



TRƯỜNG ĐẠI HỌC VINH
VIỆN KỸ THUẬT VÀ CÔNG NGHỆ

CHƯƠNG 1

HTML5, CSS3, BOOTSTRAP4

Nghệ An, 2020

Nội dung giảng dạy

- 1.1. Giới thiệu về HTML5
- 1.2. Các phần tử mới trong HTML5
- 1.3. Các phần tử ngữ nghĩa trong HTML5
- 1.4. Chuyển đổi mã HTML4 thành HTML5
- 1.5. Đồ họa trong HTML5
- 1.6. Giới thiệu về CSS3
- 1.7. CSS3 cơ bản
- 1.8. CSS3 nâng cao
- 1.9. Giới thiệu về Bootstrap
- 1.10. Hệ thống lưới trong Bootstrap
- 1.11. Các thành phần trong Bootstrap



1.1. Giới thiệu về HTML5

- HTML5 được phát hành chính thức vào năm 2014
- Là phiên bản lớn tiếp theo của HTML thay thế cho HTML 4.01, XHTML 1.0, và XHTML 1.1
- Là một chuẩn để cấu trúc và hiện diện nội dung trên WWW, chuẩn mới này kết hợp các tính năng như Video Playback, drag-and-drop... mà trước đây đã phụ thuộc vào các plug-ins như Adobe Flash, Microsoft Silverlight...
- Được hỗ trợ trên các trình duyệt phổ biến hiện nay, bao gồm các trình duyệt cho thiết bị di động
- Bổ sung thêm phần tử mới, loại bỏ một số phần tử cũ





1.1. Giới thiệu về HTML5

- Lý do sử dụng HTML5
 - HTML5 mạnh mẽ hơn và dễ dàng sử dụng hơn HTML4.
 - Cho phép phát âm thanh và video trên trình duyệt.
 - Có thể vẽ các hình dạng như hình tròn, hình chữ nhật, hình tam giác. Điều không thể thực hiện được trong HTML phiên bản trước.
 - Có thể sử dụng trực tiếp thẻ (header) đầu trang và chân trang (footer) để xác định phần đầu trang và chân trang.
 - HTML5 cho phép chạy mã JavaScript trong nền sử dụng Web workers.
 - Nó cung cấp bộ nhớ cục bộ (local storage) thay cho cookie.



1.1. Giới thiệu về HTML5

- Các phần tử mới:
 - Các phần tử ngữ nghĩa: header, footer, article, section.
 - Các thuộc tính mới của các phần tử form: number, date, time, calendar, range
 - Các phần tử đồ họa: svg, canvas
 - Các phần tử đa phương tiện: audio, video
 - ...



1.1. Giới thiệu về HTML5

- Các API (Application Programming Interface) mới:
 - HTML Geolocation: HTML vị trí địa lý
 - HTML Drag and Drop: HTML kéo và thả
 - HTML Local Storage: HTML lưu trữ cục bộ, là một sự thay thế tốt cho các Cookie
 - HTML Application Cache: HTML Cache ứng dụng
 - HTML Web Workers: Các HTML Web Worker – Web Worker là một đoạn mã JavaScript chạy ở nền tảng trình duyệt, độc lập và không ảnh hưởng tới các đoạn mã khác
 - HTML SSE (Server-Sent Event): Cho phép một trang Web nhận cập nhật từ Server



1.1. Giới thiệu về HTML5

- Cấu trúc chuẩn một trang xây dựng bằng HTML5

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <title>Tiêu đề tài liệu</title>
</head>

<body>
  Nội dung tài liệu
</body>

</html>
```

1.2. Các phần tử mới trong HTML5

- Các phần tử cấu trúc/ngữ nghĩa mới

Thẻ	Mô tả
<article>	Định nghĩa một bài viết trong tài liệu
<aside>	Định nghĩa một bên nội dung trong nội dung trang
<bdi>	Cô lập một phần văn bản có thể được định dạng khác với các văn bản khác
<details>	Định nghĩa các chi tiết thêm để người dùng có thể xem hoặc ẩn đi
<dialog>	Định nghĩa một hộp thoại hoặc cửa sổ
<figcaption>	Định nghĩa một nhãn cho phần tử <figure>
<figure>	Định nghĩa nội dung tự chứa
<footer>	Định nghĩa một phần chân (footer) cho tài liệu hoặc 1 phần nội dung nào đó
<header>	Định nghĩa một phần đầu (header) cho tài liệu hoặc 1 phần nội dung nào đó
<main>	Định nghĩa nội dung chính của tài liệu

1.2. Các phần tử mới trong HTML5

- Các phần tử cấu trúc/ngữ nghĩa mới

Thẻ	Mô tả
<mark>	Định nghĩa văn bản được đánh dấu/tô sáng
<menuitem>	Định nghĩa một phần tử menu/command mà người dùng có thể chọn từ một popup menu (menu bật lên)
<meter>	Định nghĩa một đơn vị đo lường trong khoảng biết trước
<nav>	Định nghĩa các liên kết điều hướng
<progress>	Thẻ hiện tiến trình của 1 tác vụ
<rp>	Định nghĩa cái gì cần hiển thị trong trình duyệt mà không hỗ trợ các chú thích ruby
<rt>	Định nghĩa một giải thích/phát âm của các ký tự (dành cho kiểu chữ Đông Á)
<ruby>	Định nghĩa một chú thích ruby (dành cho kiểu chữ Đông Á)
<section>	Định nghĩa một phần trong 1 tài liệu
<summary>	Định nghĩa phần đầu có thể nhìn thấy cho một phần tử <details>

1.2. Các phần tử mới trong HTML5

- Các dạng nhập liệu mới
 - Color: nhập màu sắc
 - Date: nhập ngày tháng
 - Datetime
 - Datetime-local
 - Email: nhập email
 - Month: điều khiển chọn ngày tháng
 - Number: số
 - Range: dải số
 - Search: hộp tìm kiếm
 - Tel: nhập số điện thoại
 - Time: thời gian
 - url: nhập địa chỉ url
 - Week: điều khiển chọn tuần

1.2. Các phần tử mới trong HTML5

```
<form autocomplete="off">
  <label for="email">Nhập địa chỉ email:</label>
  <input type="text" name="email" placeholder="email@example.com" autofocus required/>

  <input type="text" list="colors">
  <datalist id="colors">
    <option value="Red"></option>
    <option value="Green"></option>
    <option value="Yellow"></option>
  </datalist>

  <input type="submit" value="Submit">
</form>
```

Ví dụ

1.2. Các phần tử mới trong HTML5

- Một số thuộc tính nhập liệu mới:
 - Placeholder: cung cấp thông tin gợi ý khi người dùng nhập dữ liệu
 - Autofocus: tự động kích hoạt phần tử khi load trang
 - Required: bắt buộc nhập dữ liệu cho phần tử (nếu để trong form thì form không submit được)
 - Autocomplete: cho biết các phần tử trong form có tự động điền dữ liệu vào hay không
 - List: danh sách các gợi ý khi tìm kiếm

1.2. Các phần tử mới trong HTML5

- Các thuộc tính nhập liệu mới:
 - form
 - formaction
 - formenctype
 - formmethod
 - formnovalidate
 - formtarget
 - height và width
 - min và max
 - multiple
 - pattern (regexp)
 - step

1.2. Các phần tử mới trong HTML5

- Các thuộc tính Custom (tùy chỉnh)
 - Là các thuộc tính mà developer định nghĩa riêng. Đây là một tính năng mới được giới thiệu trong HTML5.
 - Một thuộc tính custom bắt đầu với data- và sẽ được đặt tên dựa trên yêu cầu của developer.

```
<div class="example" data-subject="C" data-level="1">  
    ...  
</div>
```



1.3. Các phần tử ngữ nghĩa trong HTML5

- Một phần tử ngữ nghĩa (semantic) mô tả ý nghĩa của nó đối với trình duyệt và nhà phát triển.
 - Những phần tử không phải semantic như: div và span. Hai phần tử này không có thể hiện ý nghĩa nào khi nhắc đến.
 - Các phần tử ngữ nghĩa như: form, table và article thì lại mô tả đầy đủ ý nghĩa. Table làm hình dung tới là 1 cái bảng biểu, form là 1 mẫu đơn, và article là 1 bài viết.



1.3. Các phần tử ngữ nghĩa trong HTML5

- Trước đây, khi xây dựng giao diện website, developer thường dùng như sau để mô tả phần thanh menu (navigation – điều hướng), phần tiêu đề, và phần chân trang thường khai báo như sau:
 - `<div id="nav">`, `<div class="header">`, `<div id="footer">`...
- Tuy nhiên, cách khai báo này làm các cơ chế tìm kiếm (Google) không thể nhận diện được nội dung và cấu trúc trang dễ dàng
- HTML5 cung cấp một số phần tử ngữ nghĩa giúp định nghĩa các phần trong một trang và giúp các search engine hiểu rõ hơn về cấu trúc trang web. Một số phần tử mới như:
 - `<article>`, `<aside>`, `<details>`, `<figcaption>`, `<figure>`, `<footer>`, `<header>`, `<main>`, `<mark>`, `<nav>`, `<section>`, `<summary>`, `<time>`...



1.3. Các phần tử ngữ nghĩa trong HTML5

- Thẻ `<main>`: thể hiện nội dung chính của tài liệu. Chỉ dùng một thẻ `<main>` trong tài liệu.
- Thẻ `<header>`: chứa phần nội dung đầu của khối hoặc tài liệu. Có thể có nhiều thẻ `<header>` trong 1 tài liệu.
- Thẻ `<nav>`: định nghĩa khu vực thiết lập menu điều hướng.
- Thẻ `<section>`: định nghĩa các khối (block) của trang web, chứa các nội dung chính.
- Thẻ `<aside>`: sử dụng cho vùng sidebar của website, định nghĩa các phần nội dung bên lề, không phải nội dung chính.
 - Thẻ `<aside>` thường sử dụng bên trong thẻ `<section>`, chứa các nội dung như quảng cáo, slide, thông tin bên lề...



1.3. Các phần tử ngữ nghĩa trong HTML5

- Một số ví dụ:

```
<article>
  <header>
    <h1>Thông tin về ĐHV</h1>
    <p>Chức năng:</p>
  </header>
  <p>Trường Đại học Vinh là trường đại học trọng điểm quốc gia...</p>
</article>
```

```
<nav>
  <ul>
    <li><a href="#">Trang chủ</a></li>
    <li><a href="#">Giới thiệu</a></li>
    <li><a href="#">Liên hệ</a></li>
  </ul>
</nav>
```

```
<section>
  <h1>LTW</h1>
  <p>Học phần Lập trình WEB</p>
</section>
```

```
<aside>
  <p>Quảng cáo</p>
  <p>Tin tức ngoài lề</p>
</aside>
```



1.3. Các phần tử ngữ nghĩa trong HTML5

- Thẻ `<article>`: là thành phần thường chứa nội dung một bài viết, tin tức...
 - Có thể lồng thẻ `<article>` vào trong thẻ `<section>`.
 - Thẻ `<article>` có thể chứa các thẻ `<header>`, `<footer>`, `<section>`
- Thẻ `<footer>` chỉ rõ phần nội dung chân trang của website. Có thể có nhiều thẻ `<footer>` trong 1 trang web.
 - Nội dung bên trong footer như: địa chỉ liên lạc, email, số điện thoại, thông tin chung website...
- Thẻ `<figure>`, `<figcaption>`: là giải thích có thể thấy được cho một hình ảnh, một hình ảnh và một nhãn có thể gom nhóm trong phần tử figure
 - Ví dụ sau `img` là phần tử định nghĩa hình ảnh, và phần tử `figcaption` định nghĩa nhãn mô tả hình ảnh đó.

1.3. Các phần tử ngữ nghĩa trong HTML5

- Một số ví dụ:

```
<article>
  <h1>Tin tức trong ngày</h1>
  <p>Nội dung tin tức</p>
</article>
```

```
<footer>
  <p>Email: test.vinhuni@gmail.com</p>
  <p>SDT: 0123 456 789</p>
  <p>Địa chỉ: 182 Lê Duẩn</p>
</footer>
```

```
<figure>
  
  <figcaption>Hình ảnh đẹp</figcaption>
</figure>
```



1.3. Các phần tử ngữ nghĩa trong HTML5

- Thẻ `<video>`: nhúng video vào trang web mà không cần dùng plugin của trình duyệt (chỉ hỗ trợ 3 dạng video sau: MP4, WebM, Ogg). Các thuộc tính bao gồm:
 - controls: thanh điều khiển
 - poster: ảnh đại diện
 - autoplay: tự động chạy
 - width, height: chiều rộng, cao
 - muted: tắt âm lượng
 - loop: tự động lặp lại
- Thẻ `<audio>`: nhúng file âm vào trang web mà không cần dùng plugin của trình duyệt (chỉ hỗ trợ 3 dạng audio sau: MP3, Wav, Ogg). Các thuộc tính bao gồm:
 - Controls, autoplay, loop, muted
- Có thể chèn video trên youtube bằng cách sử dụng thẻ `<iframe>`

1.3. Các phần tử ngữ nghĩa trong HTML5

- Một số ví dụ:

```
<video poster="image/1_html5_video.jpg" controls>  
  <source src="video/Jack_la_mot_thang_con_trai.mp4">  
</video>
```

```
<audio controls>  
  <source src="audio/Em-Gi-Oi-Jack-K-ICM.mp3" type="audio/mp3">  
</audio>
```

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/FN7A  
LfpGxiI" frameborder="0" allow="accelerometer; autoplay; encrypted-  
media; gyroscope; picture-in-picture" allowfullscreen>  
</iframe>
```



1.3. Các phần tử ngữ nghĩa trong HTML5

- Thẻ <progress>: hiển thị quá trình của công việc. Mục đích chính là hiển thị tiến trình upload file.
- Thẻ <meter>: hiển thị dữ liệu trên ổ đĩa đã được sử dụng mức độ như thế nào so với tổng dung lượng.
- Thẻ <detail>: xác định thông tin chi tiết của một nội dung trên web, có thể ẩn/hiện
- Thẻ <mark>: đánh dấu nội dung
- Thẻ <dialog>: hiển thị một hộp thoại hoặc cửa sổ con
 - Ví dụ: [1](#)

1.3. Các phần tử ngữ nghĩa trong HTML5

- Một số ví dụ:

```
<progress value="40" max="100" style="height: 40px;"></progress>
```

```
<meter value="0.7">70%</meter>
```

```
<details>  
<summary>VinhUni:</summary>  
<p>Vinh University</p>  
</details>
```

```
<p>Html is very <mark>simple</mark> to use and it is also <mark>easy</mark>  
to learn.</p>
```

```
<dialog open="">This is an open dialog window</dialog>
```




