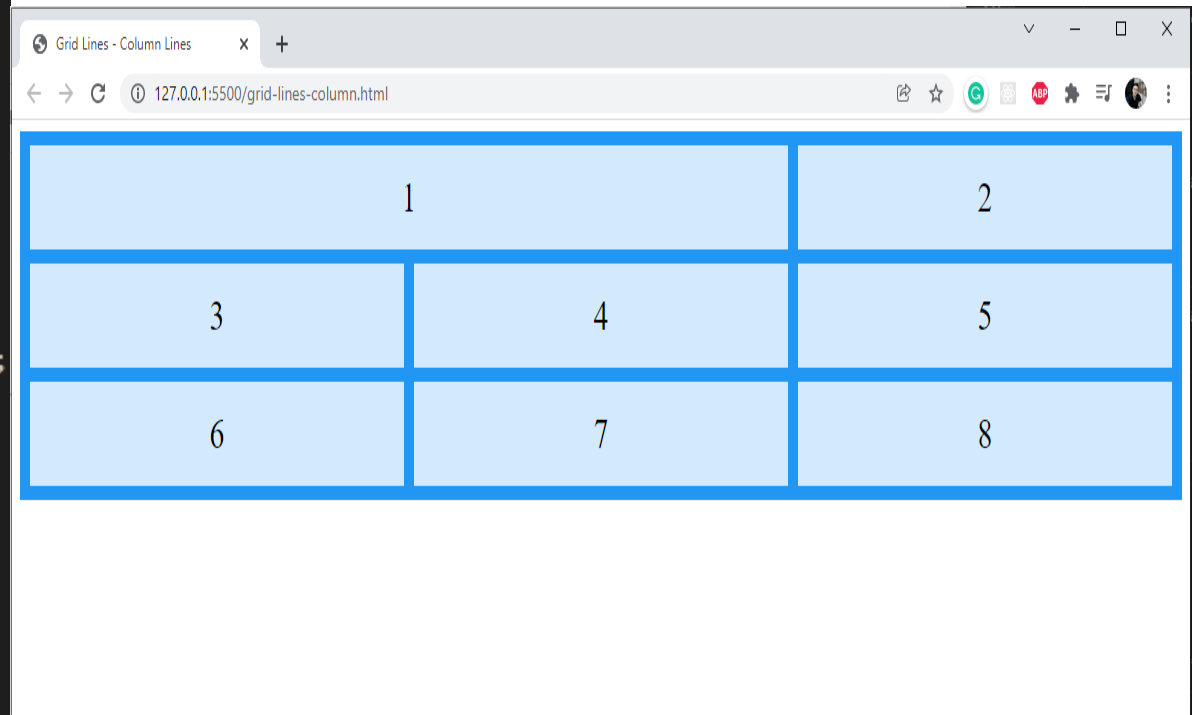
❑ Đường kẻ lưới (Grid Lines):

- ◆ Đặt một **mục lưới** (grid item) **trong** một **vùng chứa lưới** (grid container) **theo cột**:
 - Đặt một mục lưới ở đường kẻ cột 1 và để nó kết thúc trên đường kẻ cột 3:

```
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}

.item1 {
  grid-column-start: 1;
  grid-column-end: 3;
}
</style>
```





Thành phần của Grid Layout

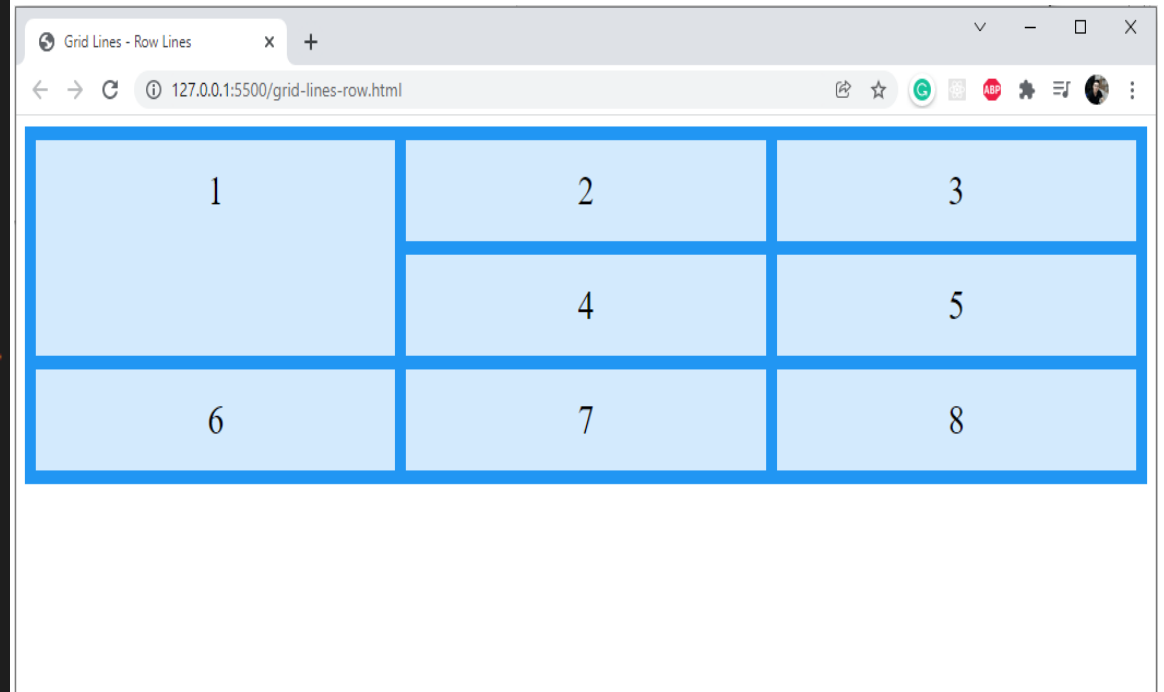
❑ Đường kẻ lưới (Grid Lines):

- ◆ Đặt một **mục lưới** (grid item) **trong** một **vùng chứa lưới** (grid container) **theo cột**:
 - Đặt một mục lưới ở đường kẻ dòng 1 và để nó kết thúc trên đường kẻ dòng 3:

```
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}

.item1 {
  grid-row-start: 1;
  grid-row-end: 3;
}
</style>
```



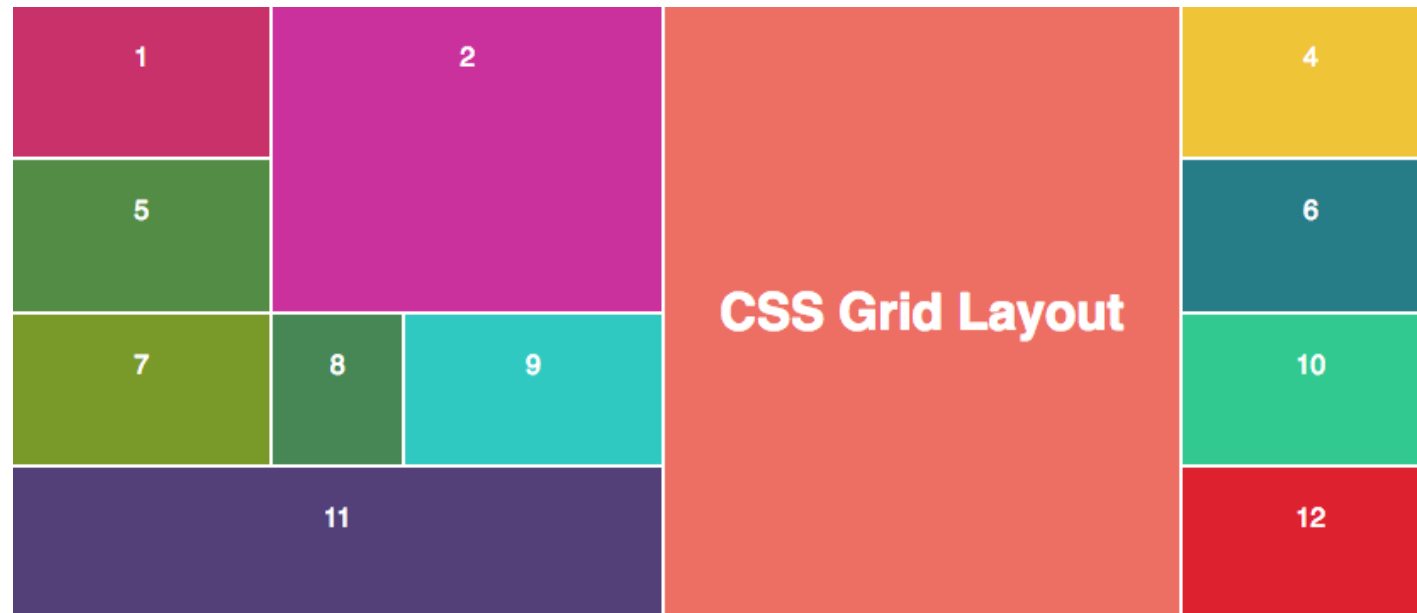


Thuộc tính của Grid Container

❑ Các thuộc tính của Grid Container:

