

Slides 2.4 - Thiết kế Responsive cho Desktop, Tablet

Mentor: Nguyễn Bá Minh Đạo



Nội dung

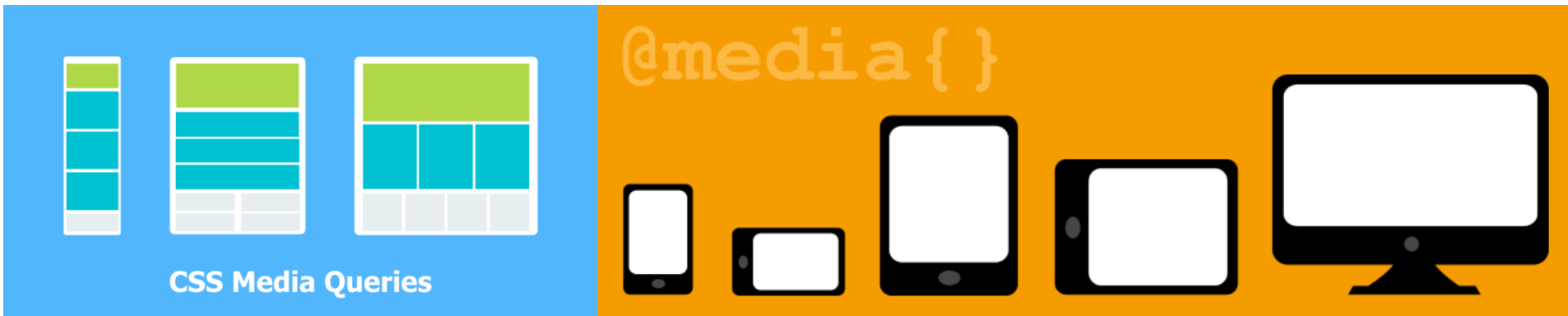
1. Sử dụng truy vấn Media Query
2. Thiết kế định kiểu CSS cho Tablet Viewports
3. Thiết kế định kiểu CSS cho Desktop Viewports
4. Thiết kế các lớp giả sử dụng Pseudo-Classes
5. Thiết kế độ dốc sử dụng Gradients



Sử dụng truy vấn Media Queries

❑ Giới thiệu:

- ♦ Mobile-First **tập trung** vào việc **tạo kiểu**, **nội dung phù hợp cho** các thiết bị **mobile**.
- ♦ Mobile-First **sử dụng bố cục một cột** để **ngăn cuộn ngang** và **giảm** thiểu **cuộn dọc**.
-> **Bố cục này tạo ra trải nghiệm thú vị hơn cho người dùng** thiết bị **mobile**.
- ♦ Thiết kế cho màn hình **tablet** và **desktop** cho phép **bố cục trang web linh hoạt hơn**.
 - Thiết kế cho **tablet** có thể sử dụng **bố cục hai cột**.
 - Thiết kế cho **desktop** có thể sử dụng **bố cục nhiều cột**.
- ♦ Thiết kế cho màn hình **tablet** và **desktop** được đơn giản hóa thông qua việc sử dụng các truy vấn phương tiện **Media Queries**.