1.3. Các phần tử ngữ nghĩa trong HTML5

- Một trang tài liệu HTML5 có thể sẽ có cấu trúc như sau:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>...</title>
  </head>
  <body>
    <header>...</header>
    <nav>...</nav>
    <article>
      <section>
        ...
      </section>
    </article>
    <aside>...</aside>
    <footer>...</footer>
  </body>
</html>
```

Ví dụ

1.4. Chuyển đổi mã HTML4 thành HTML5

- Sử dụng quy tắc sau để chuyển đổi mã HTML4 thành HTML5 (Ví dụ minh họa: [HTML4](#) → [HTML5](#))

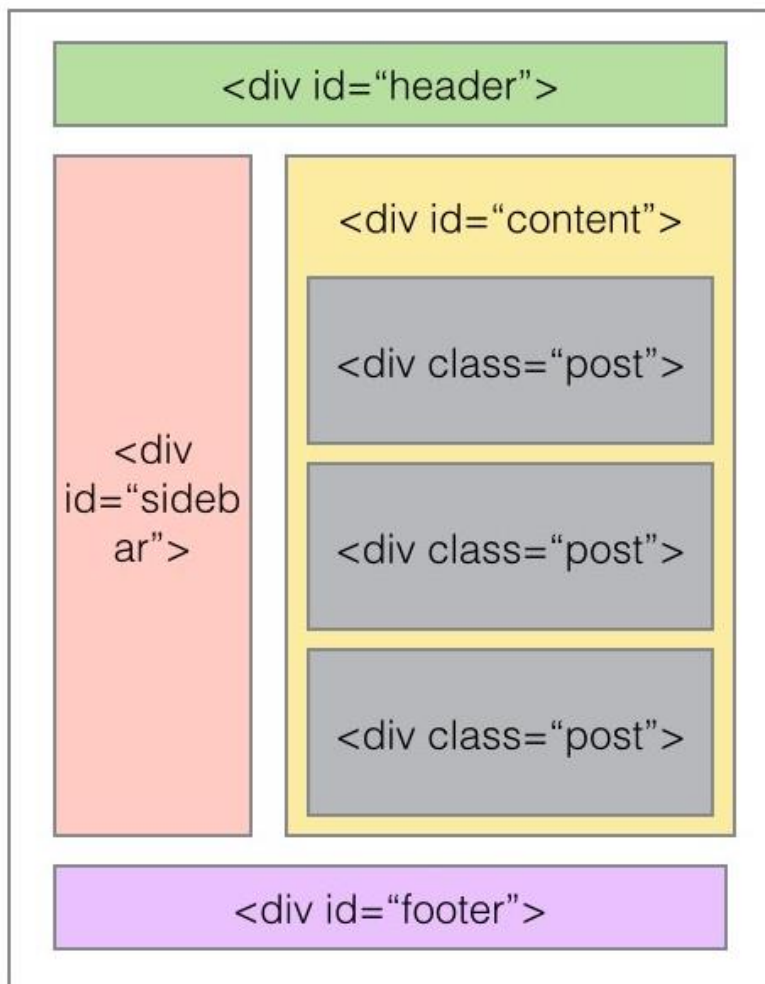
HTML4	HTML5
<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"></code>	<code><!DOCTYPE html></code>
<code><meta http-equiv="Content-Type" content="text/html; charset=utf-8"></code>	<code><meta charset="utf-8"></code>
<code><div id="header"></code>	<code><header></code>
<code><div id="menu"></code>	<code><nav></code>
<code><div id="content"></code>	<code><section></code>
<code><div class="article"></code>	<code><article></code>
<code><div id="footer"></code>	<code><footer></code>
...	...



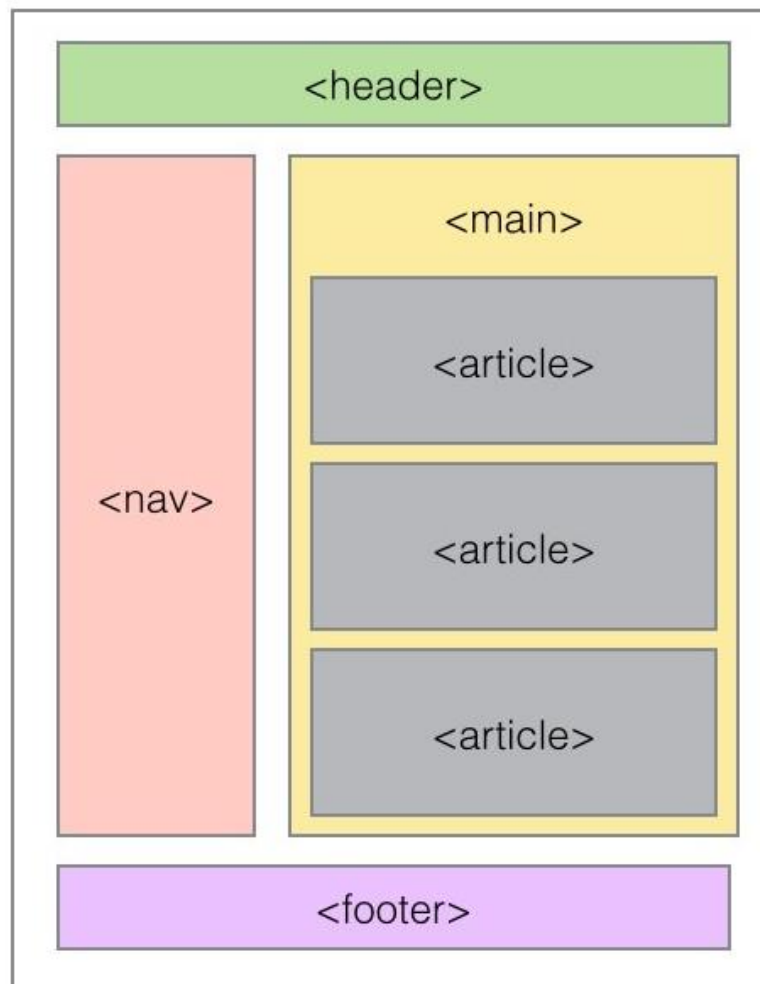


1.4. Chuyển đổi mã HTML4 thành HTML5

HTML4: Lots of Classes/IDs



HTML5: Semantic Tags/Sections





1.4. Chuyển đổi mã HTML4 thành HTML5

- Sử dụng quy tắc sau để chuyển đổi mã HTML4 thành HTML5. Ví dụ: có một file [HTML4](#)
 - Chuyển [CSS HTML4](#) → [CSS HTML5](#)
 - Chuyển Code HTML4 → [Code HTML5](#)
 - Ví dụ khác về HTML5: 1, 2
- Một số website về convert HTML 4 → HTML5 : [1](#), [2](#)
- Website kiểm tra Code HTML và CSS:
 - [HTML](#)
 - [CSS](#)

1.5. Đồ họa trong HTML5: <canvas>

- Thẻ <canvas> được dùng để vẽ đối tượng đồ họa trong một trang web, thông qua JavaScript. Phần tử canvas là vùng chứa cho đối tượng đồ họa.

```
<!-- Vẽ hình chữ nhật -  
<canvas id="myCanvas" width="300" height="400" style="border: 2px solid  
red;"></canvas>
```

```
<!-- Vẽ đường thẳng -->  
<canvas id="line"></canvas>  
<script>  
  var c=document.getElementById("line");  
  var ct = c.getContext("2d");//Vẽ hình 2D  
  ct.moveTo(50, 50);//Điểm bắt đầu  
  ct.lineTo(200, 100);//Điểm kết thúc  
  ct.lineTo(200, 300);//Điểm kết thúc  
  ct.lineWidth = 10;  
  ct.strokeStyle = "blue";  
  ct.stroke();//Vẽ  
</script>
```



1.5. Đồ họa trong HTML5: <canvas>

```
<!-- Vẽ hình tròn -->
<!--
arc(x, y, r, rc1, rc2) để vẽ cung tròn, x, y là tọa độ tâm; r là bán kính,
rc1 và rc2 là vẽ cung tròn từ độ dài chu vi rc1 đến rc2-->
<canvas id="circle"></canvas>
<script>
    var c = document.getElementById("circle");
    var ctx = c.getContext("2d");
    ctx.beginPath();
    ctx.arc(100, 100, 80, 0, 2 * Math.PI);
    ctx.stroke();
</script>
```

```
<!-- Vẽ text -->
<canvas id="text"></canvas>
<script>
    var c = document.getElementById("text");
    var ct = c.getContext("2d");
    ct.font = "30px Arial"; //Định dạng font chữ
    ct.fillText("Xin chào", 30, 90); //Vẽ text không viền
    ct.strokeText("Xin chào", 50, 50); //Vẽ text có viền
</script>
```

1.5. Đồ họa trong HTML5: <canvas>

```
<!-- Vẽ hình chữ nhật gradient -->
<canvas id="myCanvas"></canvas>
<script>
    var c = document.getElementById("myCanvas");
    var ctx = c.getContext("2d");
    var grd = ctx.createLinearGradient(0, 100, 200, 0);
    grd.addColorStop(0, "red");
    grd.addColorStop(1, "white");
    ctx.fillStyle = grd;
    ctx.fillRect(10, 10, 150, 80);
</script>
```

```
<!-- Dịch chuyển tọa độ -->
<canvas id="coordinate" width="800"></canvas>
<script>
    var ctx = document.getElementById('coordinate').getContext('2d');
    ctx.font = '48px serif'; ctx.fillStyle = 'red';
    ctx.fillText('Hello world', 20, 50);
    ctx.translate(300, 100);
    ctx.rotate(-Math.PI/4); ctx.scale(0.5, 0.5);
    ctx.fillText('Hello world', 0, 0);
</script>
```

1.5. Đồ họa trong HTML5: <canvas>

- Một số ví dụ đơn giản khác:
 - [Đường cong bậc 2](#)
 - [Nối nhiều đường](#)
 - [Vẽ cờ Việt Nam](#)
- Một số ví dụ phức tạp:
 - [Paint](#)
 - [Game](#)

1.5. Đồ họa trong HTML5: <svg>

- SVG (Scalable Vector Graphics, là một loại ảnh vector) là một ngôn ngữ đánh dấu dùng để định nghĩa các đối tượng đồ họa trên web. Phần tử SVG, tương tự như Canvas là một vùng chứa cho các đối tượng đồ họa.

```
<!-- Vẽ hình tròn -->
<svg width="250" height="250">
  <circle cx="100" cy="100" r="90" stroke="blue" stroke-
width="3" fill="red" />
</svg>
```

```
<!-- Vẽ hình chữ nhật -->
<svg width="400" height="200">
  <rect width="400" height="200" style="fill:pink;stroke-
width:15;stroke:blue" />
</svg>
```

1.5. Đồ họa trong HTML5: <svg>

- Ưu điểm
 - Có thể phóng to nhưng không vỡ ảnh.
 - Có thể được in với chất lượng cao ở bất kỳ độ phân giải nào.
 - Có thể được tạo và chỉnh sửa bằng javascript.
 - Có thể tìm kiếm, đánh chỉ mục, nén...
 - Có thể chuyển động sử dụng các thành phần animation đã xây dựng sẵn.
 - Có thể chứa liên kết đến bất kỳ tài liệu nào khác.
 - Hầu như được hỗ trợ trên tất cả các trình duyệt

1.5. Đồ họa trong HTML5: <svg>

```
<!-- Vẽ đường thẳng -->
<svg width="220" height="60">
  <line x1="30" y1="10" x2="200" y2="50" style="stroke: green; stroke-
linecap:round; stroke-width:20;" />
</svg>
```

```
<!-- Vẽ đường gấp khúc -->
<svg width="130" height="160">
  <polyline points="0,40 40,40 40,80 80,80 80,120 120,120 120,160" fil
l="#F9F38C" stroke="green" stroke-width="5" />
</svg>
```

```
<!-- Vẽ ngôi sao -->
<svg width="300" height="200">
  <polygon points="100,10 40,198 190,78 10,78 160,198"
style="fill:violet;stroke:brown;stroke-width:5;fill-
rule:evenodd;" />
</svg>
```

1.5. Đồ họa trong HTML5: <svg>

```
<!-- Vẽ logo -->
<svg height="500" width="500">
  <defs>
    <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">
      <stop offset="0%" style="stop-color:red;stop-opacity:1" />
      <stop offset="100%" style="stop-color:blue;stop-opacity:1" />
    </linearGradient>
  </defs>
  <ellipse cx="200" cy="100" rx="175" ry="70" fill="url(#grad1)" />
  <text fill="#ffffff" font-size="45" font-
family="Verdana" x="100" y="116">Vinh Uni</text>
</svg>
```

```
<!-- Vẽ ảnh động -->
<svg width="500" height="30">
  <rect width="20" height="20" fill="orange">
    <animate attributeName="x" from="0" to="300" dur="2s" fill="freeze" repeat
Count="indefinite" />
  </rect>
  <circle cx="0" cy="15" r="5" fill="green" stroke="yellow">
    <animate attributeName="cx" from="0" to="300" dur="5s" fill="freeze" repea
tCount="indefinite" />
  </circle>
</svg>
```

1.5. Đồ họa trong HTML5: <svg>

- So sánh Canvas và SVG:
 - SVG là ngôn ngữ để mô tả đồ họa 2D trong XML, trong khi đó, Canvas vẽ đồ họa 2D bằng ngôn ngữ JavaScript.

Canvas	SVG
<ul style="list-style-type: none">• Phụ thuộc độ phân giải• Không hỗ trợ các xử lý sự kiện• Khả năng hiển thị văn bản không cao (mờ)• Có thể lưu tập tin kết quả hình ảnh dưới dạng .png hay jpg• Thích hợp các trò chơi đồ họa chuyên sâu	<ul style="list-style-type: none">• Độc lập với độ phân giải• Hỗ trợ các xử lý sự kiện• Thích hợp cho các ứng dụng với vùng hiển thị lớn (Google Maps)• Chậm hiển thị nếu phức tạp• Không thích hợp cho các ứng dụng trò chơi

HTML5: Web Storage

- HTML5 giới thiệu hai kỹ thuật, tương tự như HTTP Cookie, để lưu giữ dữ liệu được cấu trúc trên Client-Side:
 - Session storage (sessionStorage):
 - Ý nghĩa: lưu giữ dữ liệu cho phiên làm việc duy nhất, nhưng có thể thực hiện nhiều công việc trong các cửa sổ khác nhau tại cùng một thời điểm.
 - Xóa Web Storage: đóng cửa sổ website
 - Local storage (localStorage):
 - Ý nghĩa: lưu giữ dữ liệu không thời hạn. Dữ liệu vẫn còn được lưu trữ khi đóng trang web hoặc trình duyệt, tùy thuộc vào giá trị thiết lập dung lượng lưu trữ cho trình duyệt của người dùng.
 - Xóa Web Storage: `localStorage.remove('key')` (`localStorage.clear()` để xóa tất cả local storage).



1.6. Giới thiệu về CSS3

- CSS3 (Cascading Style Sheet Level 3) là phiên bản mới nhất của CSS dùng để trang trí và giúp trang web nổi bật hơn.
- Được chia ra làm nhiều module và mỗi module sẽ có những nhiệm vụ khác nhau, khi kết hợp chúng với nhau sẽ tạo nên những hiệu ứng đẹp.
- CSS3 không phải là một thành phần của HTML5 nhưng lại có quan hệ mật thiết với HTML5, được phát triển song song với HTML5.
- Điểm khác nhau giữa CSS và CSS3: Media Querie, Namespace, Selector Level 3, Color.



1.6. Giới thiệu về CSS3

- Ưu điểm:
 - Giúp website chạy nhanh hơn do có rất nhiều mẫu layout có thể sử dụng CSS để thiết kế thay vì sử dụng background tốn rất nhiều tài nguyên để lưu trữ và tải.
 - CSS3 giúp tạo ra 1 số hiệu ứng đẹp và phong cách như transparent background, shadows, gradient... mà không cần sử dụng ảnh.
 - Tạo các animation mà không cần sử dụng flash.
 - Tù chỉnh layout và thiết kế website sao cho phù hợp với thiết bị của người dùng mà không sử dụng javascript.
 - Cung cấp rất nhiều thuộc tính mới giúp cho website sinh động và ít phải phụ thuộc vào javascript hơn.





1.6. Giới thiệu về CSS3

- CSS3 được tách nhỏ thành nhiều module và có luôn các module của CSS cũ, sau đây là các module:
 - Selectors
 - Box Model
 - Backgrounds and Borders
 - Image Values and Replaced Content
 - Text Effects
 - 2D/3D Transformations
 - Animations
 - Multiple Column Layout
 - User Interface



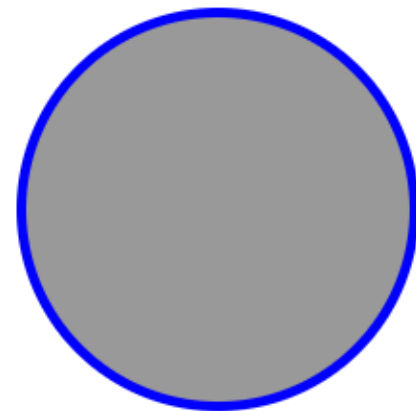
1.6. Giới thiệu về CSS3

- Một số thuộc tính mới trong CSS3:
 - CSS3 Rounded Corners
 - CSS3 Gradient
 - CSS3 Shadows
 - CSS3 Fonts
 - CSS3 2D, 3D Transform
 - CSS3 Animation
 - CSS3 Transition
 - CSS3 Flexbox
 - CSS3 Boxsizing
 - CSS3 Media

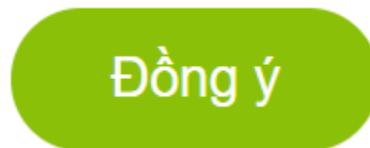
1.7. CSS3 cơ bản

- Border-Radius: bo góc tròn đường viền