◆ Thuộc tính **grid-template-columns**:

- **Xác định số lượng cột trong bố cục lưới** của bạn và nó **có thể xác định chiều rộng của mỗi cột**.
- **Giá trị là một danh sách được phân tách bằng dấu cách**, trong đó mỗi giá trị xác định chiều rộng của cột tương ứng.





Thuộc tính của Grid Container

❑ Các thuộc tính của Grid Container:

◆ Thuộc tính **grid-template-columns**:

- Ví dụ tạo lưới có 4 cột:

```
<div class="grid-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
  <div>7</div>
  <div>8</div>
</div>
```



```
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto auto;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}
</style>
```



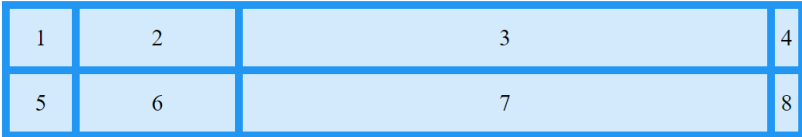
Thuộc tính của Grid Container

❑ Các thuộc tính của Grid Container:

◆ Thuộc tính **grid-template-columns**:

- Ví dụ tạo lưới có 4 cột với độ rộng các cột tự thiết lập:

```
<div class="grid-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
  <div>7</div>
  <div>8</div>
</div>
```



```
<style>
  .grid-container {
    display: grid;
    grid-template-columns: 80px 200px auto 30px;
    grid-gap: 10px;
    background-color: #2196F3;
    padding: 10px;
  }

  .grid-container > div {
    background-color: rgba(255, 255, 255, 0.8);
    text-align: center;
    padding: 20px 0;
    font-size: 30px;
  }
</style>
```

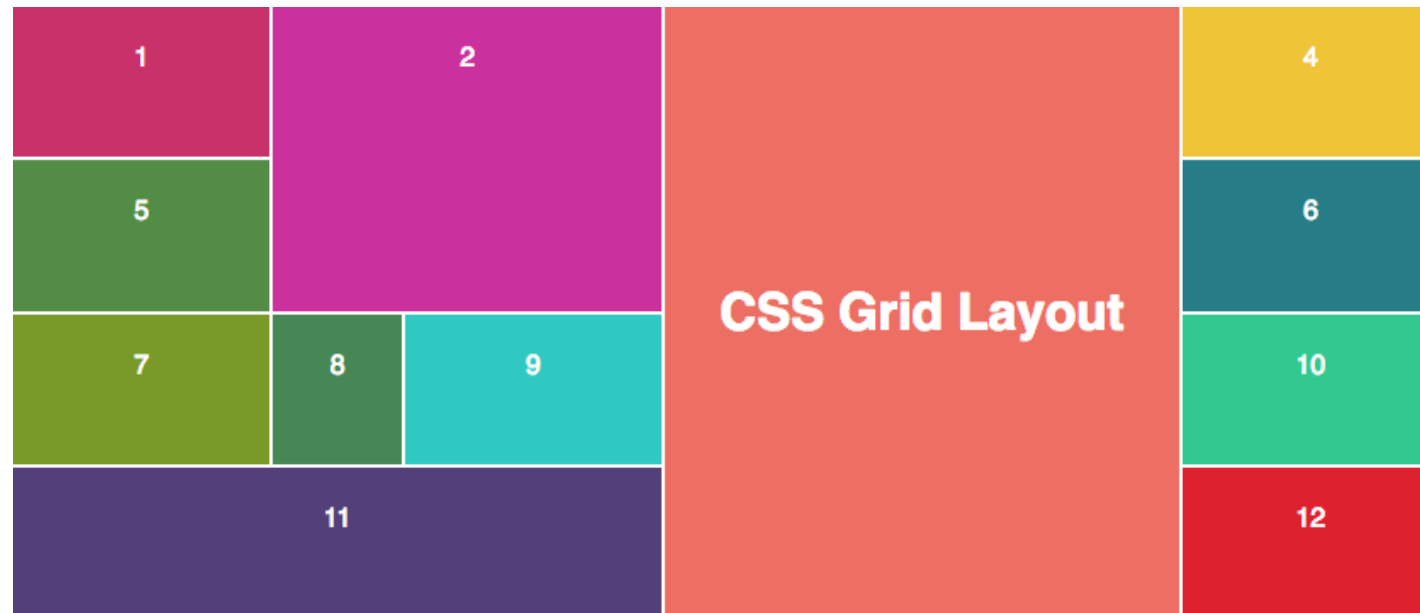


Thuộc tính của Grid Container

❑ Các thuộc tính của Grid Container:

◆ Thuộc tính **grid-template-rows**:

- **Xác định số lượng cột trong bố cục lưới** của bạn và nó **có thể xác định chiều cao của mỗi dòng**.
- **Giá trị là một danh sách được phân tách bằng dấu cách**, trong đó mỗi giá trị xác định chiều rộng của cột tương ứng.