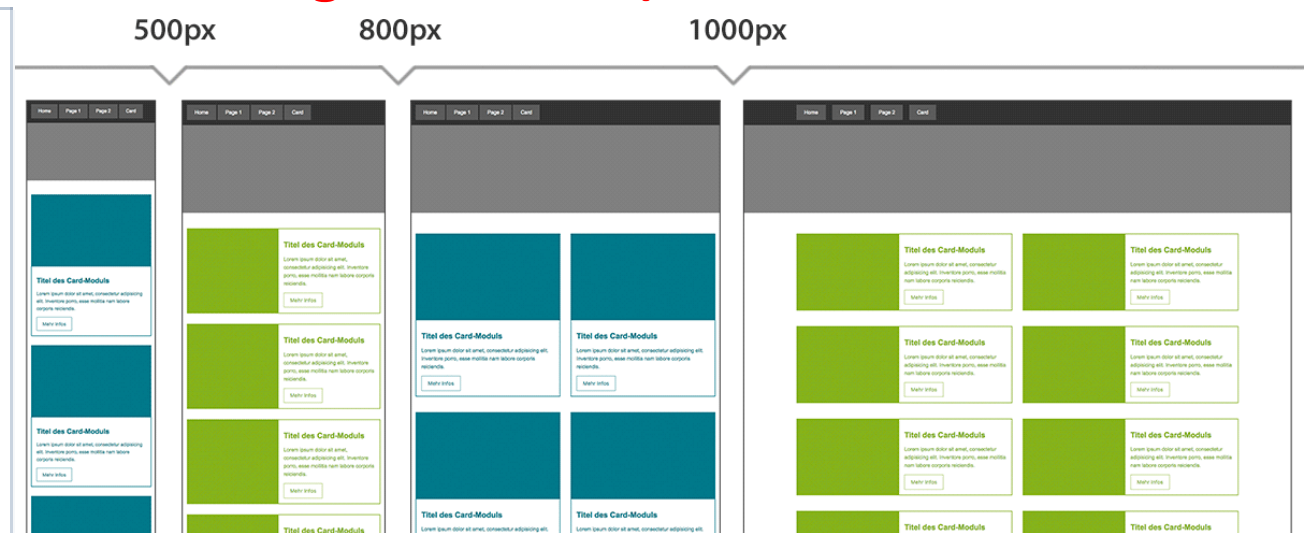
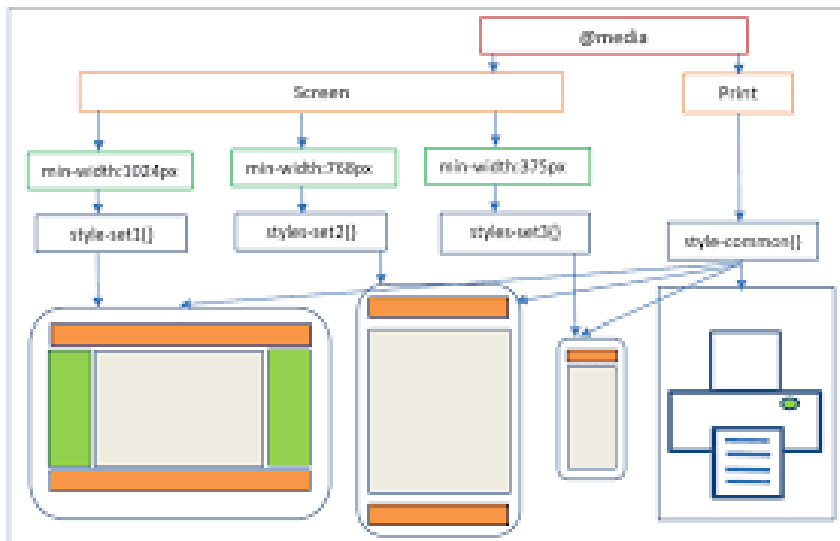




Sử dụng truy vấn Media Queries

❑ Giới thiệu:

- ♦ Quy tắc **@media**, được giới thiệu trong **CSS2**, giúp bạn có thể xác định các quy tắc kiểu **CSS** khác nhau cho các loại phương tiện khác nhau.
- ♦ Không may, những loại phương tiện này **không bao giờ được hỗ trợ hết vì có rất nhiều các loại thiết bị khác nhau**. Ví dụ: Samsung Galaxy S3, S4, S5,...
- ♦ Truy vấn **media queries** trong **CSS3** mở rộng ý tưởng hơn media queries **CSS2**: Thay vì tìm kiếm một loại thiết bị, họ **xem xét khả năng của thiết bị**.