```
.rounded-corners {  
  margin: 50px;  
  width: 200px;  
  height: 200px;  
  border: solid 5px blue;  
  background: #999;  
  border-radius: 50%;  
}
```



```
input {  
  padding: 10px 25px;  
  border-radius: 20px;  
  border: none;  
  background: #8AC007;  
  color: #fff;  
  cursor: pointer;  
}
```

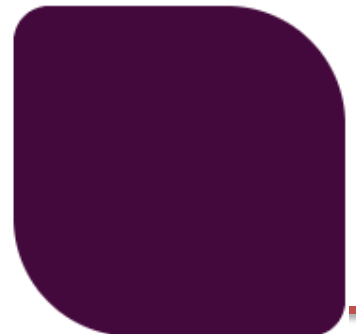




1.7. CSS3 cơ bản: Border-Radius

- Border-Radius: bo góc tròn đường viền

```
#rcorners1 {  
  border-radius: 15px 50px 30px 5px;  
  background: #12b952;  
  padding: 20px;  
  width: 100px;  
  height: 100px;  
}  
#rcorners2 {  
  border-radius: 15px 50px 30px;  
  background: #9ace0c;  
  padding: 20px;  
  width: 100px;  
  height: 100px;  
}  
#rcorners3 {  
  border-radius: 15px 50px;  
  background: #44093c;  
  padding: 20px;  
  width: 100px;  
  height: 100px;}
```

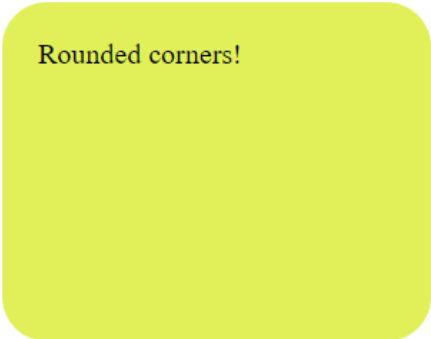


1.7. CSS3 cơ bản: Border-Radius

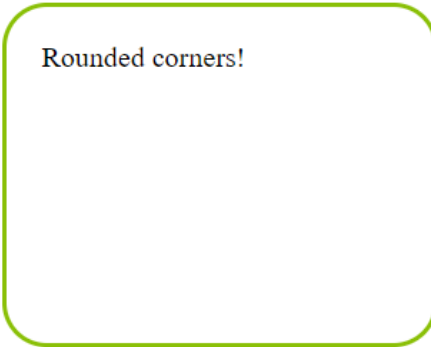
- Border-Radius: bo góc tròn đường viền

```
#rcorners1 {  
    border-radius: 25px;  
    background: rgb(225, 240, 89);  
    padding: 20px;  
    width: 200px;  
    height: 150px;  
}  
#rcorners2 {  
    border-radius: 25px;  
    border: 2px solid #8AC007;  
    padding: 20px;  
    width: 200px;  
    height: 150px;  
}  
#rcorners3 {  
    border-radius: 25px;  
    background: url(./image/5_html5_canvas.jpg);  
    padding: 20px;  
    width: 200px;  
    height: 150px;}
```

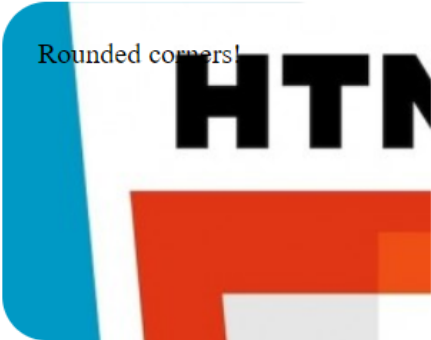
Rounded corners!



Rounded corners!



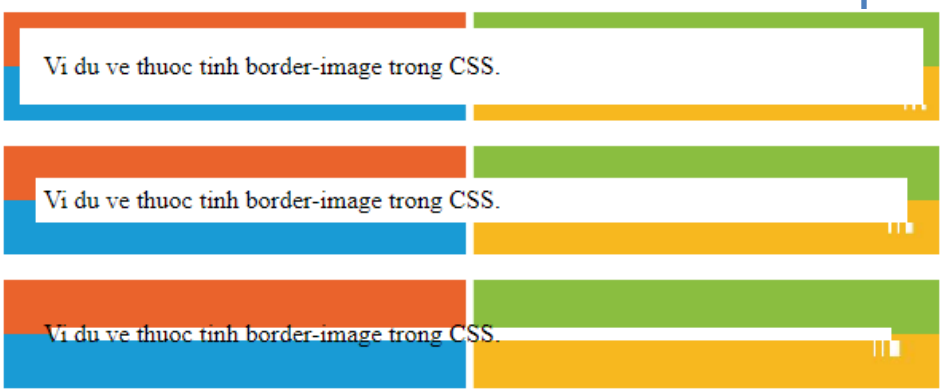
Rounded corners!



1.7. CSS3 cơ bản

- Border Image: tạo đường viền bằng hình ảnh

```
#borderimg1 {  
    border: 10px solid transparent;  
    padding: 15px;  
    border-image: url(./image/1_html5_figure.jpg) 30 round;  
    border-image-width: 10px;  
}  
  
#borderimg2 {  
    border: 10px solid transparent;  
    padding: 15px;  
    border-image: url(./image/1_html5_figure.jpg) 30 round;  
    border-image-width: 20px;  
}  
  
#borderimg3 {  
    border: 10px solid transparent;  
    padding: 15px;  
    border-image: url(./image/1_html5_figure.jpg) 30 round;  
    border-image-width: 30px;  
}
```



Ví dụ về thuộc tính border-image trong CSS.

Ví dụ về thuộc tính border-image trong CSS.

Ví dụ về thuộc tính border-image trong CSS.

1.7. CSS3 cơ bản: Border-Image

- Border Image: tạo đường viền bằng hình ảnh

```
div {  
  width: 108px;  
  height: 108px;  
  border-style: solid;  
  border-width: 27px;  
  border-image: url(./image/7_css3_border_image.png) 27 stretch;  
  (round, repeat)  
}
```



1.7. CSS3 cơ bản

- Multi Background: xử lý nâng cao background (thêm một hoặc nhiều hình ảnh tại một thời điểm).

Giá trị	Thuộc tính	Miêu tả
background		Thiết lập tất cả các thuộc tính về background
background-clip	border-box, padding-box, content-box	Khai báo một khu vực dành để Painting trong Background
background-image	url(link.png)	Xác định Background Image
background-origin	border-box, padding-box, content-box	Xác định vị trí của Background Image
background-size	contain, cover	Xác định kích cỡ của Background Image

1.7. CSS3 cơ bản: Background

- Multi Background: xử lý nâng cao background

```
#multibackground {  
    background-  
image: url(./image/7_css3_border_image.png), url(./image/5_html5_canvas.jpg);  
background-position: left top, left top;  
background-repeat: no-repeat, no-repeat;  
width: 400px; height: 400px;  
/* background-size: contain;  
Co giãn theo chiều rộng và chiều cao theo đúng tỉ lệ của hình ảnh */  
/* background-size: cover;  
Nếu chiều rộng và chiều cao của thẻ HTML lớn hơn hình ảnh thì nó sẽ giãn  
ra (full) */  
}
```



1.7. CSS3 cơ bản: Background

- Multi Background: xử lý nâng cao background

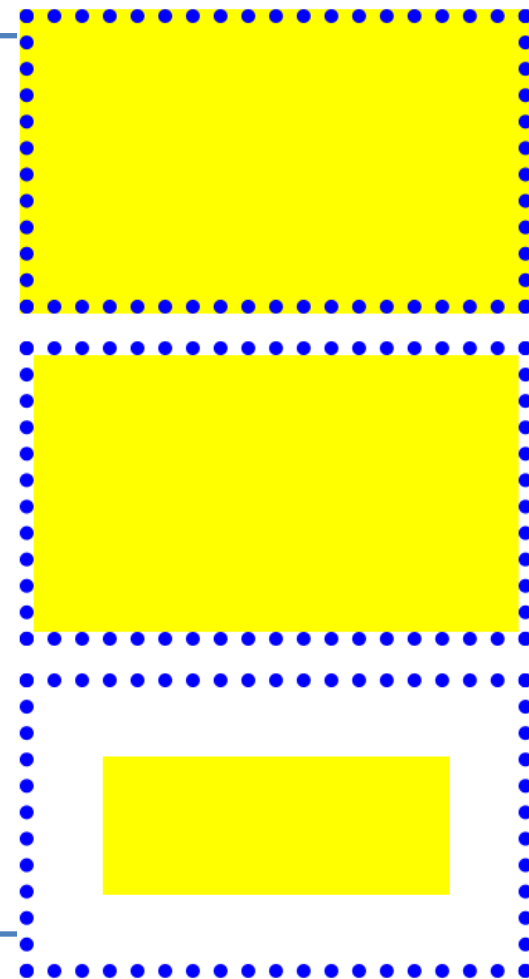
```
div {  
  width: 250px;  
  height: 100px;  
  margin: 20px auto;  
  border: solid 20px;  
  padding: 50px;  
  background: url(../image/1_html5_figure.jpg)  
    no-repeat top left;  
  background-size: cover;  
}  
#div1 {  
  background-origin: border-box;  
}  
#div2 {  
  background-origin: padding-box;  
}  
#div3 {  
  background-origin: content-box;  
}
```



1.7. CSS3 cơ bản: Background

- Multi Background: xử lý nâng cao background

```
div {  
  width: 250px;  
  height: 100px;  
  margin: 20px auto;  
  border: dotted 10px blue;  
  padding: 50px;  
  background: yellow;  
}  
#div1 {  
  background-clip: border-box;  
}  
#div2 {  
  background-clip: padding-box;  
}  
#div3 {  
  background-clip: content-box;  
}
```



1.7. CSS3 cơ bản

- Color: CSS3 hỗ trợ thêm các thuộc tính sau để định dạng màu sắc
 - [RGBA color](#) (Red, Green, Blue, Alpha, với Alpha từ 0 → 1)
 - [HSL color](#) (Hue, Saturation, Lightness)
 - [HSLA color](#) (Hue, Saturation, Lightness, Alpha)
 - [Opacity](#)

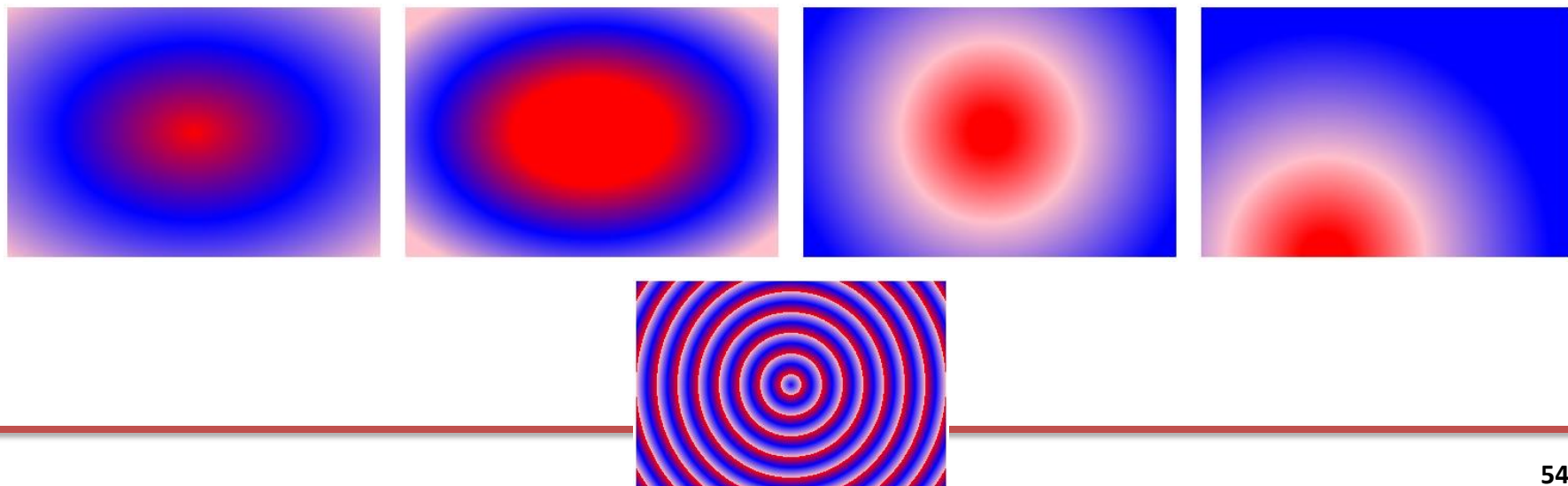
1.7. CSS3 cơ bản

- Gradient Background: kết hợp của hai hay nhiều màu
 - Linear Gradients: Kéo theo các vị trí lên, xuống, trái, phải, đường chéo.
 - background: linear-gradient(direction, color1, color2, color3, color4, ...); (direction nhận một trong các giá trị: to top, to left, to right, to top left, to bottom left...).
 - background: linear-gradient(angle, color1, color2, ...); (với angle là góc xác định giữa đường ngang và đường Gradient từ với giá trị từ 0 → 360deg).
 - background: repeating-linear-gradient(direction, color1 x1%, color2 x2%, ...); (lặp lại gradient).



1.7. CSS3 cơ bản

- Gradient Background: sự kết hợp của hai hay nhiều màu
 - Radial Gradients: Kéo tại vị trí do lập trình viên chọn và lan tỏa ra tứ phía theo một hình nào đó (ví dụ ellipse).
 - [background](#): radial-gradient(shape size at position, start-color, ..., last-color); (trong đó shape mặc định là hình ellipse, size là farthest-corner và position là center).
 - [background](#): repeating-radial-gradient(...); (lặp lại gradient).



1.7. CSS3 cơ bản

- Text-shadow: tạo hiệu ứng shadow dành cho đoạn text
 - text-shadow: h-shadow v-shadow blur-radius color|none|initial|inherit;
 - h-shadow: vị trí bóng ngang so với chữ, số âm sẽ đẩy lên trên và số dương sẽ đẩy xuống dưới.
 - v-shadow: vị trí bóng dọc so với chữ, số âm sẽ đẩy lui phía sau và số dương sẽ đẩy tới phía trước.
 - blur-radius: độ nhòe của chữ bóng, tính bằng pixel.
 - color: màu sắc của bóng.
 - initial: thiết lập giá trị mặc định.
 - inherit: kế thừa giá trị từ thẻ HTML cha.

1.7. CSS3 cơ bản

- Text-shadow: tạo hiệu ứng shadow dành cho đoạn text