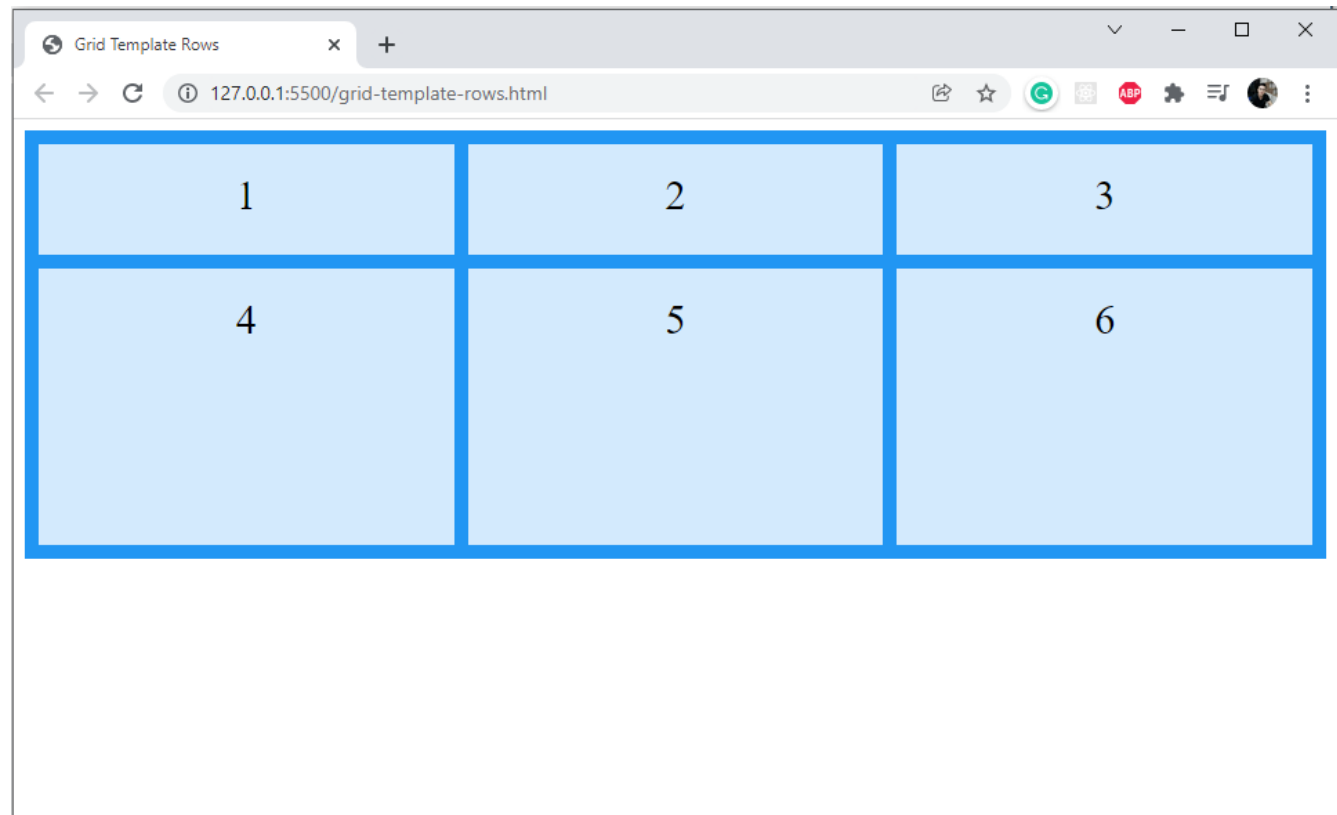
Thuộc tính của Grid Container

❑ Các thuộc tính của Grid Container:

◆ Thuộc tính **grid-template-rows**:

- Ví dụ tạo lưới có 2 dòng với độ cao mỗi dòng tự thiết lập:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Grid Template Rows</title>
  <style>
    .grid-container {
      display: grid;
      grid-template-columns: auto auto auto;
      grid-template-rows: 80px 200px;
      grid-gap: 10px;
      background-color: #2196F3;
      padding: 10px;
    }
    .grid-container > div {
      background-color: rgba(255, 255, 255, 0.8);
      text-align: center;
      padding: 20px 0;
      font-size: 30px;
    }
  </style>
</head>
<body>
  <div class="grid-container">
    <div>1</div>
    <div>2</div>
    <div>3</div>
    <div>4</div>
    <div>5</div>
    <div>6</div>
  </div>
</body>
</html>
```





Thuộc tính của Grid Container

❑ Các thuộc tính của Grid Container:

◆ Thuộc tính **justify-content**:

- **justify-content**: Canh chỉnh **toàn bộ các grid** bên **trong container**.
- Có các thuộc tính phổ biến như sau:
 - **space-evenly**: Giá trị "khoảng trắng đồng đều" sẽ cung cấp cho các cột khoảng cách bằng nhau giữa và xung quanh chúng.

```
<div class="grid-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
</div>
```

```
<style>
.grid-container {
  display: grid;
  justify-content: space-evenly; /*khoảng trắng đồng đều*/
  grid-template-columns: 50px 50px 50px;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}
</style>
```

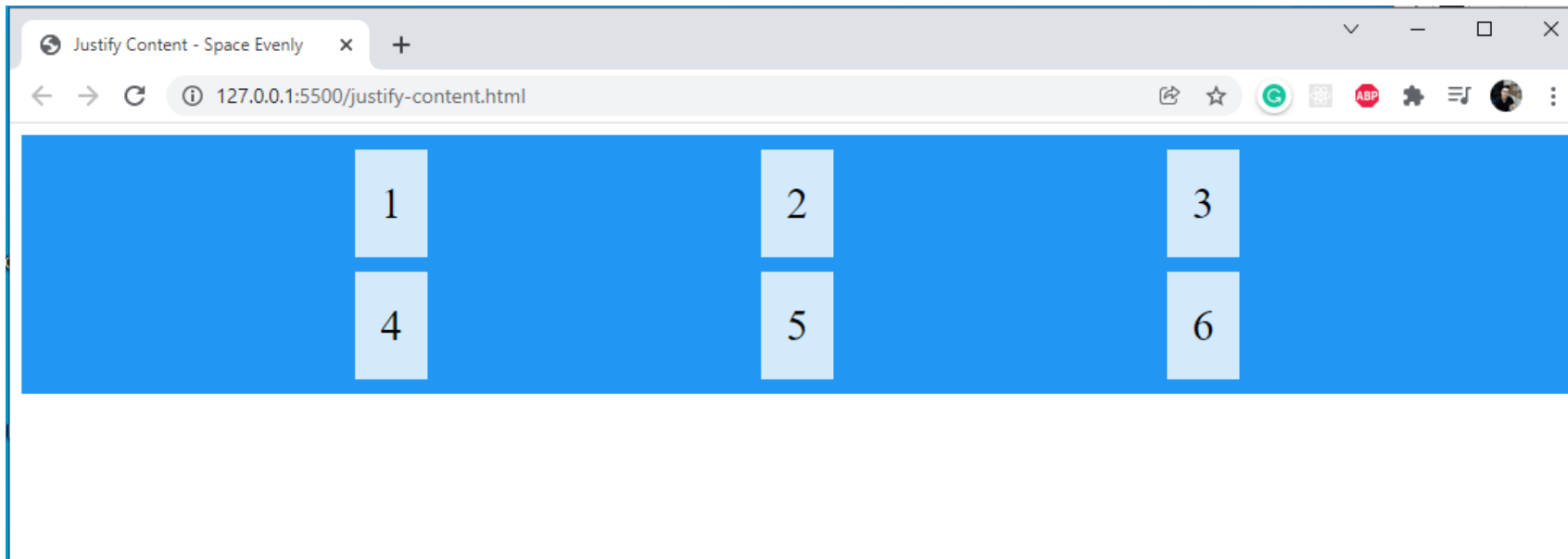


Thuộc tính của Grid Container

❑ Các thuộc tính của Grid Container:

◆ Thuộc tính **justify-content**:

- **justify-content**: Canh chỉnh **toàn bộ các grid** bên **trong container**.
- Có các thuộc tính phổ biến như sau:
 - **space-evenly**: Giá trị "khoảng trắng đồng đều" sẽ cung cấp cho các cột khoảng cách bằng nhau giữa và xung quanh chúng.





Thuộc tính của Grid Container

❑ Các thuộc tính của Grid Container:

◆ Thuộc tính **justify-content**:

- Có các thuộc tính phổ biến như sau:

- **space-around**: Giá trị "khoảng trống xung quanh" sẽ cung cấp cho các cột khoảng không gian bằng nhau xung quanh chúng.
- **space-between**: Giá trị "khoảng trắng giữa" sẽ cung cấp cho các cột khoảng cách bằng nhau giữa chúng.
- **center**: Giá trị "center" sẽ căn chỉnh lưới ở giữa vùng chứa.
- **start**: Giá trị "bắt đầu" sẽ căn chỉnh lưới ở đầu vùng chứa.
- **end**: Giá trị "end" sẽ căn chỉnh lưới ở cuối vùng chứa.



Thuộc tính của Grid Container

❑ Các thuộc tính của Grid Container:

◆ Thuộc tính **align-content**:

- **align-content**: Canh chỉnh **theo chiều dọc toàn bộ các grid** trong container.
- Có các thuộc tính phổ biến như sau:
 - **space-evenly**: Giá trị "khoảng trắng đồng đều" sẽ cung cấp cho các hàng khoảng cách bằng nhau giữa và xung quanh chúng.