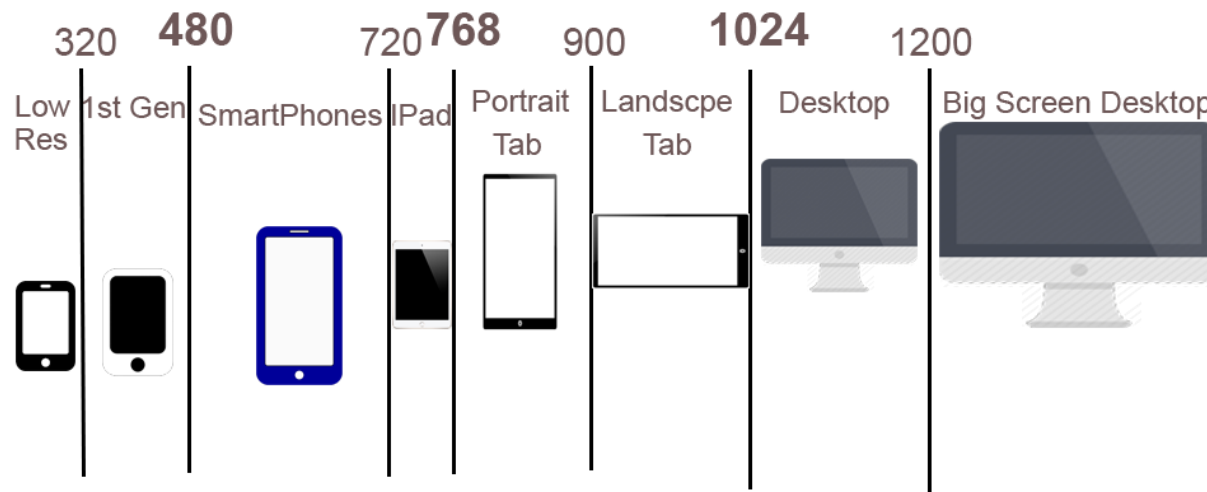




Sử dụng truy vấn Media Queries

□ Giới thiệu:

- ♦ Quy tắc **@media** có thể **áp dụng** các định kiểu **CSS** để **giúp**:
 - **Di chuyển, ẩn** hoặc **hiển thị** nội dung trên **trang web**.
 - **Thay đổi** văn bản hoặc màu sắc hoặc **thêm** bất kỳ định kiểu **CSS** nào **khác** để **giúp** trang dễ đọc hơn trong một tình huống cụ thể.
- ♦ Nếu bạn **thay đổi** kích thước trang web và **thấy hệ thống điều hướng thay đổi** hoặc thấy **bố cục nhiều cột giảm xuống một cột** khi viewport thu hẹp lại bằng kích thước của điện thoại, đó là do các **media queries** đang hoạt động định kiểu lại **viewport** phù hợp.





Sử dụng truy vấn Media Queries

□ Giới thiệu:

♦ Quy tắc **@media** có thể:

▪ Được nhúng trong thẻ **link** kết nối định kiểu **CSS** bên ngoài với tệp HTML.

• Ví dụ 1: `<link rel="stylesheet" href="css/styles.css" media="screen">`

• Ý nghĩa: **styles.css** được tìm thấy trong folder CSS sẽ được áp dụng nếu trang web được hiển thị trên màn hình (screen).

• Ví dụ 2: `<link rel="stylesheet" href="css/stylesprint.css" media="print">`

• Ý nghĩa: **stylesprint.css** được tìm thấy trong folder CSS sẽ được áp dụng nếu trang web được in (print).

▪ Hoặc có thể được chèn vào định kiểu **CSS** bên ngoài (tức file CSS ở ngoài).






♦ Đây là **một kỹ thuật phổ biến** cho phép tạo kiểu CSS với màu sắc phong phú, hình ảnh nhiều màu sắc trên màn hình, nhưng text vẫn màu đen trên khi trang web được in.