```
#heading1 {  
    text-shadow: 0px 0px 20px red;  
}
```

heading1

```
#heading2 {  
    text-shadow: 20px 10px 2px blue;  
}
```

heading2

```
#heading3 {  
    text-shadow: 1px -10px 2px yellow;  
}
```

heading3

```
#heading4 {  
    text-shadow: 5px 5px 5px #500b0b;  
}
```

heading4

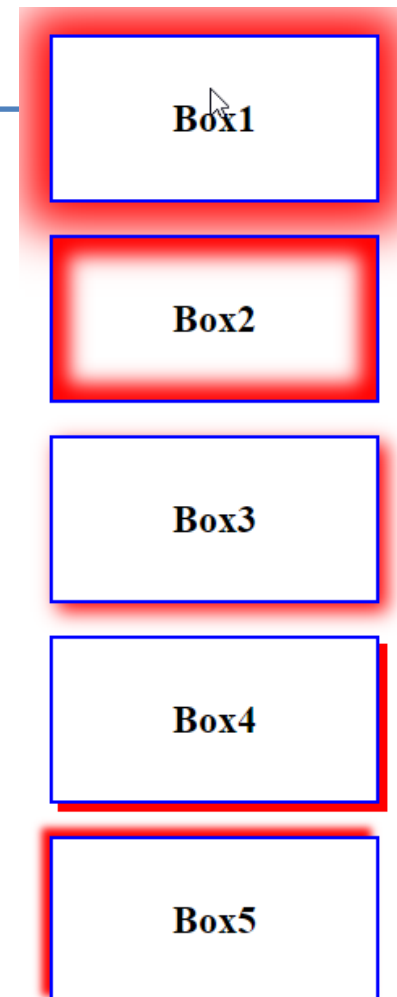
1.7. CSS3 cơ bản

- Box-shadow: tạo hiệu ứng shadow dành cho thẻ HTML (đường viền)
 - box-shadow: h-shadow v-shadow blur spread color |inset|initial|inherit;
 - h-shadow: vị trí bóng ngang so với chữ, số âm sẽ đẩy lên trên và số dương sẽ đẩy xuống dưới.
 - v-shadow: vị trí bóng dọc so với chữ, số âm sẽ đẩy lui phía sau và số dương sẽ đẩy tới phía trước.
 - blur-radius: độ nhòe của chữ bóng, tính bằng pixel.
 - spread: kích thước của bóng tối.
 - color: màu sắc của bóng.
 - inset: thay đổi bóng từ bên ngoài vào trong thay vì từ trong ra ngoài.
 - initial: thiết lập giá trị mặc định.
 - inherit: kế thừa giá trị từ thẻ HTML cha.

1.7. CSS3 cơ bản

- Box-shadow: tạo hiệu ứng shadow dành cho thẻ HTML (đường viền)

```
#box1 {  
    box-shadow: 0px 0px 30px 15px red;  
}  
#box2 {  
    box-shadow: 0px 0px 12px 10px red inset;  
}  
#box3 {  
    box-shadow: 5px 5px 12px 0px red;  
}  
#box4 {  
    box-shadow: 5px 5px 0px 0px red;  
}  
#box5 {  
    box-shadow: -5px -5px 5px 0px red;  
}
```



1.7. CSS3 cơ bản

- Text-overflow: xử lý một đoạn text khi bị tràn ra ngoài thẻ HTML.
 - text-overflow: ellipsis;
- Word-wrap: cho phép đoạn text xuống hàng cho dù chữ đó dài cỡ nào đi nữa.
 - word-wrap: break-word;
- Word-break: break chữ khi xuống hàng
 - word-break: break-all|keep-all;
 - break-all: xuống hàng bất kì lúc nào khi nó đã hiển thị full width
 - keep-all: xuống hàng nếu chữ hiển thị sẽ bị tràn (overflow)

Chào mừng ...

No break-word

ChaoMungCacBanDenDHV

Break-word

ChaoMungCac
BanDenVoiDH
V

Break All

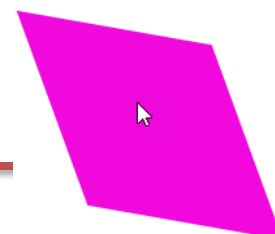
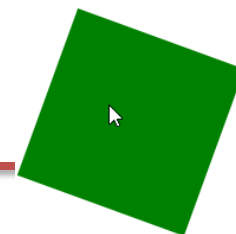
Chào mừng các
bạn đến với
trường DHV

Keep All

Chào mừng các bạ
n đến với trường
DHV

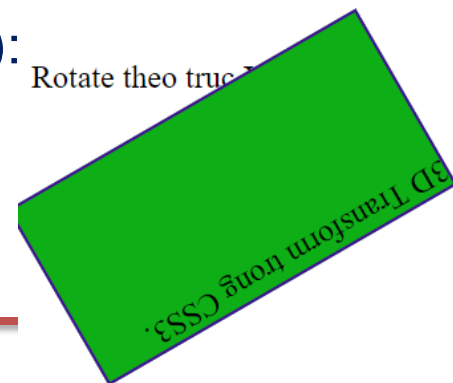
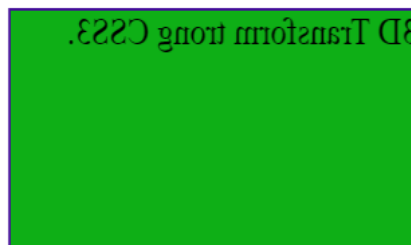
1.7. CSS3 cơ bản

- 2D Transforms: những thuộc tính dùng để xử lý hiệu ứng di chuyển 2D.
 - transform: value; với value nhận một trong các giá trị sau
 - [translate](#)(x, y); translateX(x); translateY(y): di chuyển đối tượng từ vị trí hiện tại (x, y là độ dịch).
 - [rotate](#)(xdeg): xoay đối tượng theo một góc (xdeg là góc xoay).
 - [scale](#)(x, y); scaleX(x); scaleY(y): kéo giãn đối tượng (x là số lần tăng theo chiều rộng, y là số lần tăng theo chiều cao).
 - [skew](#)(xdeg, ydeg); skewX(xdeg); skewY(ydeg): bẻ góc độ của chiều rộng và chiều cao của đối tượng (xdeg là góc độ của hai cạnh hai bên, ydeg là góc độ của hai cạnh trên dưới).
 - [matrix](#)(n, n, n, n, n, n);



1.8. CSS3 nâng cao

- 3D Transforms: những thuộc tính dùng để xử lý hiệu ứng di chuyển 3D.
 - transform: value; với value nhận một trong các giá trị sau
 - translate3d(x, y, z); translateX(x); translateY(y); translateZ(z): tịnh tiến phần tử theo trục x, y, z.
 - rotateX(xdeg); rotateY(ydeg); rotateZ(zdeg): xoay đối tượng theo một góc (xdeg, ydeg, zdeg là góc xoay).
 - scaleX(x); scaleY(y); scaleZ(z): kéo giãn đối tượng (x, y, z là các trục).
 - matrix(n, n, n, n, n, n, n, n, n, n, n, n, n, n, n, n):





1.8. CSS3 nâng cao

- Animation: tạo hiệu ứng chuyển động của cho phần tử (có thể thay thế một phần cho ảnh động, flash, Javascript, jQuery).
 - animation: name duration timing-function delay iteration-count direction play-state;
 - name: tên của animation.
 - duration: thời gian của animation.
 - timing-function: mô tả animation sẽ diễn ra như thế nào trên chu kì của nó, giá trị mặc định ease.
 - delay: sau bao lâu animation sẽ bắt đầu (độ trễ), mặc định 0s.
 - iteration-count: số lần thực hiện animation.
 - direction: chỉ định xem animation sẽ diễn ra lần lượt từng chu kì hay ở chu kì tiếp theo sẽ đảo ngược lại với 2 giá trị: normal và alternate, giá trị này chỉ có tác dụng khi iteration-count có giá trị từ 2 trở lên.
 - play-state: chỉ định animation có diễn ra hay ko (running hoặc paused).

1.8. CSS3 nâng cao

- Animation:
 - Qui tắc @keyframes: qui tắc @keyframes sẽ điều khiển các bước hiệu ứng trung gian
 - @keyframes: từ khóa.
 - animation_name: bắt buộc phải có, tên của animation và có tác dụng khi gọi animation.
 - from, to: trạng thái bắt đầu, trạng thái kết thúc (có thể sử dụng % thay cho from, to)
 - css-style: Cần có 1 hay nhiều thuộc tính css.

```
@keyframes animation_name {  
    from {  
        css-styles; ...;  
    to {  
        css-styles; ...;  
    }  
}
```



1.8. CSS3 nâng cao

- Animation:

```
@keyframes animation {  
  from {  
    background-color: pink;  
  }  
  
  to {  
    background-color: green;  
  }  
}  
  
div {  
  width: 100px;  
  height: 100px;  
  background-color: red;  
  animation-name: animation;  
  animation-duration: 5s;  
  animation-iteration-count: infinite;  
  animation-direction: alternate;  
}
```





1.8. CSS3 nâng cao

- Animation:

```
@keyframes keyframes {  
  from {  
    left: 0;  
    width: 10px;  
    height: 10px;  
    background-color: yellow;  
  }  
  to {  
    left: 300px;  
    height: 100px;  
    width: 300px;  
    background-color: red;  
  }  
}  
div {  
  animation-name: keyframes;  
  animation-duration: 3s;  
  animation-timing-function: ease-in;  
  animation-iteration-count: infinite;  
  animation-direction: alternate-reverse;}
```



1.8. CSS3 nâng cao

- Animation: (các VD khác: [1](#), [2](#))

```
@keyframes animation {  
  from {  
    margin-left: 100%;  
    font-size: 300%;  
    color: red;}  
  75% {  
    font-size: 100%;  
    margin-left: 25%;  
    color: orange;}  
  to {  
    margin-left: 0%;  
    font-size: 400%;  
    color: blue;}  
}  
  
div {  
  animation-name: animation;  
  animation-duration: 5s;  
  animation-iteration-count: infinite;}
```

Đại học Vinh



Đại học Vinh



Đại học Vinh

1.8. CSS3 nâng cao

- Multi Column: sắp xếp văn bản dưới dạng cấu trúc một bài báo.

Giá trị	Miêu tả
column-count	Được sử dụng để đếm số cột mà phần tử nên được phân chia thành
column-fill	Được sử dụng để xác định cách fill dữ liệu vào cột
column-gap	Được sử dụng để xác định khoảng trống giữa các cột
column-rule	Được sử dụng để xác định một số qui tắc
rule-color	Được sử dụng để xác định qui tắc màu cho cột
rule-style	Được sử dụng để xác định qui tắc style cho cột
rule-width	Được sử dụng để xác định độ rộng
column-span	Được sử dụng để xác định span giữa các cột

1.8. CSS3 nâng cao

- Multi Column:

```
.multi {  
  column-count: 4;  
  column-gap: 40px;  
  column-rule-style: solid;  
  column-rule-color: orangered;  
  column-rule-width: thick;  
}
```

Sáng ngày 21/8/2020, CĐGD Việt Nam tổ chức hội nghị tổng kết công tác công đoàn năm học 2019 - 2020, triển khai công tác năm học 2020 - 2021 và sơ kết giữa nhiệm kỳ thực hiện Nghị quyết Đại hội XV nhiệm kỳ 2018 - 2020. Hội nghị được tổ chức bằng hình thức trực tuyến tại điểm cầu của CĐGD Việt Nam tại

Hà Nội và hơn 130 điểm cầu tại Công đoàn Giáo dục 63 tỉnh, thành phố và các đại học quốc gia, đại học vùng, các trường đại học, cao đẳng và đơn vị trực thuộc với tổng số trên 1.300 đại biểu tham dự. Dự và chỉ đạo Hội nghị tại điểm cầu Hà Nội có GS.TS Phùng Xuân Nhạ - Ủy viên BCH Trung ương Đảng, Bí

thư Ban Cán sự Đảng, Bộ trưởng Bộ GD&ĐT và đồng chí Trần Văn Thuật - Phó Chủ tịch Tổng LĐLĐ Việt Nam. Tham dự Hội nghị có đại diện lãnh đạo Cục NG&CBQLGD và Văn phòng Bộ GD&ĐT; Ban Kinh tế Chính sách và thi đua khen thưởng, Ban Tổ chức Tổng LĐLĐ; lãnh đạo của 10 Công

đoàn ngành Trung ương trong khối thi đua; Ban Thường vụ Công đoàn cơ quan Bộ GD&ĐT. Về phía CĐGD Việt Nam có đồng chí Chủ tịch, các Phó Chủ tịch CĐGD Việt Nam và cán bộ công chức cơ quan CĐGD Việt Nam.



1.8. CSS3 nâng cao

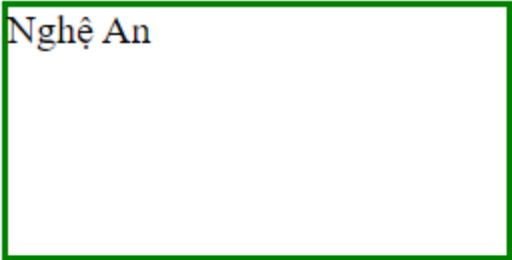
- Thuộc tính box-sizing: thay đổi chiều cao và độ rộng của phần tử (khi đã xác định chiều cao và chiều rộng, thì kích cỡ của box sẽ lớn hơn một chút do phải cộng thêm padding và border vào chiều rộng và cao đã thiết lập).
 - $\text{width} + \text{padding} + \text{border} = \text{chiều rộng thực sự của một phần tử box.}$
 - $\text{height} + \text{padding} + \text{border} = \text{chiều cao thực sự của một phần tử box.}$



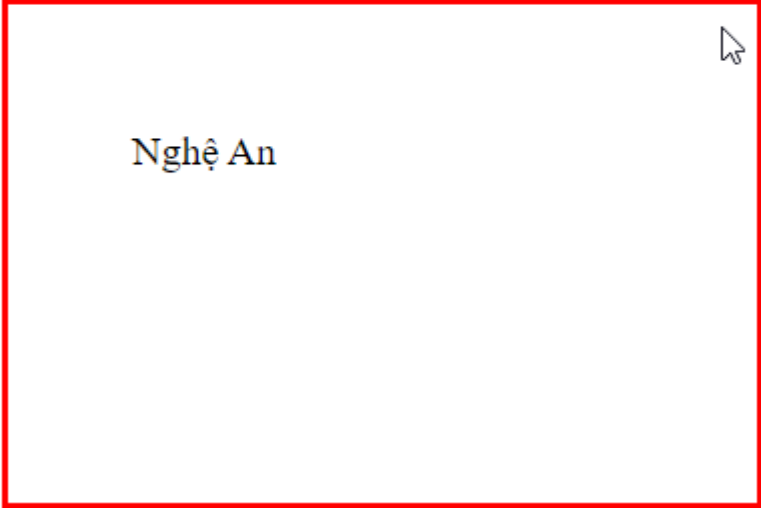
1.8. CSS3 nâng cao

- Thuộc tính box-sizing:

```
.div1 {  
  width: 200px;  
  height: 100px;  
  border: 2px solid green;  
}  
  
.div2 {  
  width: 200px;  
  height: 100px;  
  padding: 50px;  
  border: 2px solid red;  
}
```



Nghệ An



Nghệ An



1.8. CSS3 nâng cao

- Thuộc tính box-sizing:

```
.div1 {  
  width: 300px;  
  height: 100px;  
  border: 2px solid blue;  
  box-sizing: border-box;  
}
```

Đại học Vinh