```
<div class="grid-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
</div>
```

```
<style>
.grid-container {
  display: grid;
  height: 400px;
  align-content: space-evenly;
  grid-template-columns: auto auto auto;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}
</style>
```

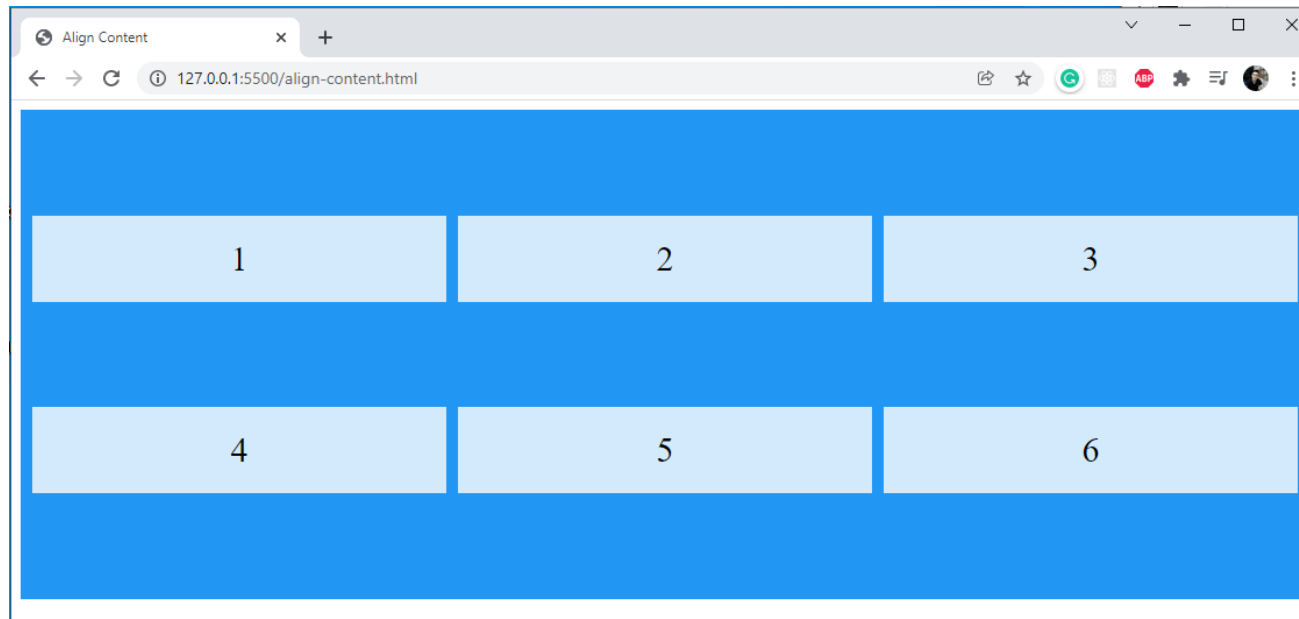


Thuộc tính của Grid Container

❑ Các thuộc tính của Grid Container:

◆ Thuộc tính **align-content**:

- **align-content**: Canh chỉnh **theo chiều dọc toàn bộ các grid** trong container.
- Có các thuộc tính phổ biến như sau:
 - **space-evenly**: Giá trị "khoảng trắng đồng đều" sẽ cung cấp cho các hàng khoảng cách bằng nhau giữa và xung quanh chúng.





Thuộc tính của Grid Container

❑ Các thuộc tính của Grid Container:

◆ Thuộc tính **align-content**:

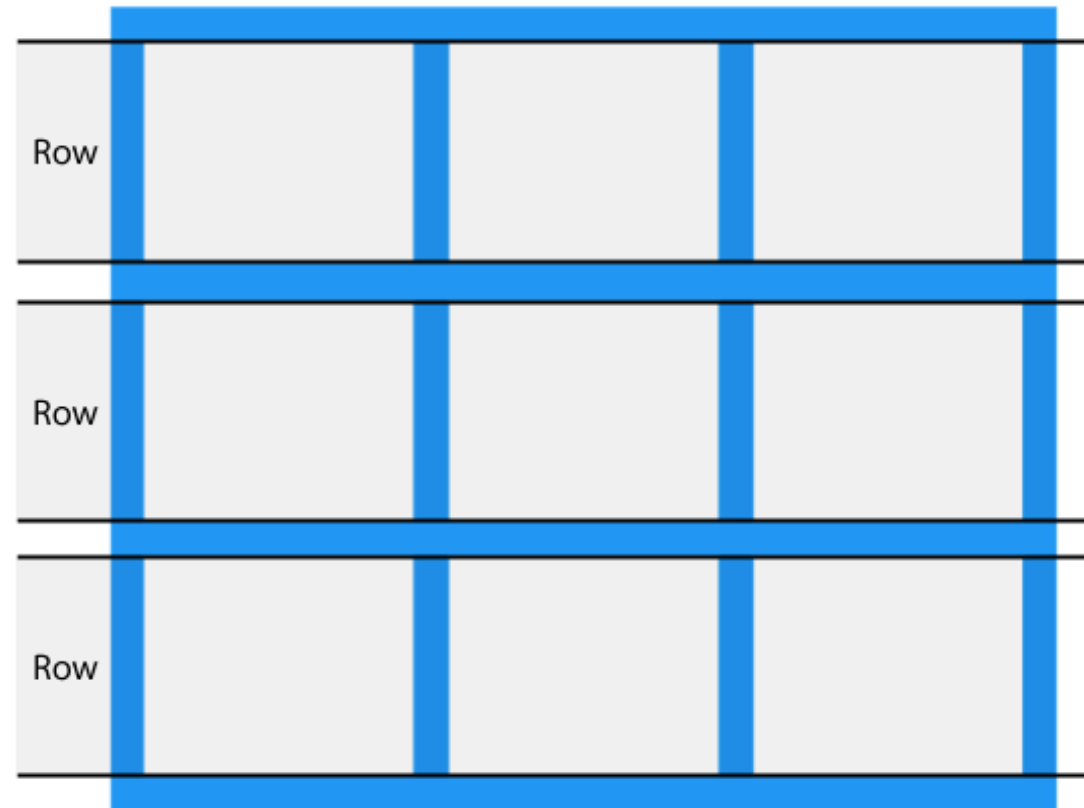
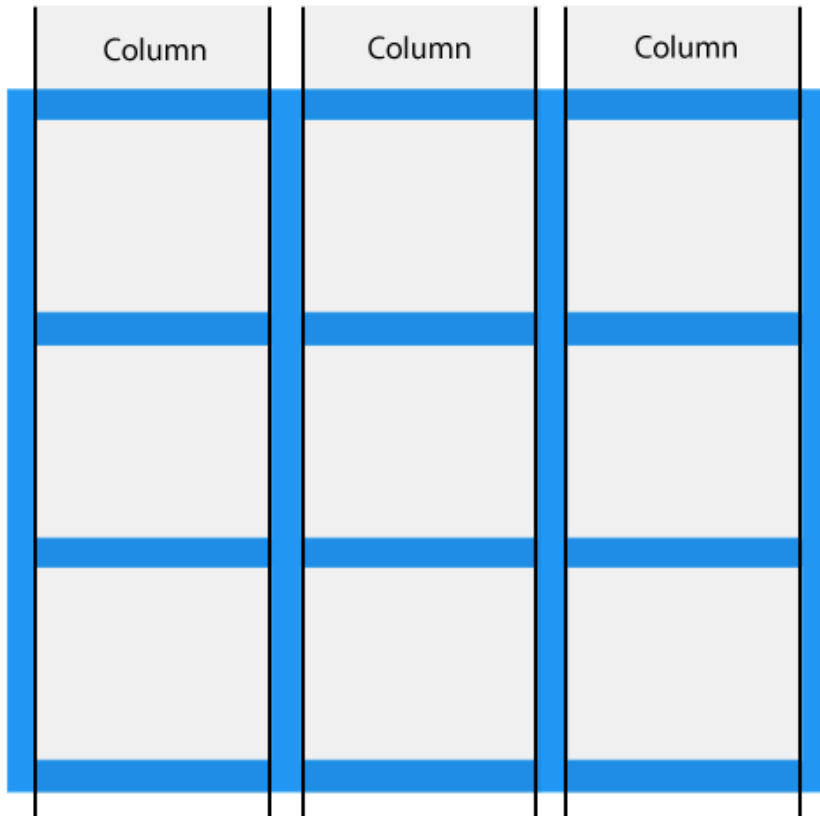
- **align-content**: Canh chỉnh **theo chiều dọc toàn bộ các grid** trong container.
- Có các thuộc tính phổ biến như sau:
 - **space-around**: Giá trị "khoảng trống xung quanh" sẽ cung cấp cho các hàng khoảng không gian bằng nhau xung quanh chúng.
 - **space-between**: Giá trị "khoảng trắng giữa" sẽ cung cấp cho các hàng khoảng cách bằng nhau giữa chúng.
 - **center**: Giá trị "center" sẽ căn chỉnh các hàng ở giữa vùng chứa.
 - **start**: Giá trị "start" sẽ căn chỉnh các hàng ở đầu vùng chứa.
 - **end**: Giá trị "end" sẽ căn chỉnh các hàng ở cuối vùng chứa.



Thuộc tính của các Grid Items

❑ Các dạng bố cục của Grid Items:

- ◆ **Bố cục Grid Columns:** các đường thẳng đứng của các grid items dạng columns (cột).
- ◆ **Bố cục Grid Rows:** các đường ngang của các grid items dạng rows (hàng).





Thuộc tính của các Grid Items

❑ Các thuộc tính của Grid Items:

◆ Thuộc tính **grid-column**:

- Là một thuộc tính viết tắt cho thuộc tính **grid-column-start** và thuộc tính **grid-column-end**.
- Để đặt 1 item trong grid-container, bạn có thể sử dụng thông số line numbers hoặc dùng từ khóa “span” để thiết lập vị trí item sẽ kéo dài bao nhiêu cột.

1				2	3
4	5	6	7	8	9
10	11	12	13	14	15



Thuộc tính của các Grid Items

❑ Các thuộc tính của Grid Items:

◆ Thuộc tính **grid-column**:

- Ví dụ tạo “item1” bắt đầu trên cột 1 và kết thúc trước cột 5:

```
<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
  <div class="item7">7</div>
  <div class="item8">8</div>
  <div class="item9">9</div>
  <div class="item10">10</div>
  <div class="item11">11</div>
  <div class="item12">12</div>
  <div class="item13">13</div>
  <div class="item14">14</div>
  <div class="item15">15</div>
</div>
```

```
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto auto auto auto;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}

.item1 {
  grid-column: 1 / 5;
}
</style>
```



Thuộc tính của các Grid Items

❑ Các thuộc tính của Grid Items:

◆ Thuộc tính **grid-column**:

- Ví dụ tạo “item1” bắt đầu trên cột 1 và kết thúc trước cột 5:

1				2	3
4	5	6	7	8	9
10	11	12	13	14	15



Thuộc tính của các Grid Items

❑ Các thuộc tính của Grid Items:

◆ Thuộc tính **grid-column**:

- Ví dụ tạo “item1” bắt đầu trên cột 1 và span thêm 3 cột tiếp theo:

```
<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
  <div class="item7">7</div>
  <div class="item8">8</div>
  <div class="item9">9</div>
  <div class="item10">10</div>
  <div class="item11">11</div>
  <div class="item12">12</div>
  <div class="item13">13</div>
  <div class="item14">14</div>
  <div class="item15">15</div>
  <div class="item16">16</div>
</div>
```

```
<style>
  .grid-container {
    display: grid;
    grid-template-columns: auto auto auto auto auto auto;
    grid-gap: 10px;
    background-color: #2196F3;
    padding: 10px;
  }

  .grid-container > div {
    background-color: rgba(255, 255, 255, 0.8);
    text-align: center;
    padding: 20px 0;
    font-size: 30px;
  }

  .item1 {
    grid-column: 1 / span 3;
  }
</style>
```




Thuộc tính của các Grid Items

❑ Các thuộc tính của Grid Items:

◆ Thuộc tính **grid-column**:

- Ví dụ tạo “item1” bắt đầu trên cột 1 và span thêm 3 cột tiếp theo:

The screenshot shows a web browser window titled "Grid Column 02" with the address bar displaying "127.0.0.1:5500/grid-column-02.html". The browser content displays a grid of 16 light blue items, each with a black border and a black number. The grid is organized into three rows and six columns. The first row contains items 1, 2, 3, and 4. Item 1 is a wide rectangle that spans the first three columns. The second row contains items 5, 6, 7, 8, 9, and 10. The third row contains items 11, 12, 13, 14, 15, and 16.

1			2	3	4
5	6	7	8	9	10
11	12	13	14	15	16



Thuộc tính của các Grid Items

❑ Các thuộc tính của Grid Items:

◆ Thuộc tính **grid-row**:

- Là một thuộc tính viết tắt thuộc tính **grid-row-start** và thuộc tính **grid-row-end**.
- Để đặt 1 item trong grid-container, bạn có thể sử dụng thông số line numbers hoặc dùng từ khóa “span” để thiết lập vị trí item sẽ kéo dài bao nhiêu dòng.

1	2	3	4	5	6
	7	8	9	10	11
	12	13	14	15	16



Thuộc tính của các Grid Items

❑ Các thuộc tính của Grid Items:

◆ Thuộc tính **grid-row**:

- Ví dụ tạo “item1” bắt đầu trên đường kẻ dòng 1 và kết thúc ở đường kẻ dòng 4:

```
<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
  <div class="item7">7</div>
  <div class="item8">8</div>
  <div class="item9">9</div>
  <div class="item10">10</div>
  <div class="item11">11</div>
  <div class="item12">12</div>
  <div class="item13">13</div>
  <div class="item14">14</div>
  <div class="item15">15</div>
  <div class="item16">16</div>
</div>
```

```
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto auto auto auto;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}

.item1 {
  grid-row: 1 / 4;
}
</style>
```

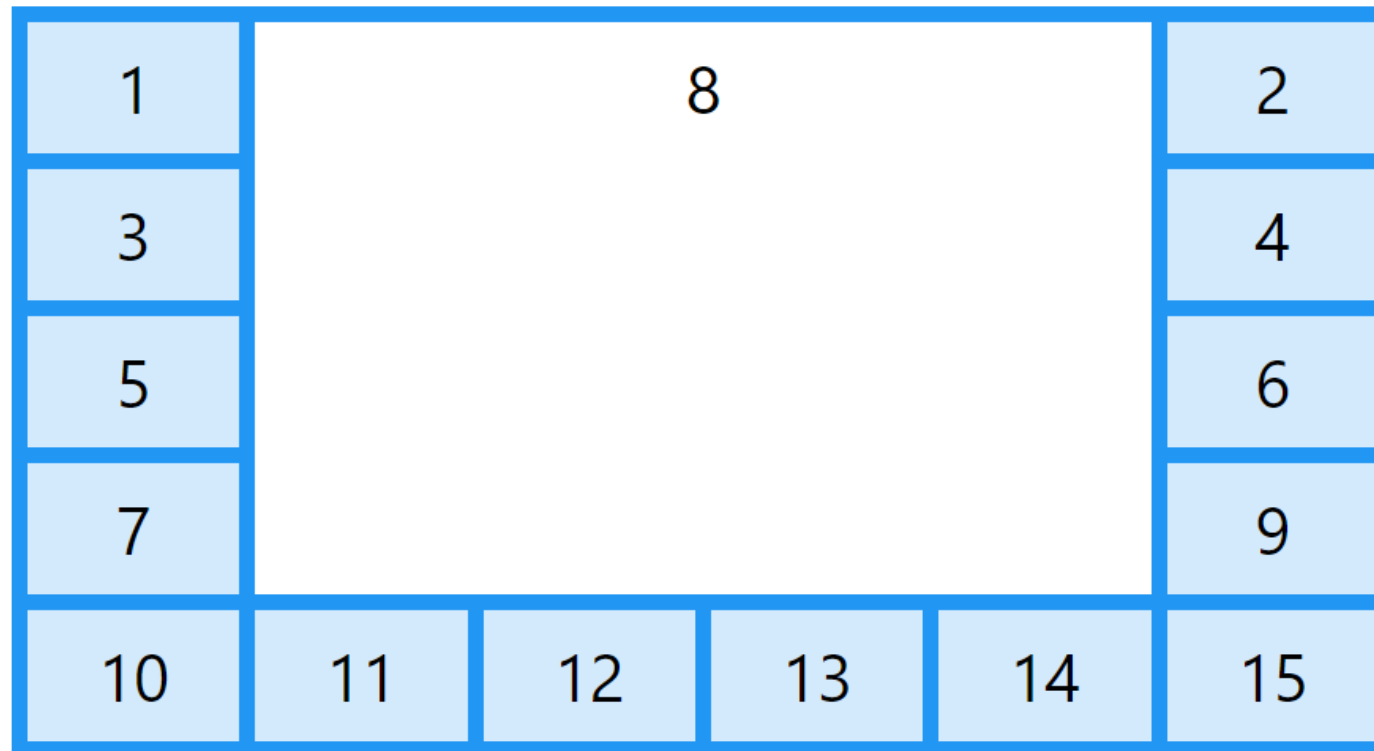


Thuộc tính của các Grid Items

❑ Các thuộc tính của Grid Items:

◆ Thuộc tính **grid-area**:

- Là một thuộc tính viết tắt cho thuộc tính **grid-row-start** và thuộc tính **grid-column-start**, **grid-row-end** và **grid-column-end**.





Thuộc tính của các Grid Items

❑ Các thuộc tính của Grid Items:

◆ Thuộc tính **grid-area**:

- Ví dụ tạo item8 bắt đầu trên row-line 1 và column-line 2, và kết thúc trên row-line 5 và column-line 6.