Sử dụng truy vấn Media Queries

□ Giới thiệu:

- ♦ **Media Query** có thể được dùng để kiểm tra rất nhiều thứ, ví dụ như:
 - rộng và cao của khung nhìn (**viewport**).
 - rộng và cao của thiết bị (**devices**).
 - chiều hướng (đang là chế độ **landscape** hoặc **portrait** của **tablet/phone**)
 - độ phân giải của màn hình. (thường dùng đơn vị đo là **pixel – px**)
- ♦ Sử dụng media query là **kỹ thuật phổ biến để phân loại desktops, laptops, tablets, và mobile phones** (điện thoại **iPhone** hoặc **Android**)
- ♦ Các phiên bản trình duyệt hỗ trợ quy tắc **@media**:

Thuộc tính					
@media	21.0	9.0	3.5	4.0	9.0



Sử dụng truy vấn Media Queries

❑ Điểm ngắt (breakpoints):

♦ Media queries có thể **xác định loại thiết bị hiện tại thông qua viewport size bằng cách thiết lập** một **điểm ngắt (breakpoints)** cho **trang web**.

♦ Breakpoints là **điểm chuyển giao giữa 2 định kiểu màn hình khác nhau** giúp trang web dễ đọc hơn trên từng loại màn hình khác nhau:

Thiết bị	Độ rộng viewport nhỏ nhất	Độ rộng viewport lớn nhất
Smartphones cỡ nhỏ	320px	480px
Tablets, smartphones cỡ lớn	481px	768px
Tablets (hướng ngang), laptops, màn hình desktop cỡ nhỏ	769px	1279px
Màn hình desktop cỡ lớn	1280px	NA



Sử dụng truy vấn Media Queries

❑ Cú pháp:

- ♦ **Cú pháp** của **Media Query**:

```
@media not|only mediatype and (expressions) {  
    CSS-Code;  
}
```

- ♦ Nếu thông số màn hình đang được chỉ định khớp với các biểu thức **media queries** -> **media queries** trả về giá trị **true** -> **CSS** nằm trong **media queries** này sẽ được áp dụng.
- ♦ Trừ khi bạn sử dụng toán tử **not** hoặc **only**, **mediatype** là tùy bạn chọn. Còn không, **all** sẽ là mặc định và sẽ áp dụng **media queries** chọn tất cả các loại thiết bị khác nhau.





Sử dụng truy vấn Media Queries

❑ Cú pháp:

♦ Bạn cũng có thể có các định kiểu CSS khác nhau cho các phương tiện khác nhau:

`<link rel="stylesheet" media="mediatype and|not|only (expressions)" href="print.css">`

♦ Các loại CSS3 Media thường dùng là:

Giá trị	Mô tả
all	Được sử dụng cho tất cả các loại thiết bị media
print	Dùng cho máy in
screen	Được sử dụng cho màn hình máy tính, máy tính bảng, điện thoại thông minh, v.v...
speech	Được sử dụng cho trình đọc màn hình dùng đọc trang web



Sử dụng truy vấn Media Queries

□ Cú pháp:

♦ Cách 1: Biểu thức truy vấn phương tiện (Media Query Expressions) **có thể được nhúng trực tiếp vào thẻ link** khi thêm đường dẫn file CSS bên ngoài như sau:

▪ Ví dụ 1: `<link rel="stylesheet" href="css/styles-mobile.css" media="screen and (max-width: 480px)">`

➔ Mã này hướng các trình duyệt sử dụng biểu định kiểu **styles-mobile.css** trong thư mục CSS khi **màn hình có chiều rộng khung nhìn <= 480px**.

▪ Ví dụ 2: `<link rel="stylesheet" href="css/styles-tablet.css" media="screen and (min-width: 481px) and (max-width: 768px)">`

➔ Mã này hướng các trình duyệt sử dụng biểu định kiểu **styles-tablet.css** trong thư mục CSS khi **màn hình có chiều rộng khung nhìn >= 481px và <= 768px**.