```
.div2 {  
  width: 300px;  
  height: 100px;  
  padding: 50px;  
  border: 2px solid red;  
  box-sizing: border-box;  
}
```

Đại học Vinh



1.8. CSS3 nâng cao

- Thuộc tính object-fit: định dạng hình ảnh trên web, các giá trị bao gồm:
 - fill: là giá trị mặc định, ảnh sẽ được thu nhỏ lại hoặc kéo dãn cho vừa với kích thước của khung bao bên ngoài.
 - contain: ảnh sẽ được thu gọn hoặc dãn ra để nằm trọn trong khung bao bên ngoài nhưng vẫn giữ được tỷ lệ chiều cao và chiều dài của ảnh.
 - cover: ảnh sẽ được thu gọn hoặc dãn ra để vừa với khung bao bên ngoài, vẫn giữ nguyên tỷ lệ chiều cao và chiều rộng nhưng khung bao sẽ được lấp đầy, lưu ý ảnh có thể bị cắt bớt một số phần.
 - none: ảnh giữ nguyên kích thước gốc, không thay đổi chiều cao và chiều rộng.
 - scale-down: nếu kích thước gốc của ảnh nhỏ hơn chiều rộng và chiều dài của khung thì giá trị này tương đương với giá trị none, nếu ngược lại thì giá trị này tương đương giá trị contain.

1.8. CSS3 nâng cao

- Thuộc tính object-fit:

Origin image



Fill



Cover



Contain



Scale-down



None





1.8. CSS3 nâng cao

- Responsive Web Design:



1.8. CSS3 nâng cao

- Responsive Web Design: chỉ một website có thể hiển thị tương thích trên mọi kích thước hiển thị của trình duyệt (điện thoại, máy tính bảng, máy tính).
 - Viết CSS để cho trình duyệt hiểu và thực thi nó trên các kích thước trình duyệt nhất định.
 - Chỉ sử dụng HTML và CSS, xử lý ở Client-Side.



1.8. CSS3 nâng cao

- Responsive Web Design: các bước áp dụng
 - Bước 1. Khai báo meta viewport: thẻ thiết lập cho trình duyệt hiển thị tương ứng với kích thước màn hình.

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

- Bước 2. Viết CSS cho chiều rộng của thiết bị: sử dụng thẻ truy vấn @media (độ rộng thường tính trên đơn vị pixel)

```
body{  
    background-color: lawngreen;  
}  
@media all and (max-width: 480px){  
    body{  
        background-color: pink;  
    }  
}
```

```
@media only screen and (min-width: 320px) and (max-width: 860px) {...}
```

1.8. CSS3 nâng cao

- Responsive Web Design:
 - Có thể sử dụng stylesheets ngoài trang với các loại màn hình khác nhau:

```
<link rel="stylesheet" media="screen and (min-width: 900px)" href="widescreen.css">  
<link rel="stylesheet" media="screen and (max-width: 600px)" href="smallscreen.css">
```

- Media Type

Value	Description
all	Mặc định, sử dụng cho tất cả các loại thiết bị
print	Sử dụng khi in
screen	Sử dụng cho màn hình desktop, máy tính bảng, điện thoại...

1.8. CSS3 nâng cao

- Responsive Web Design: danh sách các kích thước màn hình thiết bị
 - max-width: 320px (điện thoại di động, hiển thị chiều dọc).
 - max-width: 480px (điện thoại di động, hiển thị chiều ngang).
 - max-width: 600px (máy tính bảng, hiển thị chiều dọc).
 - max-width: 800px (máy tính bảng, hiển thị chiều ngang).
 - max-width: 768px (máy tính bảng loại to, hiển thị chiều dọc).
 - max-width: 1024px (máy tính bảng loại to, hiển thị chiều ngang).
 - min-width: 1025px (từ size này trở lên là danh cho desktop thông thường).

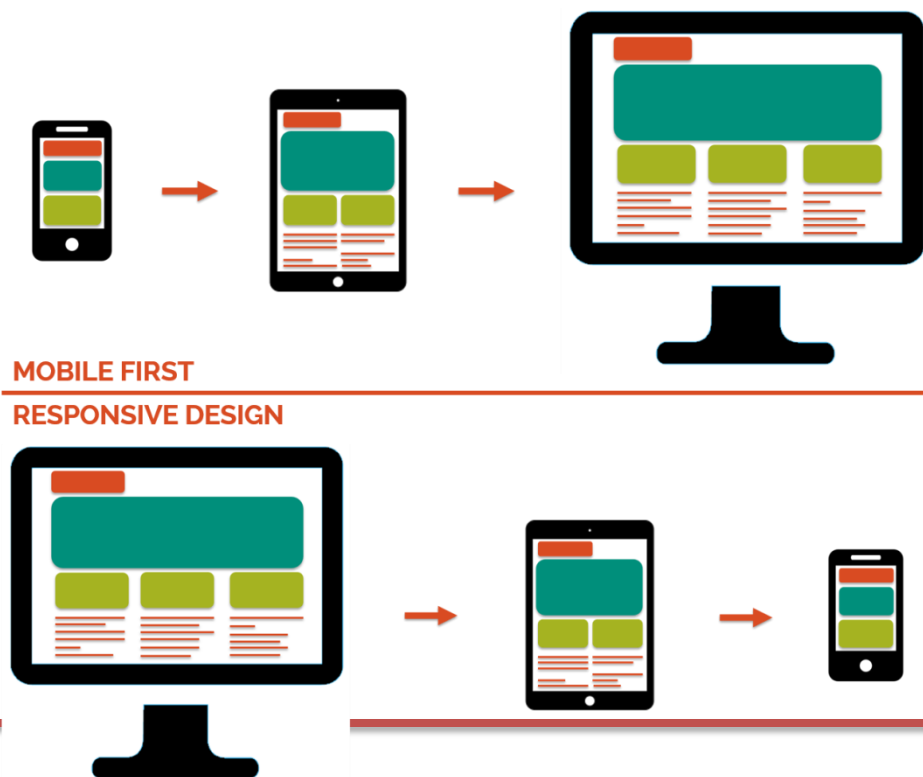


1.8. CSS3 nâng cao

- Responsive Web Design: một số ví dụ khác
 - [Color](#)
 - [Menu](#)
 - [Column](#)
 - [Print](#)
 - [Website](#)

1.8. CSS3 nâng cao

- Mobile-First: tên một quy trình thiết kế website mà sẽ bắt đầu thiết kế cho giao diện ở điện thoại trước và sử dụng điện thoại làm nền tảng ban đầu, sau đó sẽ đến các thiết bị khác như máy tính bảng, desktop...



1.8. CSS3 nâng cao

- Mobile-First: lý do sử dụng
 - Tập trung tối đa vào giao diện ở điện thoại vì xu hướng sử dụng điện thoại ngày càng tăng.
 - Tránh việc viết lại CSS, vì một CSS ở điện thoại có thể được tái sử dụng trên desktop.
 - Dễ dàng trong việc triển khai và quản lý, nâng cấp sau này.
 - Tránh các lỗi hiển thị trên điện thoại do việc tùy biến từ CSS ở desktop.
 - Tối ưu cho SEO (mobile-first indexing)



1.8. CSS3 nâng cao



- Mobile-First: có 2 giải pháp
 - Vẫn ứng dụng Responsive
 - Thiết kế một phiên bản riêng cho Mobile
- Một số tiêu chí cần quan tâm khi thiết kế web Mobile-First:
 - Bố cục nội dung: với màn hình nhỏ trên điện thoại di động, cần quan tâm đến bố cục sao cho dễ tương tác nhất.
 - Tốc độ xử lý: giảm bớt các xử lý tối đa tập trung cho những thứ cần thiết trên thiết bị di động.
 - Tương thích hành vi: tập trung xử lý các tính năng liên quan đến hành vi thao tác trên thiết bị di động.
 - Tốc độ tải web: giảm tối đa lượng yêu cầu (request), kích thước hình ảnh, nội dung ưu tiên cho màn hình nhỏ để tăng tốc độ tải web.





1.8. CSS3 nâng cao

- Mobile-First: Cách thiết kế
 - Sử dụng media features là min-width

```
body {  
    background: #b81010;  
    width: 100%;  
}  
@media all and (min-width: 320px) {  
    body {  
        background: #e7e7e7;  
        width: 50%;  
    }  
}  
@media all and (min-width: 600px) {  
    ...  
}  
@media all and (min-width: 1024px) {  
    ...  
}  
...
```

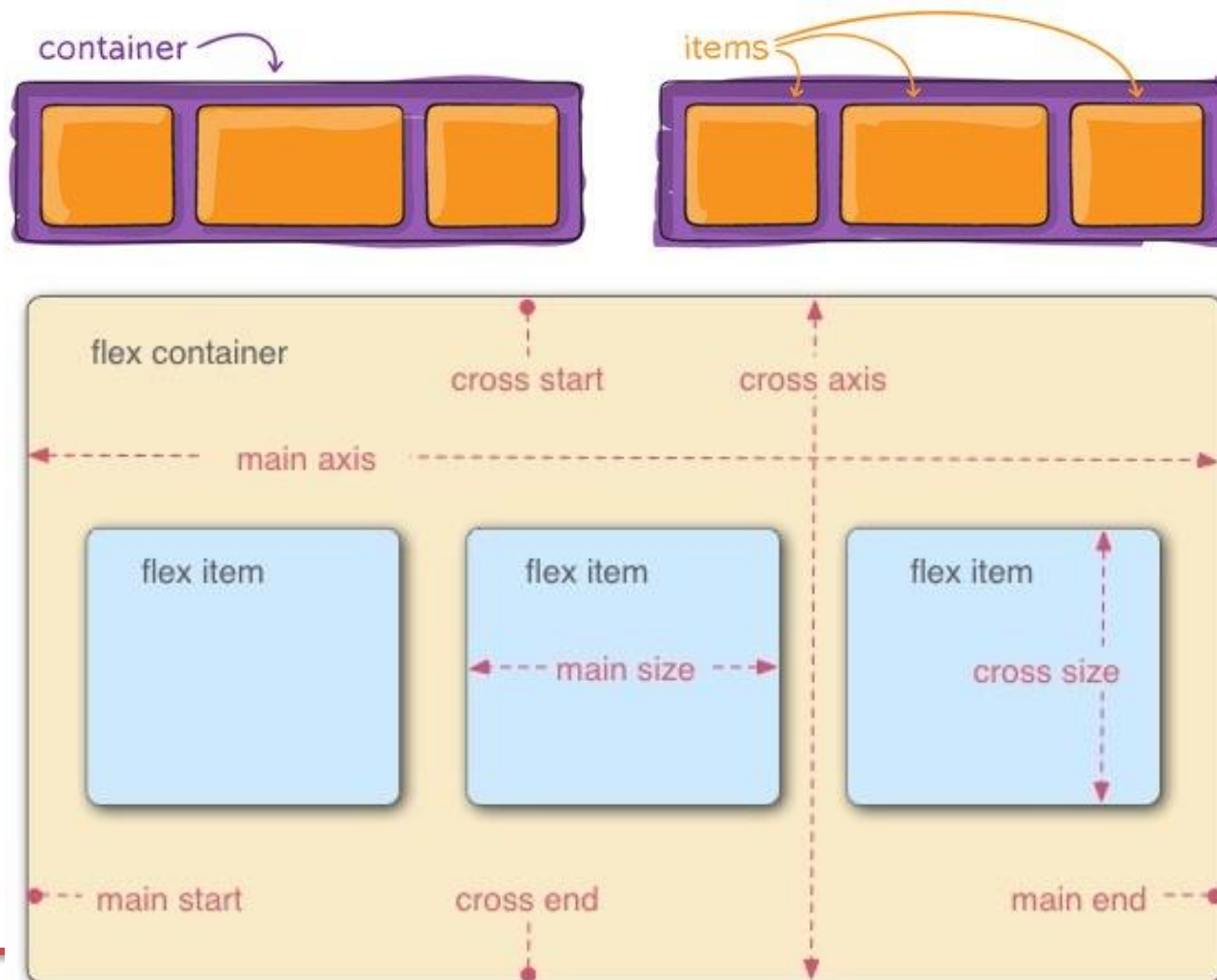


1.8. CSS3 nâng cao

- Flexbox:
 - Một kiểu bố cục trang có khả năng tự cân đối kích thước, thay đổi chiều rộng/chiều cao và thứ tự phần tử bên trong để phù hợp với tất cả các loại thiết bị hiển thị và kích thước màn hình.
 - Chỉ cần thiết lập phần hiển thị theo chiều ngang hay chiều dọc, lúc đó các phần tử bên trong có thể hiển thị theo ý muốn.
 - Phù hợp nhất để thiết lập bố cục ở quy mô nhỏ.

1.8. CSS3 nâng cao

- Flexbox: Bố cục



1.8. CSS3 nâng cao

- Flexbox: Các thành phần
 - container: thành phần lớn bao quanh các phần tử bên trong, các item bên trong sẽ hiển thị dựa trên thiết lập của container này.
 - item: phần tử con của container, có thể thiết lập nó sẽ sử dụng bao nhiêu cột trong một container, hoặc thiết lập thứ tự hiển thị của nó.
 - main axis: trục chính để điều khiển hướng mà các item sẽ hiển thị.
 - main-start, main-end: các item nằm trong container hiển thị từ điểm bắt đầu gọi là main-start tới điểm kết thúc gọi là main-end.
 - main size: kích thước (chiều rộng hoặc chiều cao) của các item, tùy thuộc vào hướng của main axis.
 - cross axis: trục vuông góc của main axis.
 - cross-start, cross-end: vuông góc với main start, main end.
 - cross size: kích thước (chiều rộng hoặc chiều cao) của các item dựa trên trục cross axis, tùy thuộc vào hướng của main axis.

1.8. CSS3 nâng cao

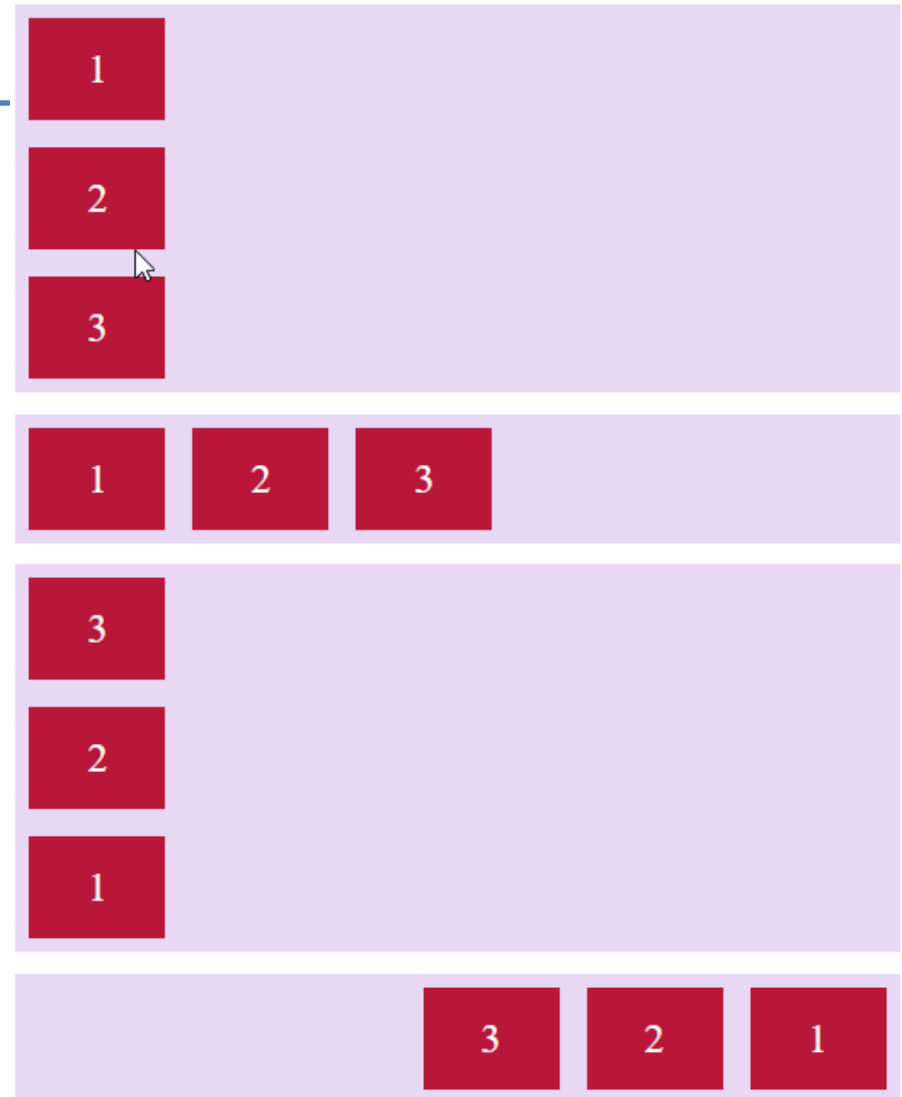
- Flexbox:
 - Khai báo: `display: flex;` (hoặc `display: inline-flex`)
- Các thuộc tính:
 - `flex-direction`: xác định hướng của main-axis để container sắp xếp các item.
 - `row`: mặc định, flex item được sắp xếp theo chiều ngang, từ trái qua phải (main axis nằm ngang).
 - `row-reverse`: flex item được sắp xếp theo chiều ngang, từ phải qua trái (main axis nằm ngang).
 - `column`: flex item được sắp xếp theo chiều dọc, từ trên xuống dưới (main axis đứng dọc).
 - `column-reverse`: flex item được sắp xếp theo chiều dọc, từ dưới lên trên (main axis đứng dọc).



1.8. CSS3 nâng cao

- Flexbox: [flex-direction](#)