```
<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
  <div class="item7">7</div>
  <div class="item8">8</div>
  <div class="item9">9</div>
  <div class="item10">10</div>
  <div class="item11">11</div>
  <div class="item12">12</div>
  <div class="item13">13</div>
  <div class="item14">14</div>
  <div class="item15">15</div>
</div>
```

```
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto auto auto auto;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}

.item8 {
  grid-area: 1 / 2 / 5 / 6;
}
</style>
```

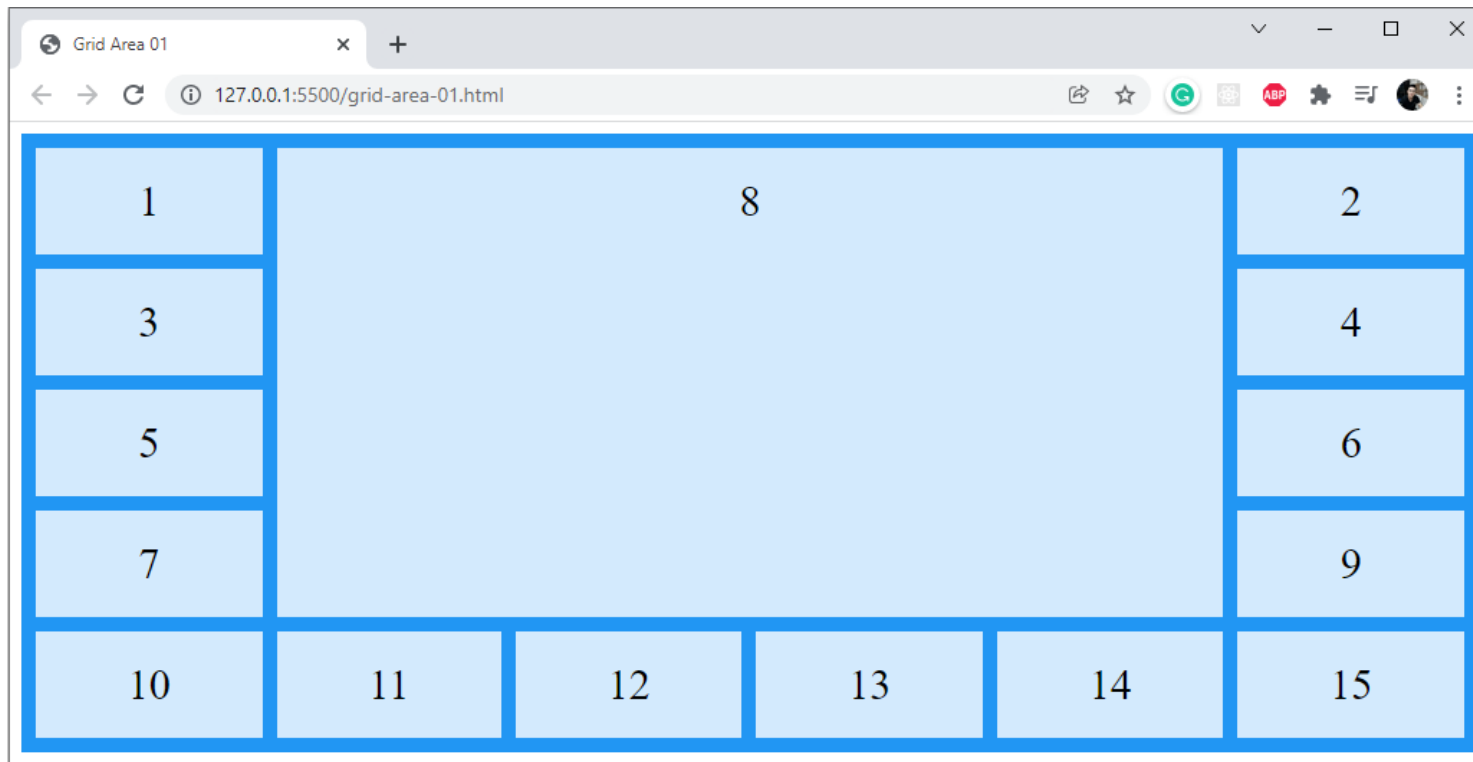


Thuộc tính của các Grid Items

❑ Các thuộc tính của Grid Items:

◆ Thuộc tính **grid-area**:

- Ví dụ tạo item8 bắt đầu trên row-line 1 và column-line 2, và kết thúc trên row-line 5 và column-line 6.





Thuộc tính của các Grid Items

❑ Các thuộc tính của Grid Items:

◆ Thuộc tính **grid-area**:

- Ví dụ tạo item8 bắt đầu trên row-line 2 và column-line 1, và kéo dài span thêm 2 rows và 3 columns.

```
<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
  <div class="item7">7</div>
  <div class="item8">8</div>
  <div class="item9">9</div>
  <div class="item10">10</div>
  <div class="item11">11</div>
  <div class="item12">12</div>
  <div class="item13">13</div>
</div>
```

```
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto auto auto auto;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}