Sử dụng truy vấn Media Queries

□ Cú pháp:

♦ Cách 2: **Biểu thức truy vấn phương tiện** (Media Query Expressions) **có thể được chèn thông qua** file **CSS bên ngoài** như sau:

▪ VD1: `@media screen {
 body { width: 80%; }
}`

▪ VD3: `@media screen and (min-width: 481px) {
 .mobile { display: none; }
}`

▪ VD2: `@media print {
 body {
 width: 100%;
 color: #000000;
 background-color: #FFFFFF;
 }
}`

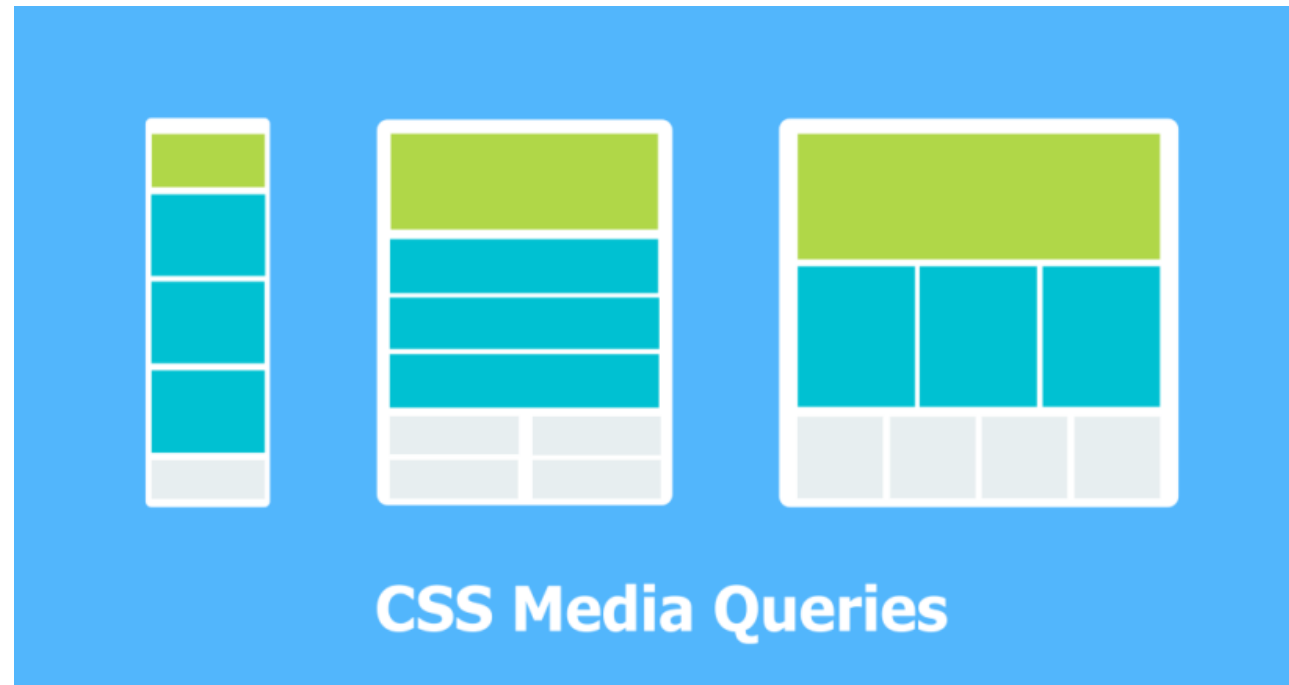
▪ VD4: `@media all and (min-width: 481px)
and (max-width: 768px) {
 body { color: #000000; }
}`



Sử dụng truy vấn Media Queries

❑ Thêm truy vấn Media Queries vào External Style Sheet:

- ♦ Trong dự án **VUS Fitness Club** website của chúng ta bây giờ đã có thêm nhiều định kiểu CSS dùng làm CSS mặc định cho hiển thị mobile với chiến lược **Mobile-First**.
- ♦ Ở bài học này, chúng ta sẽ dùng **Media Queries** để thêm các định kiểu cho một số viewport lớn hơn, quá trình tuần tự từ tablet tới laptop, desktop.