```
#flex_column {  
  display: flex;  
  flex-direction: column;  
}  
  
#flex_row {  
  display: flex;  
  flex-direction: row;  
}  
  
#flex_column-reverse {  
  display: flex;  
  flex-direction: column-reverse;  
}  
  
#flex_row-reverse {  
  display: flex;  
  flex-direction: row-reverse;  
}
```





1.8. CSS3 nâng cao

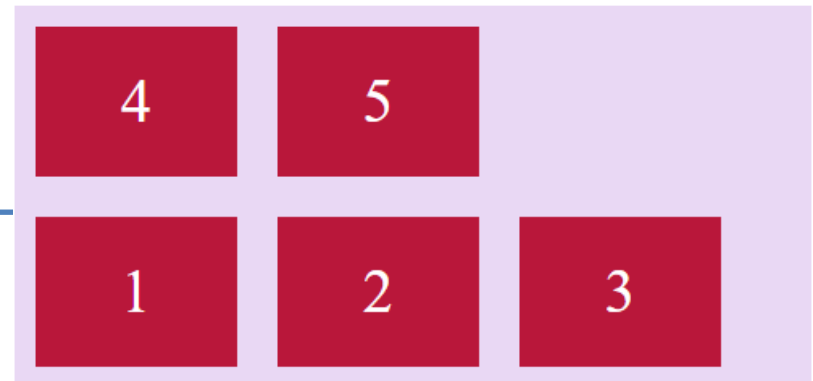
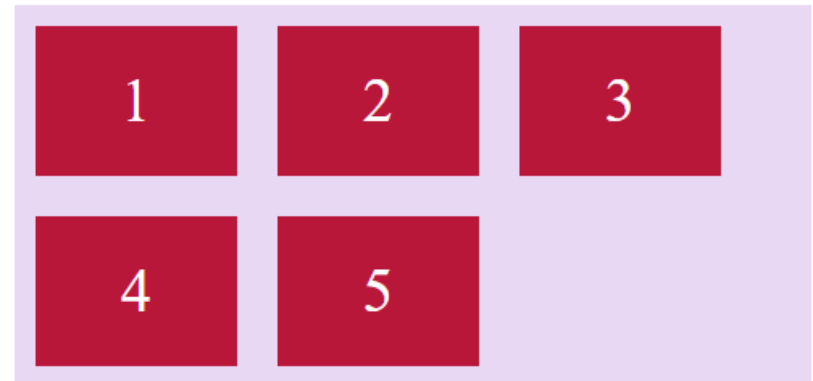
- Flexbox:
 - Thuộc tính flex-wrap: cho phép item tự động xuống dòng khi kích thước container thay đổi.
 - nowrap: mặc định, tất cả các item sẽ nằm trên một dòng.
 - wrap: khi kích thước container thay đổi và tổng chiều rộng các item cộng lại lớn hơn chiều rộng của container thì item sẽ tự động xuống dòng.
 - wrap-reverse: tương tự như wrap, nhưng thay vì xuống dòng thì item sẽ tự động nhảy lên trên.
 - Thuộc tính flex-flow: gộp chung hai thuộc tính flex-direction và flex-wrap.
 - flex-flow: “flex-direction” “flex-wrap”



1.8. CSS3 nâng cao

- Flexbox: flex-wrap

```
#flex1 {  
  display: flex;  
  flex-wrap: nowrap;  
}  
  
#flex2 {  
  display: flex;  
  flex-wrap: wrap;  
}  
  
#flex3 {  
  display: flex;  
  flex-wrap: wrap-reverse;  
}
```





1.8. CSS3 nâng cao

- Flexbox:
 - Thuộc tính justify-content: điều chỉnh vị trí bắt đầu và căn chỉnh các item theo trục main axis.
 - flex-start: giá trị mặc định, item sẽ bắt đầu từ lề chính main-start của container.
 - flex-end: item sẽ bắt đầu từ lề chính main-end của container
 - center: item sẽ nằm giữa container.
 - space-between: các item sẽ có khoảng cách giữa các phần tử bằng nhau do container sẽ tự động căn khoảng cách, item đầu tiên sát lề chứa điểm main-start, item cuối cùng sát lề chứa điểm main-end.
 - space-around: tương tự space-between, nhưng khác ở chỗ là mỗi item có khoảng cách 2 bên cạnh và những khoảng cách này bằng nhau.
 - space-evenly: các item được phân phối sao cho khoảng cách giữa hai item bất kỳ, giữa item và các lề là bằng nhau.

1.8. CSS3 nâng cao

- Flexbox: justify-content

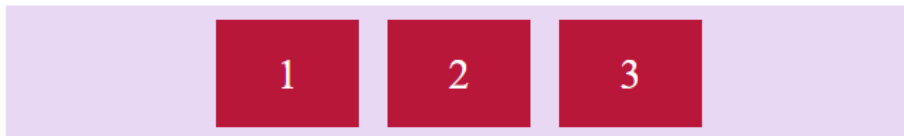
```
#flex1 {  
  justify-content: flex-start;}
```



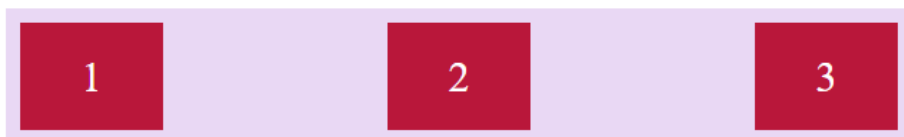
```
#flex2 {  
  justify-content: flex-end;}
```



```
#flex3 {  
  justify-content: center;}
```



```
#flex4 {  
  justify-content: space-between;}
```



```
#flex5 {  
  justify-content: space-evenly;}
```



```
#flex6 {  
  justify-content: space-around;}
```





1.8. CSS3 nâng cao

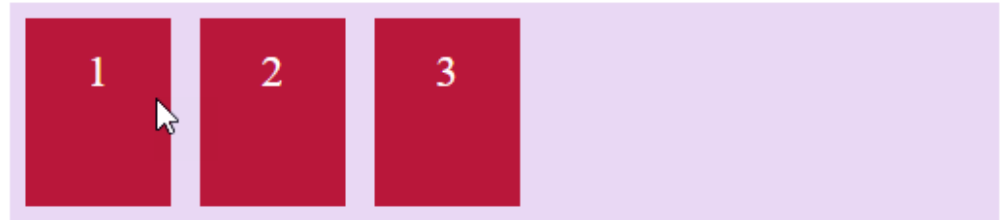
- Flexbox:
 - Thuộc tính align-items: điều chỉnh vị trí bắt đầu và căn chỉnh các item theo trục cross axis.
 - stretch: giá trị mặc định, các phần tử sẽ được kéo dài để lấp đầy container chứa nó, nhưng sẽ ưu tiên giá trị height/width nếu có.
 - flex-start: item sẽ bắt đầu từ lề cross-start của container.
 - flex-end: item sẽ bắt đầu từ lề cross-end của container. Trường hợp mặc định với cross axis đứng dọc, flex-direction: row thì các item sẽ dồn xuống dưới.
 - center: item sẽ căn giữa theo chiều của cross axis.



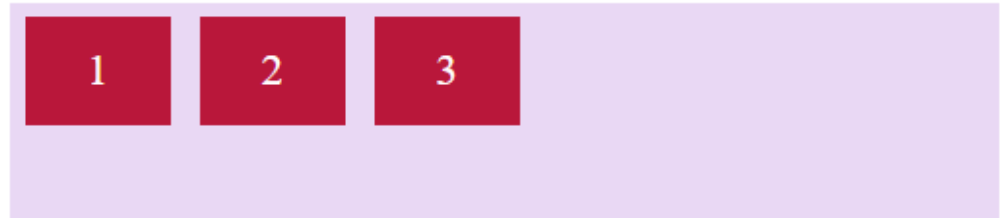
1.8. CSS3 nâng cao

- Flexbox: [align-items](#)

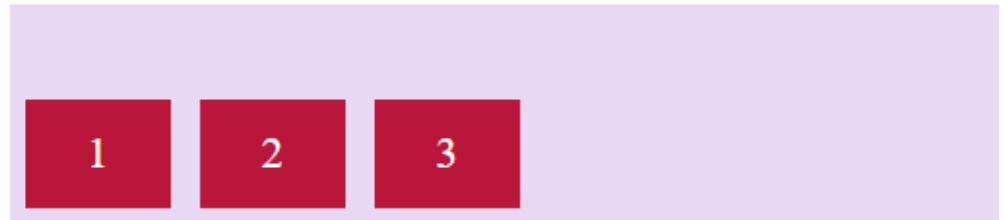
```
#flex1 {  
  display: flex;  
  align-items: stretch;  
}
```



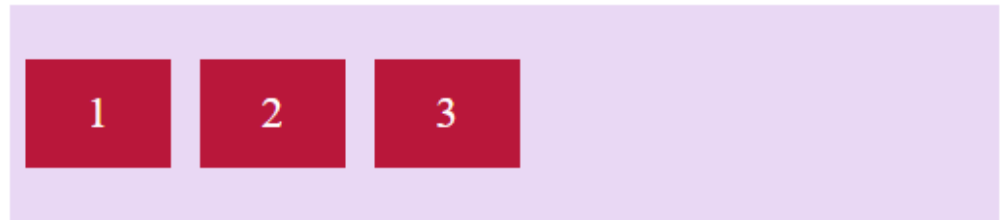
```
#flex2 {  
  display: flex;  
  align-items: flex-start;  
}
```



```
#flex3 {  
  display: flex;  
  align-items: flex-end;  
}
```



```
#flex4 {  
  display: flex;  
  align-items: center;  
}
```





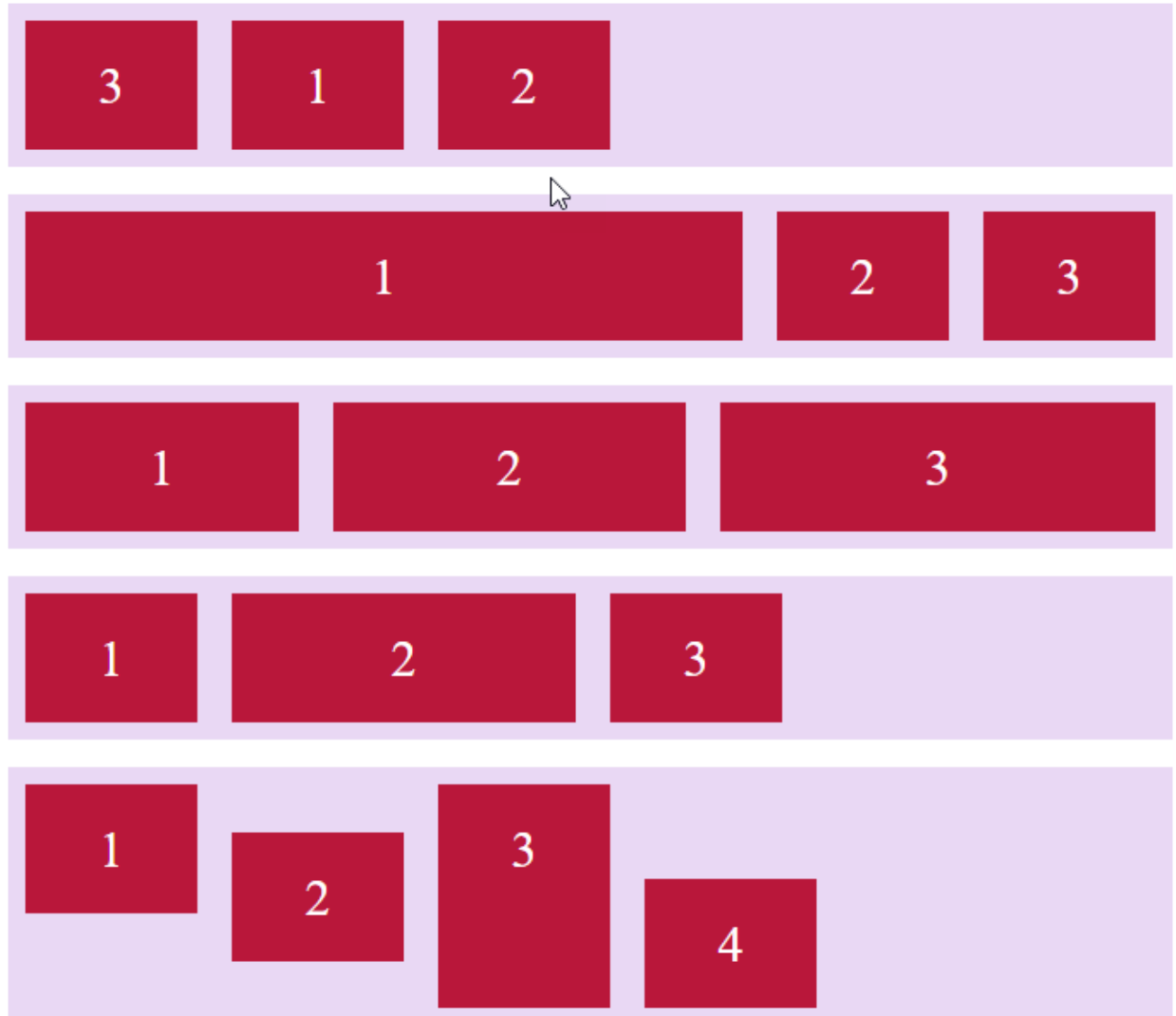
1.8. CSS3 nâng cao

- Flexbox:
 - Order: sắp xếp lại vị trí sắp xếp của các item.
 - `order: <integer>; /* mặc định là 0 */`
 - Flex-grow: giãn các phần tử theo độ rộng của container.
 - `flex-grow: <number>; /* mặc định là 0 */`
 - Flex-shrink: co lại khi thay đổi độ rộng của container.
 - `flex-shrink: <number>; /* mặc định là 1 */`
 - Flex-basis: xác định độ dài ban đầu của một item.
 - `flex-basis: <length> | auto; /* mặc định là auto */`
 - Flex: gộp chung ba thuộc tính flex-grow, flex-shrink và flex-basis.
 - `flex: none | [<'flex-grow'> <'flex-shrink'> <'flex-basis'>]`
 - Align-self: tương tự như align-items của phần container nhưng sử dụng riêng cho từng item.
 - `align-self: auto | flex-start | flex-end | center | baseline | stretch;`



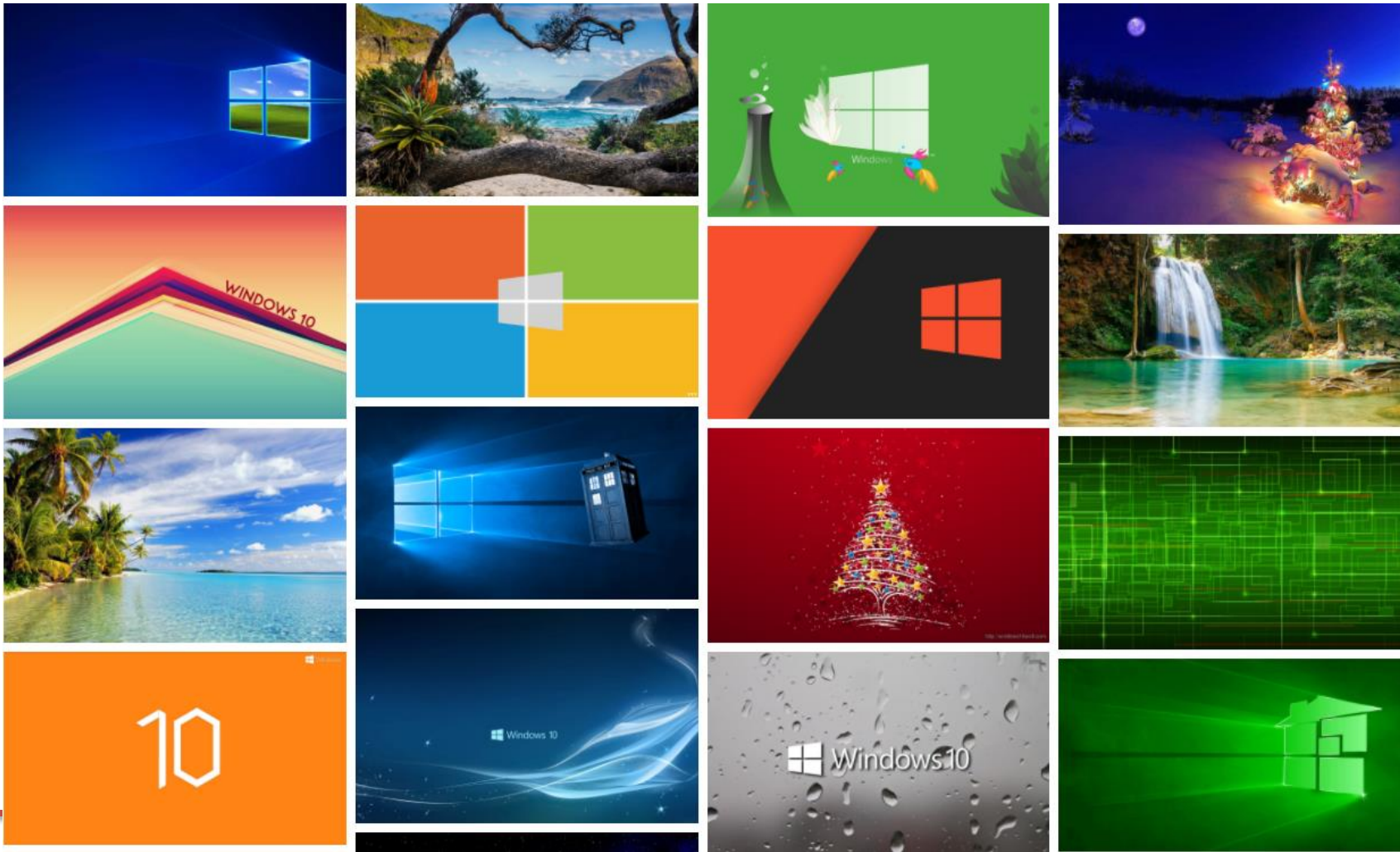
1.8. CSS3 nâng cao

- Flexbox:
 - Order
 - Flex-grow
 - Flex-shrink
 - Flex-basis
 - Flex
 - Align-self



1.8. CSS3 nâng cao

- Flexbox: [Photo Album](#)





1.9. Giới thiệu về Bootstrap

- Là một nền tảng (framework) miễn phí, mã nguồn mở, dựa trên HTML, CSS & Javascript, được tạo ra để xây dựng các giao diện website tương thích với các thiết bị có kích thước màn hình khác nhau.
- Có các thuộc tính về giao diện được quy định sẵn như kích thước, màu sắc, độ cao, độ rộng...
- Bao gồm những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals...
- Là một trong những framework được sử dụng nhiều nhất để tạo ra các responsive website.
- Các ưu điểm:
 - Dễ sử dụng, tùy chỉnh, giảm thiểu viết code
 - Responsive (tiếp cận ưu tiên thiết bị di động đầu tiên)
 - Tương thích cao với các trình duyệt
 - Dễ dàng tích hợp với hầu hết các website open source





1.9. Giới thiệu về Bootstrap

- Các phiên bản:

Phiên bản	Ngày phát hành	Nội dung chính
1.0	19/08/2011	Phiên bản đầu tiên của Bootstrap, chưa hỗ trợ Mobile.
2.0	31/01/2012	Bổ sung hệ thống Grid-Layout 12 cột. Thêm một số thành phần (component) mới. Thay đổi một vài thành phần sẵn có. Vẫn chưa hỗ trợ Mobile.
3.0	19/08/2013	Các thành phần được thiết kế lại theo phong cách thiết kế phẳng (flat design). Và lần đầu tiên hỗ trợ các thiết bị Mobile.
4.0	19/01/2018	Bootstrap 4 gần như viết lại hoàn toàn từ Bootstrap 3, được đánh giá là dễ sử dụng hơn rất nhiều so với phiên bản trước.





1.9. Giới thiệu về Bootstrap

- Cách sử dụng:
 - Tải trực tiếp từ trang cung cấp Bootstrap (<https://getbootstrap.com/>).
 - Sử dụng thông qua CDN Bootstrap.