.item8 {
  grid-area: 2 / 1 / span 2 / span 3;
}
</style>
```



Thuộc tính của các Grid Items

❑ Các thuộc tính của Grid Items:

◆ Thuộc tính **grid-area**:

- Ví dụ tạo item8 bắt đầu trên row-line 2 và column-line 1, và kéo dài span thêm 2 rows và 3 columns.

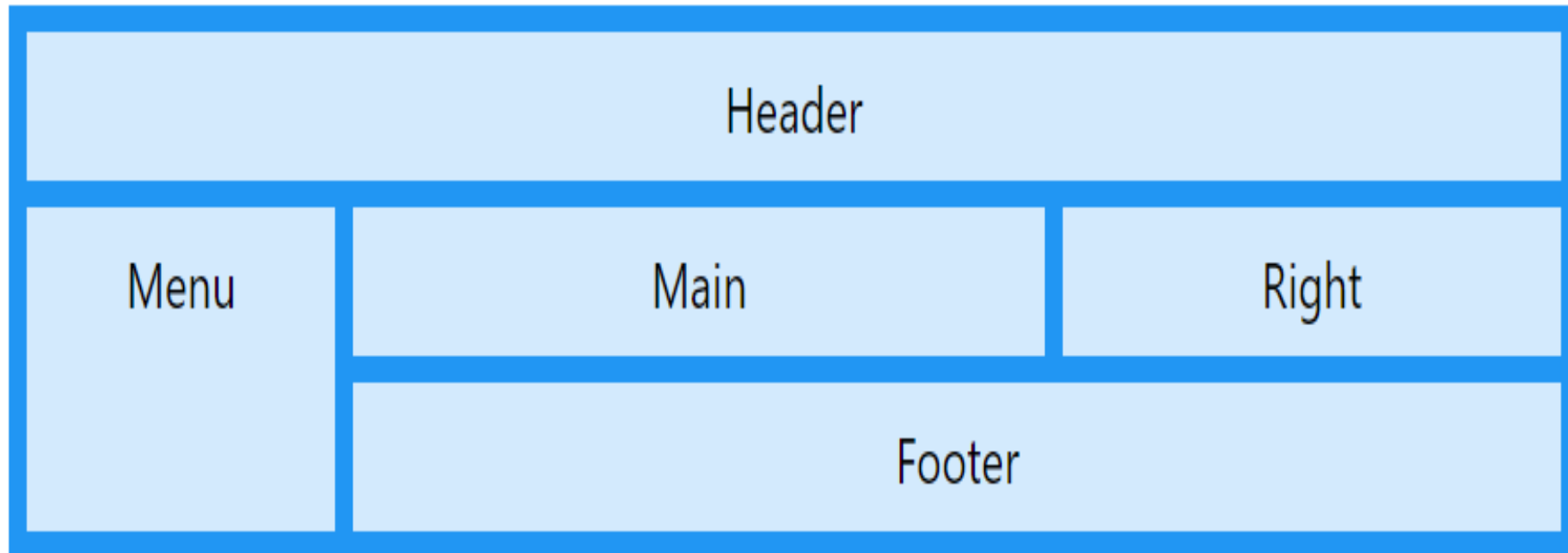
1	2	3	4	5	6
8			7	9	10
			11	12	13



Đặt tên cho các Grid Items

❑ Đặt tên các Grid Items:

- ◆ Thuộc tính **grid-area** và thuộc tính **grid-template-areas**:
 - Thuộc tính **grid-areas** còn dùng để đặt tên cho các **grid-items**.
 - Các **grid-items** cũng có thể được đặt tên bởi thuộc tính **grid-template-areas** của một **grid container**.





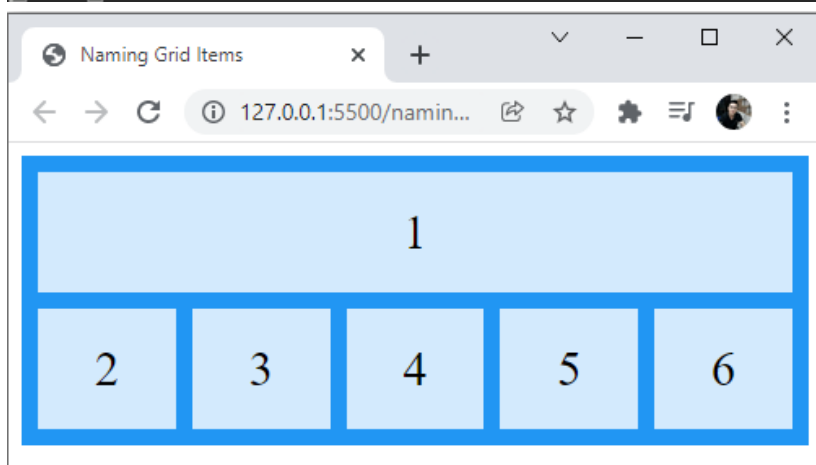
Đặt tên cho các Grid Items

❑ Đặt tên các Grid Items:

◆ Thuộc tính **grid-area** và thuộc tính **grid-template-areas**:

▪ Ví dụ item1 được đặt tên là “myArea” và độ rộng span 5 columns trong tất cả 5 columns của grid layout:

```
<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
</div>
```



```
<style>
.item1 {
  grid-area: myArea;
}

.grid-container {
  display: grid;
  grid-template-areas: 'myArea myArea myArea myArea myArea';
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}
</style>
```

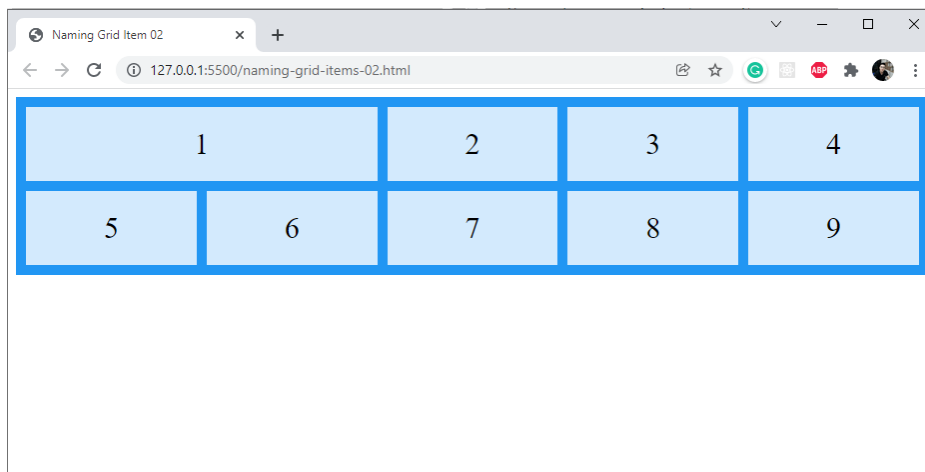


Đặt tên cho các Grid Items

❑ Đặt tên các Grid Items:

- ◆ Thuộc tính **grid-area** và thuộc tính **grid-template-areas**:
 - Ví dụ myArea span 2 columns trong tất cả 5 columns của grid layout:

```
<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
  <div class="item7">7</div>
  <div class="item8">8</div>
  <div class="item9">9</div>
</div>
```



```
<style>
  .item1 {
    grid-area: myArea;
  }

  .grid-container {
    display: grid;
    grid-template-areas: 'myArea myArea . . .';
    grid-gap: 10px;
    background-color: #2196F3;
    padding: 10px;
  }

  .grid-container > div {
    background-color: rgba(255, 255, 255, 0.8);
    text-align: center;
    padding: 20px 0;
    font-size: 30px;
  }
</style>
```



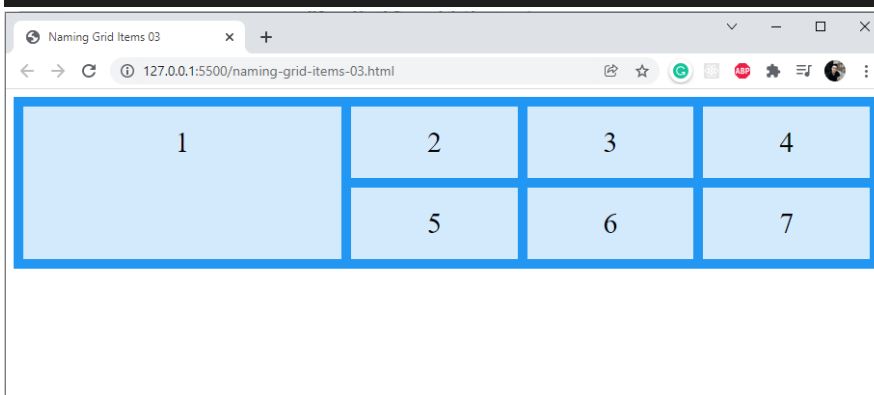
Đặt tên cho các Grid Items

❑ Đặt tên các Grid Items:

◆ Thuộc tính **grid-area** và thuộc tính **grid-template-areas**:

▪ Mỗi dòng được **định nghĩa trong cặp dấu nháy đơn (' ')** và mỗi cột trong 1 dòng được định nghĩa bên trong cặp nháy đơn đó và phân cách nhau bởi 1 space