Thiết kế định kiểu CSS cho Tablet Viewports

□ Giới thiệu:

- ♦ Thị trường **tablet** ngày càng **tăng cao**, có rất nhiều các thiết bị **tablet** khác nhau tùy theo hệ điều hành, nhà sản xuất và **kích thước** màn hình được ra đời.
- ♦ Với rất nhiều **kích thước tablet**, có vẻ khó thiết kế bố cục "**một kích thước phù hợp với tất cả**" cho thiết bị **tablet**.
- ♦ Khi bạn sử dụng thiết kế **web responsive** và **media queries**, bạn **không cần** thiết kế nhiều bố cục **tablet** để thích ứng tất cả các loại **tablet**.
- ♦ Mà **chỉ cần** tạo một bố cục để **nhắm mục tiêu** các **viewports** là **tablet**.
- ♦ Nếu một **tablet** cụ thể có viewport nhỏ hơn kích thước tối thiểu được chỉ định trong media queries, bố cục sẽ mặc định là bố cục chế độ xem trên thiết bị **tablet**.

```
147  /* Media query for tablet viewport targets screen size with a minimum */
148  @media only screen and (min-width: 481px) {
149      /* Style specifies a horizontal display for navigation links */
150      nav li {
151          display: inline;
152      }
153  }
```





Thiết kế định kiểu CSS cho Tablet Viewports

❑ Tạo một Media Query cho Tablet Viewport:

- ◆ Tiếp tục nâng cấp dự án **VUS Fitness Club** để **tương thích** với các **tablet viewports** bằng **Media Queries**. Bắt đầu bằng khu vực điều hướng của website:
 - Các liên kết ở khu vực điều hướng hiện tại đang **xuất hiện** dưới **dạng danh sách các nút nằm dọc** -> **thiết kế này phù hợp** với **nhu cầu điều hướng người dùng mobile**.
 - Vì màn hình **tablet** thường lớn hơn màn hình **mobile** nên việc duy trì một danh sách các nút điều hướng theo chiều dọc là **không cần thiết**.
 - Thay vào đó, hãy **tận dụng chiều rộng màn hình** và **căn chỉnh các nút điều hướng theo một đường ngang** để tận dụng tốt các khoảng trống, tạo sự cân bằng cho trang.
- ◆ Vậy phải làm sao để hiện thực căn chỉnh trên?



Thiết kế định kiểu CSS cho Tablet Viewports

❑ Tạo một Media Query cho Tablet Viewport:

- ♦ Để thực hiện thiết kế này, hãy **tạo quy tắc kiểu** để **hiển thị các mục** trong **danh sách điều hướng** dưới **dạng một đường ngang** khi hiển thị trong **tablet viewport**.
- ♦ Thêm các **thuộc tính** và **giá trị** khác ghi đè các **giá trị mặc định** đã được đặt cho chế độ xem trên thiết bị di động.