```
<link rel="stylesheet" href="/chapter_1/bootstrap/css/bootstrap.css">
<script src="/chapter_1/jquery/jquery-3.5.1.js"></script>
<script src="/chapter_1/popper/popper.js"></script>
<script src="/chapter_1/bootstrap/js/bootstrap.js"></script>

<!-- CDN -->
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
```



1.9. Giới thiệu về Bootstrap

- Cách sử dụng:
 - Tải trực tiếp từ trang cung cấp Bootstrap (<https://getbootstrap.com/>).
 - Sử dụng thông qua CDN Bootstrap.

```
<link rel="stylesheet" href="/chapter_1/bootstrap/css/bootstrap.css">
<script src="/chapter_1/jquery/jquery-3.5.1.js"></script>
<script src="/chapter_1/popper/popper.js"></script>
<script src="/chapter_1/bootstrap/js/bootstrap.js"></script>
```

<!-- hoặc thông qua CDN -->

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
```

1.10. Hệ thống lưới trong Bootstrap

- Vùng chứa:
 - Lớp `.container` (fixed-width): cung cấp vùng chứa có độ rộng cố định, được căn giữa màn hình.
 - Lớp `.container-fluid` (full-width): cung cấp độ rộng toàn phần, tùy theo kích thước màn hình.



Trang Bootstrap 4 đầu tiên

Đây là văn bản ví dụ về container

Trang Bootstrap 4 đầu tiên

Đây là văn bản ví dụ về container-fluid

1.10. Hệ thống lưới trong Bootstrap

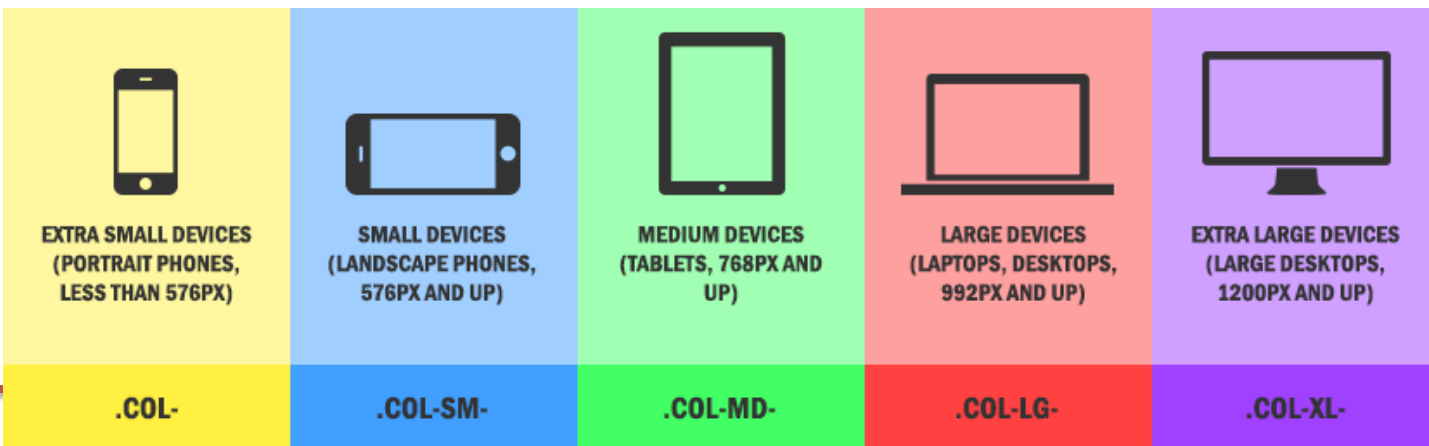
- Bootstrap 4 có hệ thống lưới 12 cột (có thể gom nhiều cột lại thành một cột rộng hơn). Hệ thống lưới có tính đáp ứng, các cột sẽ tự động sắp xếp lại tùy theo kích thước màn hình.
 - Lưu ý: Nếu chỉ có 1 khối và không có số cột thì mặc định là 12 cột. Nếu có 3 khối `<div class="col-md">` `</div>` thì sẽ tự động được chia đều như `<div class="col-md-4">` `</div>`

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

1.10. Hệ thống lưới trong Bootstrap

- Các lớp lưới:

Css Class Prefix	Mô tả
.col-	Dành cho các thiết bị rất nhỏ (Extra Small), có chiều rộng nhỏ hơn 576px
.col-sm-	Dành cho các thiết bị nhỏ (Small), có chiều rộng lớn hơn hoặc bằng 577px
.col-md-	Dành cho các thiết bị trung bình (Medium), có chiều rộng lớn hơn hoặc bằng 768px
.col-lg-	Dành cho các thiết bị lớn (Large), có chiều rộng lớn hơn hoặc bằng 992px
.col-xl-	Dành cho các thiết bị rất lớn (Extra Large), có chiều rộng lớn hơn hoặc bằng 1200px



1.10. Hệ thống lưới trong Bootstrap

- Cấu trúc cơ bản của mạng lưới:
 - .col-*-* có dấu sao đầu tiên (*) là kích thước màn hình; dấu sao thứ hai thể hiện một số từ 1 đến 12.

```
<!-- Điều khiển chiều rộng cột và cách hiển thị trên các thiết bị khác nhau -->
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>
```

```
<!-- Bootstrap tự động định nghĩa kích thước bố cục theo chiều rộng màn hình -->
<div class="row">
  <div class="col"></div>
  <div class="col"></div>
  <div class="col"></div>
</div>
```

1.10. Hệ thống lưới trong Bootstrap

- Ví dụ:

Trang Bootstrap 4 đầu tiên

Đây là văn bản ví dụ về container

Đây là container

Dòng 1

Dòng 2

Dòng 3

Trang Bootstrap 4 đầu tiên

Đây là văn bản ví dụ về container-fluid

col-sm-3

col-sm-9

col-md-2

col-md-3

col-md-7

col-2

col-3

col-7

col-md-2

col-md-10

col-sm-6

col-sm-6

col

col

col

col

col

col-2

col

col

col-2

Long content

Short Content

col-auto

col-auto

col-2

Long content

Short Content

col-auto

col-auto

col-2

1.10. Hệ thống lưới trong Bootstrap

- Ví dụ:

Three Equal Columns



Three Equal Columns



Three Unequal Columns



Three Unequal Columns



More Equal Columns



Row Cols



More Unequal Columns



Equal Height



1.11. Các thành phần trong Bootstrap

- Text:
 - Font chữ có kích thước mặc định là 16px, với chiều cao dòng là 1.5.
 - Kiểu chữ mặc định là “Helvetica Neue”, Helvetica, Arial, Sans-Serif.
 - Tất cả phần tử p đều có margin-top: 0 và có margin-bottom: 1rem (mặc định là 16px).

h1 Bootstrap heading (2.5rem = 40px)

h2 Bootstrap heading (2rem = 32px)

h3 Bootstrap heading (1.75rem = 28px)

h4 Bootstrap heading (1.5rem = 24px)

h5 Bootstrap heading (1.25rem = 20px)

h6 Bootstrap heading (1rem = 16px)

1.11. Các thành phần trong Bootstrap

- Text:
 - Tiêu đề Display: font chữ lớn hơn và độ đậm font nhạt hơn (.display-1, .display-2, .display-3, .display-4).
 - Thẻ small: tiêu đề phụ với font nhạt hơn.
 - Phần tử mark: đánh dấu văn bản với nền nhạt.
 - Phần tử abbr: định nghĩa cụm từ viết tắt với đường gạch chân chấm.
 - Phần tử blockquote: tạo 1 đoạn trích dẫn nội dung từ một nguồn nào đó.
 - Phần tử dl, dt: liệt kê danh sách.

1.11. Các thành phần trong Bootstrap

- Text:
 - .font-weight-bold: in đậm
 - .font-italic: in nghiêng
 - .font-weight-light: in nhạt
 - .font-weight-normal: in bình thường
 - .lead: làm văn bản nổi bật
 - .small: chữ nhỏ (85% bình thường)
 - .text-left, .text-center, .text-right: canh trái, canh giữa, canh phải
 - .text-*-left, .text-*-center, .text-*-right: canh trái, canh giữa, canh phải tùy theo màn hình small, medium, large hay xlarge.
 - .text-monospace: văn bản khoảng trống đơn
 - .text-nowrap: văn bản không bao
 - .text-lowercase, .text-uppercase, .text-capitalize: văn bản thường, văn bản hoa chữ đầu, văn bản in hoa
 - .text-justify: canh đều

1.11. Các thành phần trong Bootstrap

- Color:
 - Màu chữ: `.text-muted`, `.text-primary`, `.text-success`, `.text-info`, `.text-warning`, `.text-danger`, `.text-secondary`, `.text-white`, `.text-dark`, `.text-body`, `.text-light`.
 - Màu chữ theo độ mờ 50%: `.text-black-50`, `.text-white-50`...
 - Màu nền: `.bg-primary`, `.bg-success`, `.bg-info`, `.bg-warning`, `.bg-danger`, `.bg-secondary`, `.bg-dark` và `.bg-light`.



1.11. Các thành phần trong Bootstrap

- Color:

Các màu sắc bối cảnh

Văn bản có nghĩa là câm lặng

Văn bản có nghĩa là quan trọng.

Văn bản có nghĩa là thành công.

Văn bản thể hiện thông tin.

Văn bản thể hiện cảnh báo.

Văn bản thể hiện nguy hiểm.

Văn bản phụ.

Văn bản có màu xám đậm.

Màu văn bản mặc định (thường là màu đen).

Văn bản màu xám trắng (trên nền trắng).

Các màu sắc liên kết bối cảnh

Liên kết câm lặng. Liên kết chính. Liên kết thành công. Liên kết thông tin. Liên kết cảnh báo. Liên kết nguy hiểm. Liên kết phụ. Liên kết xám tối. Liên kết mặc định (đen). Liên kết xám sáng.

Màu sắc văn bản theo độ mờ (opacity)

Văn bản màu đen với 50% opacity trên nền trắng

Văn bản màu trắng với 50% opacity trên nền đen

Màu nền

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

Secondary background color.

Dark grey background color.

Light grey background color.

1.11. Các thành phần trong Bootstrap

- Table:
 - .table: chứa các đường phân chia màu sáng.
 - .table-striped: màu nền nhạt xen kẽ dòng chẵn lẻ.
 - .table-bordered: viền bao quanh bảng.
 - .table-hover: tạo hiệu ứng hover.
 - .table-dark: bảng có màu đen/tối.
 - .table-borderless: bảng không viền.
 - .table-primary, .table-success...: màu sắc bảng.
 - .thead-dark, .thead-light: màu sắc ở các cột tiêu đề bảng.
 - .table-sm: bảng có độ rộng các dòng nhỏ.
 - .table-responsive, .table-responsive-sm,...: bảng đáp ứng (một thanh cuộn ngang được thêm vào bảng nếu kích thước màn hình nhỏ).