```
<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
  <div class="item7">7</div>
</div>
```



```
<style>
.item1 {
  grid-area: myArea;
}

.grid-container {
  display: grid;
  grid-template-areas: 'myArea myArea . . .' 'myArea myArea . . .';
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}
</style>
```

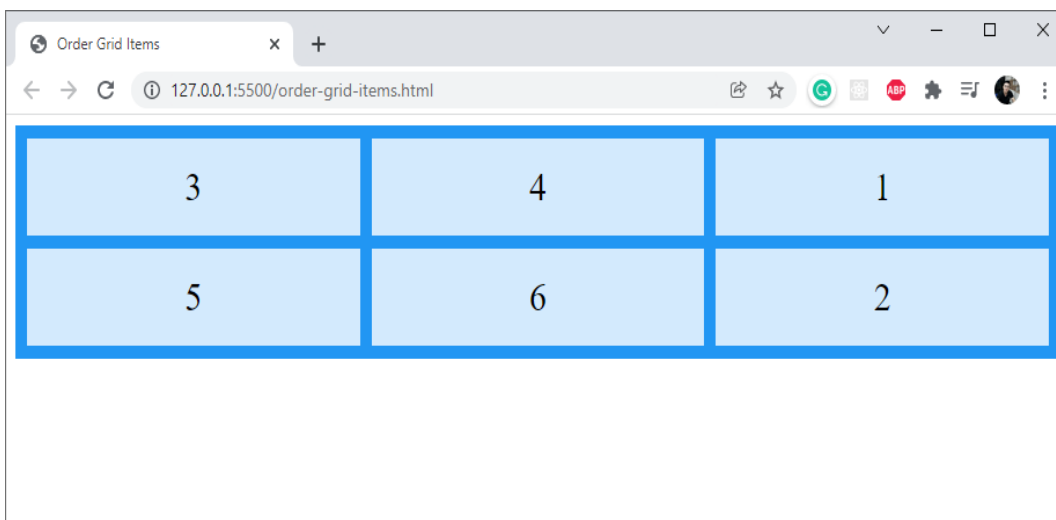


Thứ tự của các Grid Items

❑ Thứ tự các Grid Items:

- ♦ Grid Layout **cho phép** chúng ta **định vị các items** ở **bất cứ đâu** chúng ta muốn.

```
<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
</div>
```



```
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}

.item1 { grid-area: 1 / 3 / 2 / 4; }
.item2 { grid-area: 2 / 3 / 3 / 4; }
.item3 { grid-area: 1 / 1 / 2 / 2; }
.item4 { grid-area: 1 / 2 / 2 / 3; }
.item5 { grid-area: 2 / 1 / 3 / 2; }
.item6 { grid-area: 2 / 2 / 3 / 3; }
</style>
```




Responsive Web với Grid Layout

❑ Thứ tự các Grid Items:

♦ Bạn **có thể sắp xếp lại thứ tự cho các kích thước màn hình nhất định**, bằng cách sử dụng các **media queries**. (Size > 500 pixels)

```
<div class="grid-container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
  <div class="item6">6</div>
</div>
```



```
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}

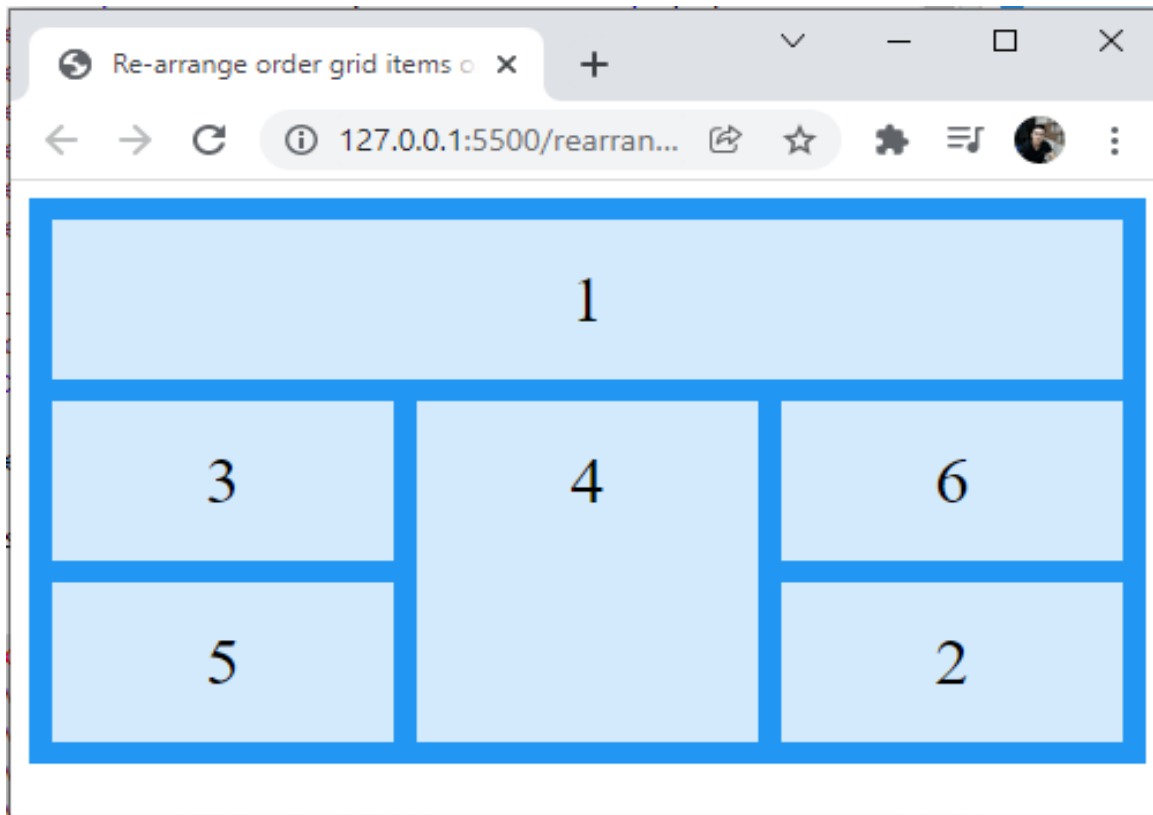
@media only screen and (max-width: 500px) {
  .item1 { grid-area: 1 / span 3 / 2 / 4; }
  .item2 { grid-area: 3 / 3 / 4 / 4; }
  .item3 { grid-area: 2 / 1 / 3 / 2; }
  .item4 { grid-area: 2 / 2 / span 2 / 3; }
  .item5 { grid-area: 3 / 1 / 4 / 2; }
  .item6 { grid-area: 2 / 3 / 3 / 4; }
}
</style>
```



Responsive Web với Grid Layout

❑ Thứ tự các Grid Items:

♦ Bạn **có thể sắp xếp lại thứ tự cho các kích thước màn hình nhất định**, bằng cách sử dụng các **media queries**. (Size ≤ 500 pixels)



```
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}

@media only screen and (max-width: 500px) {
  .item1 { grid-area: 1 / span 3 / 2 / 4; }
  .item2 { grid-area: 3 / 3 / 4 / 4; }
  .item3 { grid-area: 2 / 1 / 3 / 2; }
  .item4 { grid-area: 2 / 2 / span 2 / 3; }
  .item5 { grid-area: 3 / 1 / 4 / 2; }
  .item6 { grid-area: 2 / 3 / 3 / 4; }
}
</style>
```



Tổng kết nội dung bài học

- ☐ Tổng quan về Grid Layout
- ☐ Thành phần của Grid Layout
- ☐ Thuộc tính của Grid Container
- ☐ Thuộc tính của các Grid Items
- ☐ Đặt tên cho các Grid Items
- ☐ Thứ tự của các Grid Items
- ☐ Responsive Web với Grid Layout

Let's
Recap