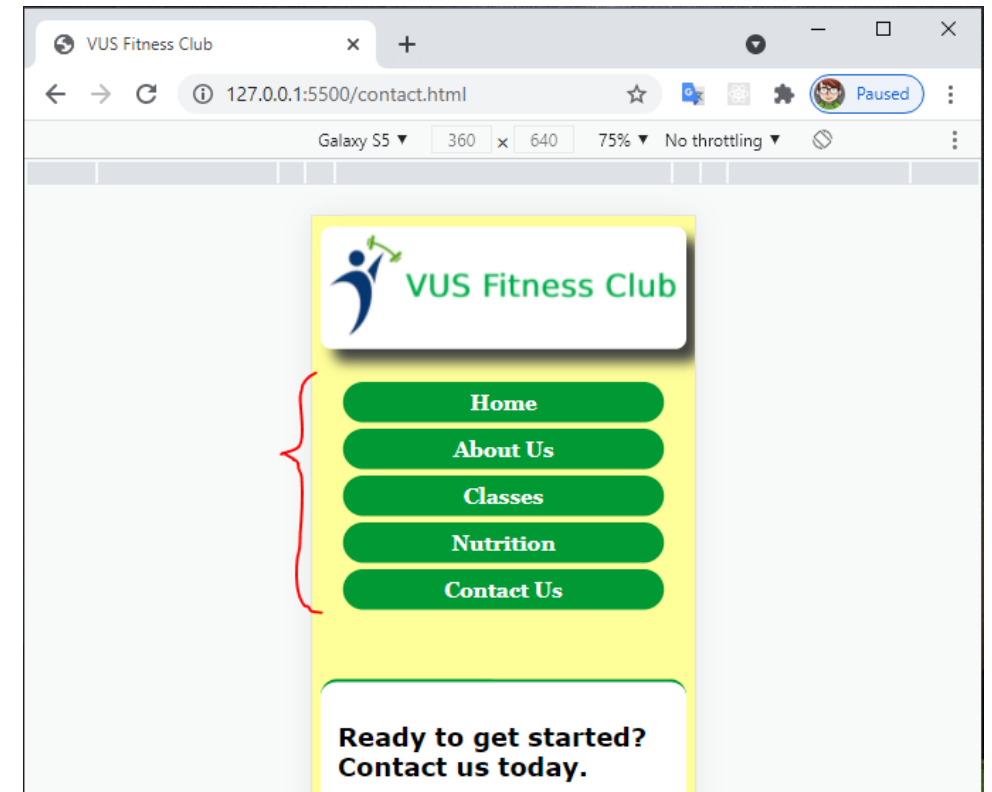
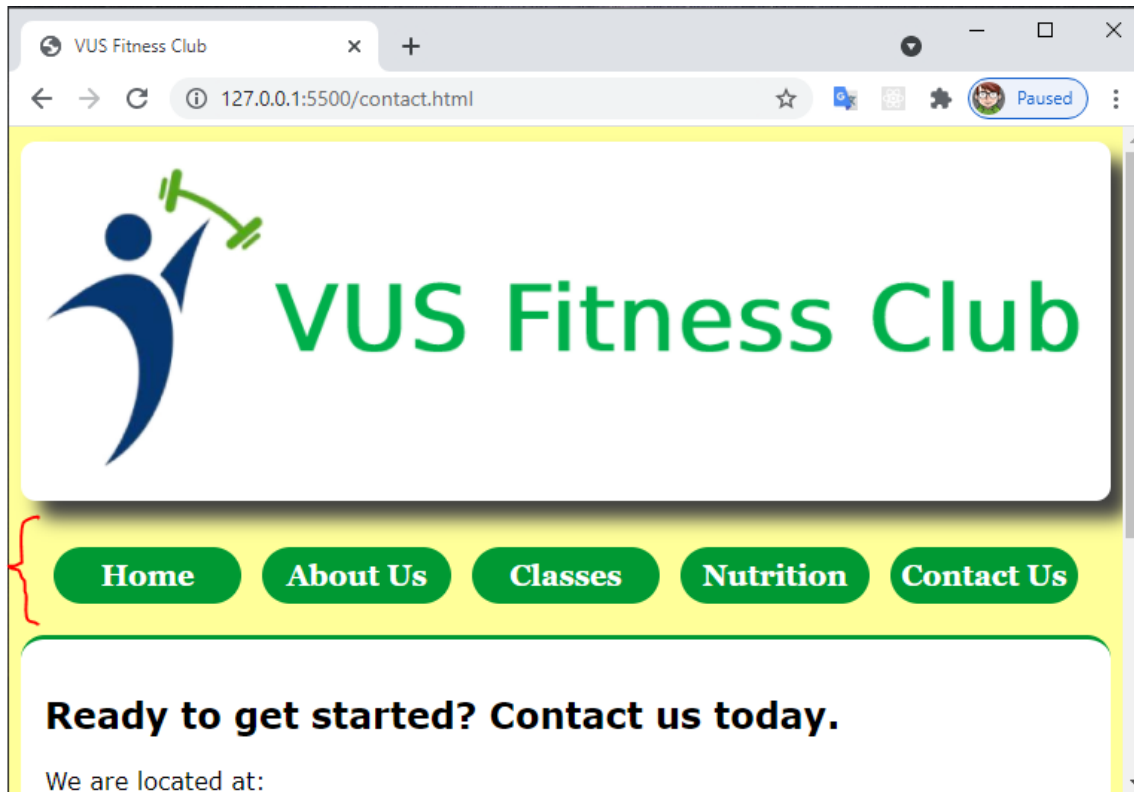
```
/* Media query for tablet viewport targets screen size with a minimum */  
  
@media only screen and (min-width: 481px) {  
    /* Style specifies a horizontal display for navigation links */  
    nav li {  
        display: inline;  
        float: left;  
        margin-left: 1%;  
        margin-right: 1%;  
        padding-left: 0;  
        padding-right: 0;  
        width: 18%;  
    }  
}
```