1.11. Các thành phần trong Bootstrap

- Table:

Basic Table

Tên	Họ	Email
A	X	test@gmail.com
B	X	test@gmail.com
C	X	test@gmail.com

Basic Table

Tên	Họ	Email
A	X	test@gmail.com
B	X	test@gmail.com
C	X	test@gmail.com

Basic Table

Tên	Họ	Email
A	X	test@gmail.com
B	X	test@gmail.com
C	X	test@gmail.com

Basic Table

Tên	Họ	Email
A	X	test@gmail.com
B	X	test@gmail.com
C	X	test@gmail.com

Basic Table

Tên	Họ	Email
A	Ta	test@gmail.com
A	Ta	test@gmail.com
A	Ta	test@gmail.com
A	Ta	test@gmail.com
A	Ta	test@gmail.com
A	Ta	test@gmail.com
A	Ta	test@gmail.com
A	Ta	test@gmail.com

1.11. Các thành phần trong Bootstrap

- Image:
 - `.rounded`: hình có góc bo tròn.
 - `.rounded-circle`: hình tròn.
 - `.img-thumbnail`: hình thu nhỏ.
 - `.float-right`, `.float-left`: canh chỉnh hình ảnh xuất hiện ở bên trái hoặc bên phải nội dung.
 - `.mx-auto` (`margin:auto`), `.d-block` (`display:block`): căn giữa hình ảnh.
 - `.img-fluid`: hình ảnh đáp ứng.

1.11. Các thành phần trong Bootstrap

- Jumbotron: vùng hộp xám chứa thông tin.

Sử dụng Jumbotron

Bootstrap là 1 framework dùng để thiết kế giao diện Web đáp ứng phổ biến hiện nay.

Đoạn văn bản 1.

Đoạn văn bản 2.

Jumbotron Full-width

Bootstrap là 1 framework dùng để thiết kế giao diện Web đáp ứng phổ biến hiện nay.

1.11. Các thành phần trong Bootstrap

- Alert:
 - Cảnh báo: `.alert`, theo sau là cách lớp `.alert-success`, `.alert-info`, `.alert-warning`, `.alert-danger`, `.alert-primary`, `.alert-secondary`, `.alert-light`, `.alert-dark`.
 - Liên kết cảnh báo: thêm lớp `.alert-link` vào thẻ liên kết.
 - Đóng cảnh báo: thêm lớp `.alert-dismissible` vào phần tử `div` chứa cảnh báo, thêm thuộc tính `class="close"` và `data-dismiss="alert"` vào phần tử nút nhấn (button).
 - Hiệu ứng mờ dần cho cảnh báo: `.fade`, `.show`.

1.11. Các thành phần trong Bootstrap

- Alert:

Các cảnh báo (Alerts)

Thành công! Lớp alert-success.

Thông tin! Lớp alert-info.

Báo động! Lớp alert-warning.

Nguy hiểm! Lớp alert-danger.

Chính! Lớp alert-primary.

Phụ! Lớp alert-secondary.

Màu tối! Lớp alert-dark.

Sáng! Lớp alert-light.

Liên kết cảnh báo

Thành công! Đây là liên kết cảnh báo.

Cảnh báo

Success! This alert box could indicate a successful or positive action.



Thành công! Cảnh báo thành công!



1.11. Các thành phần trong Bootstrap

- Button:
 - Phong cách nút nhấn: sử dụng phần tử `button`, kèm theo là lớp `.btn` và các lớp bối cảnh như `.btn-primary`, `.btn-secondary`...
 - Có thể kết hợp thêm phần tử `a`, `button`, `input` để tạo nút nhấn.
 - Nút chỉ có viền: dùng các lớp `btn-outline-xxx` với `xxx` là tên các nút theo màu sắc.
 - Kích thước nút: thêm một lớp `.btn-lg`, `.btn-sm`... để mô tả kích thước.
 - Nút theo khối: thêm lớp `.btn-block`.
 - Kích hoạt/Vô hiệu hóa nút: dùng lớp `.active` hay `.disabled`.

1.11. Các thành phần trong Bootstrap

- Button: Các phong cách nút nhấn

Cơ bản Chính Phụ Thành công Thông tin Cảnh báo Nguy hiểm Tối Sáng Liên kết

Phần tử Button

Nút liên kết Button Nút Input Nút Submit

Nút dạng viền

Primary Secondary Success Info Warning Danger Dark Light

Kích thước nút nhấn

Lớn Mặc định Nhỏ

Các nút theo khối

Button 1

Button 2

Các nút lớn theo khối

Button 1

Button 2

Các nút nhỏ theo khối

Button 1

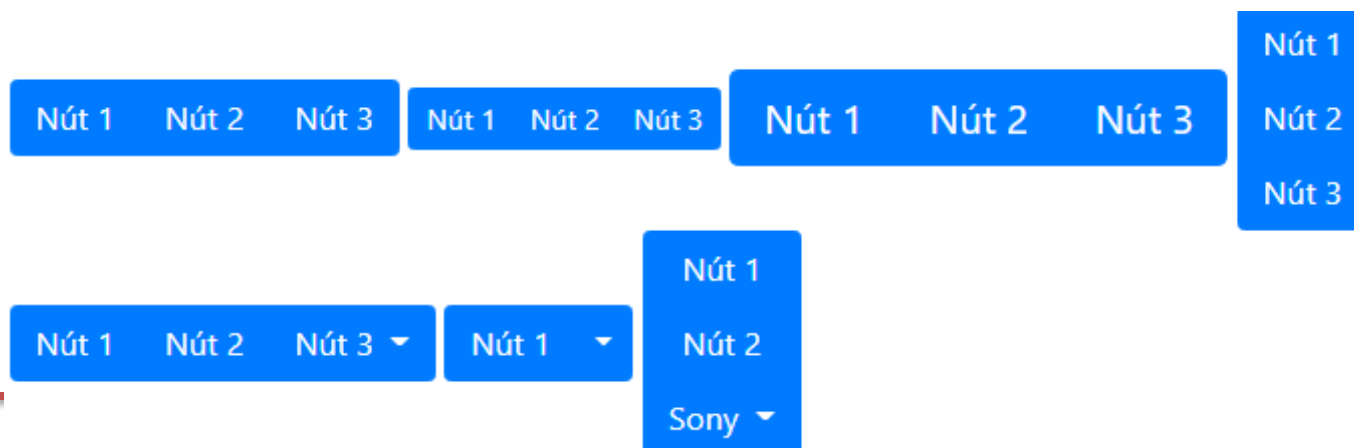
Button 2

Trạng thái nút nhấn

Primary Button Active Primary Disabled Primary Disabled Link

1.11. Các thành phần trong Bootstrap

- Button group:
 - Gom nhóm các button: sử dụng phần tử div và lớp `.btn-group`.
 - Kích thước button group: dùng lớp `.btn-group-lg`, `.btn-group-sm`...
 - Nhóm button theo chiều dọc: sử dụng lớp `.btn-group-vertical`.
 - Nhóm nút con và menu xổ xuống (dropdown): sử dụng lớp `dropdown-toggle` và thuộc tính `data-toggle` có giá trị `dropdown`.
 - Phân chia nút dropdown: dùng lớp `.caret`.
 - Menu dropdown theo chiều dọc: dùng lớp `.btn-group-vertical`.



1.11. Các thành phần trong Bootstrap

- Badge:
 - Huy hiệu, nhãn hiệu: dùng lớp `.badge` và lớp bối cảnh (ví dụ `.badge-secondary`) bên trong các phần tử `span`.
 - Huy hiệu hình con nhộng: thêm lớp `.badge-pill`.
 - Huy hiệu bên trong một phần tử HTML khác: kết hợp với các phần tử HTML để tạo ra nhiều hiệu ứng đồ họa đẹp mắt.

Thông tin chính Thông tin phụ Thao tác nào đó thành công Nguy hiểm Cảnh báo

Thông tin cho nội dung Chữ màu sáng Chữ màu tối

Thông tin chính Thông tin phụ Thao tác nào đó thành công Nguy hiểm

Cảnh báo Thông tin cho nội dung Chữ màu sáng Chữ màu tối

Hộp thư 1

Đã gửi 2

Spam 3

Quan trọng 4

Đã đánh dấu 5

1.11. Các thành phần trong Bootstrap

- Progress bar:
 - Thanh tiến trình: thêm lớp `.progress` làm phần tử chứa và lớp `.progress-bar` để thêm các phần tử con.
 - Chiều cao: thiết lập thuộc tính `height`.
 - Gán nhãn tên: thêm nội dung vào thẻ `div`.
 - Chèn màu: dùng các lớp như `.bg-success`, `.bg-info`, `.bg-warning`...
 - Tạo vạch sọc cho thanh tiến trình: thêm lớp `.progress-bar-striped`.
 - Thanh tiến trình động: thêm lớp `.progress-bar-animated`.
 - Tạo nhiều thanh tiến trình cùng lúc:
 - Hiển thị phần trăm đang thực hiện và chưa thực hiện:
 - Thanh tiến trình kèm theo hiệu ứng động: kết hợp với lệnh Javascript, jQuery.

1.11. Các thành phần trong Bootstrap

- Progress bar:



1.11. Các thành phần trong Bootstrap

- Spinner:
 - Tạo Spinner: sử dụng lớp `.spinner-border`.
 - Tô màu cho Spinner: thêm các lớp màu sắc bối cảnh như `.text-muted`, `.text-primary`, `.text-success`...
 - Spinner Growing (kiểu Spinner chấm tín hiệu hình tròn): sử dụng lớp `.spinner-grow`.
 - Kích thước Spinner: dùng các lớp `.spinner-border-sm` hoặc `.spinner-grow-sm`...
 - Spinner trong nút nhấn: kết hợp với `button`, `textbox` để tạo ra nhiều hiệu ứng đẹp.

1.11. Các thành phần trong Bootstrap

- Spinner:

Tạo Spinner (con quay)

Sử dụng lớp `.spinner-border` để tạo Spinner:



Tô màu cho Spinner



Spinner Growing



Kích thước Spinner



Spinner nút nhấn



1.11. Các thành phần trong Bootstrap

- Spinner:
 - Tạo Spinner: sử dụng lớp `.spinner-border`.
 - Tô màu cho Spinner: thêm các lớp màu sắc bối cảnh như `.text-muted`, `.text-primary`, `.text-success`...
 - Spinner Growing (kiểu Spinner chấm tín hiệu hình tròn): sử dụng lớp `.spinner-grow`.
 - Kích thước Spinner: dùng các lớp `.spinner-border-sm` hoặc `.spinner-grow-sm`...
 - Spinner trong nút nhấn: kết hợp với button, textbox để tạo ra nhiều hiệu ứng đẹp.
 - Tạo hình cho Spinner:
 - Dùng lớp `.border-radius`.
 - Dùng thuộc tính `width`, `height`.
 - Dùng thuộc tính `border-right-color`.

1.11. Các thành phần trong Bootstrap

- Spinner:

Tạo Spinner (con quay)

Sử dụng lớp `.spinner-border` để tạo Spinner:



Tô màu cho Spinner



Spinner Growing



Kích thước Spinner



Spinner nút nhấn



Tạo hình cho Spinner bằng thuộc tính border-radius



Tạo hình cho Spinner bằng thuộc tính width, height



Tạo hình cho Spinner bằng thuộc tính border-right-color và width, height



1.11. Các thành phần trong Bootstrap

- Pagination:
 - Tạo giao diện phân trang: thêm lớp `.pagination` vào phần tử `ul`. Sau đó, cứ mỗi phần tử `li` nằm trong `ul` thêm lớp `.page-item`, đối với liên kết bên trong `li` dùng lớp `.page-link`.
 - Hiện thị trang cần kích hoạt: thêm lớp `.active`.
 - Hủy kích hoạt trang: thêm lớp `.disabled`.
 - Kích thước phân trang: dùng lớp `.pagination-lg`, `.pagination-sm`.
 - Căn chỉnh phân trang: dùng lớp `.justify-content-center`, `.justify-content-end`.
 - Breadcrumb: sử dụng lớp `.breadcrumb`, `.breadcrumb-item`.
 - Nhúng giao diện phân trang vào một bảng có thanh cuộn ngang: dùng lớp `.table-responsive`.

1.11. Các thành phần trong Bootstrap

- Pagination:
 - Tạo giao diện phân trang: thêm lớp `.pagination` vào phần tử `ul`. Sau đó, cứ mỗi phần tử `li` nằm trong `ul` thêm lớp `.page-item`, đối với liên kết bên trong `li` dùng lớp `.page-link`.
 - Hiện thị trang cần kích hoạt: thêm lớp `.active`.
 - Hủy kích hoạt trang: thêm lớp `.disabled`.
 - Kích thước phân trang: dùng lớp `.pagination-lg`, `.pagination-sm`.
 - Căn chỉnh phân trang: dùng lớp `.justify-content-center`, `.justify-content-end`.
 - Breadcrumb: sử dụng lớp `.breadcrumb`, `.breadcrumb-item`.
 - Nhúng giao diện phân trang vào một bảng có thanh cuộn ngang: dùng lớp `.table-responsive`.

1.11. Các thành phần trong Bootstrap

- Pagination:

Phân trang trong Bootstrap 4

Trước	1	2	3	4	5	Sau
-------	---	---	---	---	---	-----

Trước	1	2	3	4	5	Sau
-------	---	---	---	---	---	-----

Trước	1	2	3	4	5	Sau
-------	---	---	---	---	---	-----

Trước	1	2	3	4	5	Sau
-------	---	---	---	---	---	-----

Trước	1	2	3	4	5	Sau
-------	---	---	---	---	---	-----

Trước	1	2	3	4	5	Sau
-------	---	---	---	---	---	-----

Trước	1	2	3	4	5	Sau
-------	---	---	---	---	---	-----

Breadcrumb trong Bootstrap 4

[Trang chủ](#) / [Giới thiệu](#) / [Lịch sử hình thành](#)

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

1.11. Các thành phần trong Bootstrap

- List group:
 - Nhóm danh sách cơ bản: dùng hai phần tử `ul`, `li` và hai lớp `.list-group`, `.list-group-item`.
 - Trạng thái kích hoạt: dùng lớp `.active`.
 - Trạng thái vô hiệu hóa: dùng lớp `.disabled`.
 - Các phần tử liên kết: thêm thuộc tính `href` và lớp `.list-group-item-action`.
 - Loại bỏ viền và góc bo tròn: sử dụng lớp `.list-group-flush`.
 - Loại góc bo tròn: thêm lớp `.rounded-0`.
 - Danh sách theo chiều ngang: thêm lớp `.list-group-horizontal`.
 - Lớp bối cảnh: dùng lớp `.list-group-item-success`, `.list-group-item-secondary`...
 - Nhóm danh sách chứa badge: thêm các lớp huy hiệu (`.badge`, `.badge-pill`,...) thông qua thẻ `span`.
 - Canh giữa các phần tử: dùng `div` ngoài với lớp `.text-center`.

1.11. Các thành phần trong Bootstrap

- List group:

Nhóm danh sách cơ bản

Phần tử thứ 1
Phần tử thứ 2
Phần tử thứ 3

Nhóm danh sách

Phần tử thứ 1
Phần tử thứ 2 được kích hoạt bằng lớp .active
Phần tử thứ 3

Nhóm danh sách

Phần tử thứ 1
Phần tử thứ 2 được vô hiệu hóa bằng lớp .disabled
Phần tử thứ 3

Nhóm danh sách phần tử liên kết

vinhuni.edu.vn
Phần tử thứ 2
Phần tử thứ 3

Nhóm danh sách theo chiều ngang

Phần tử thứ 1	Phần tử thứ 2	Phần tử thứ 3
---------------	---------------	---------------

Các lớp bối cảnh với các phần tử liên kết

Action item
Success item
Secondary item
Info item
Warning item
Danger item
Primary item
Dark item
Light item

Nhóm danh sách với các huy hiệu

Hộp thư	12
Quảng cáo	50
Linh tinh	99



Câu hỏi thảo luận