Thiết kế định kiểu CSS cho Tablet Viewports

❑ Tạo một Media Query cho Tablet Viewport:

♦ Sau khi thêm **Media Queries** vào khu điều hướng của website **VUS Fitness Club**, quan sát lại giao diện khi ở **viewport tablet** và **viewport mobile** sẽ hiển thị khác nhau.





Thiết kế định kiểu CSS cho Tablet Viewports

❑ Phân bổ ẩn/hiện nội dung cho Tablet Viewport:

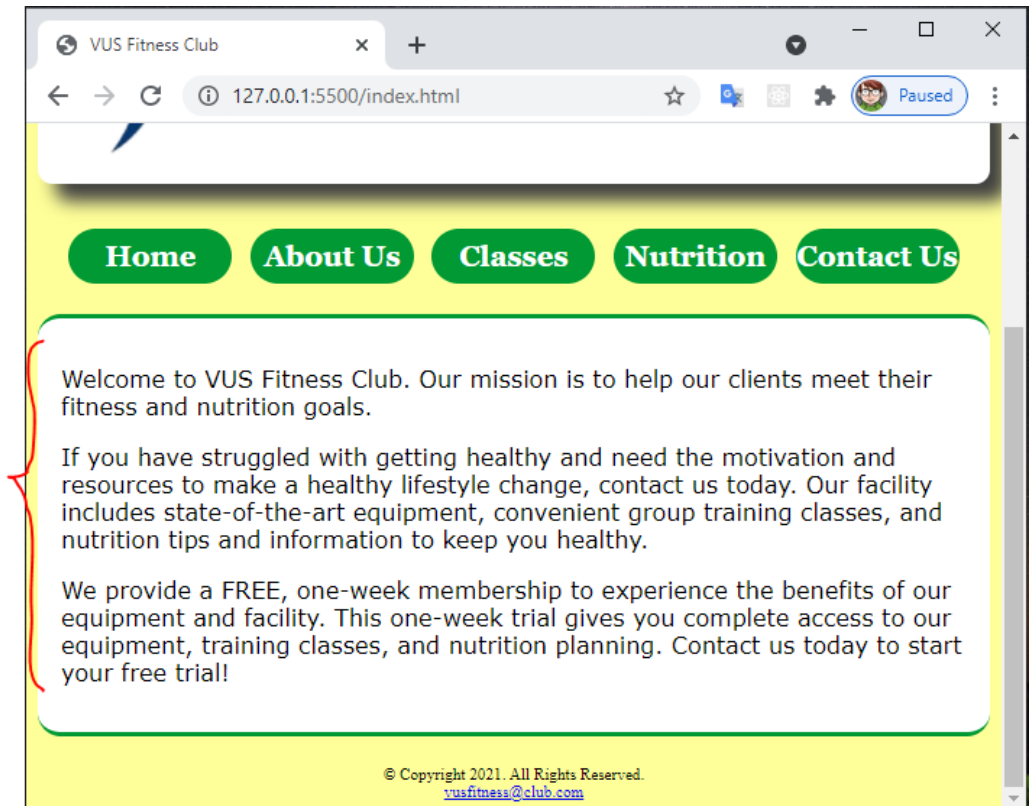
♦ Tiếp theo, bạn phải phân bổ ẩn/hiện nội dung lại phù hợp cho tablet viewport. **Sửa lại màn hình mặc định là màn hình lớn (desktop, tablet) thay vì mobile** như hiện tại.

```
/* Style display the mobile class */

.mobile {
  display: inline; → none
}

/* Style hides the desktop class */

.desktop {
  display: none; → inline
}
```

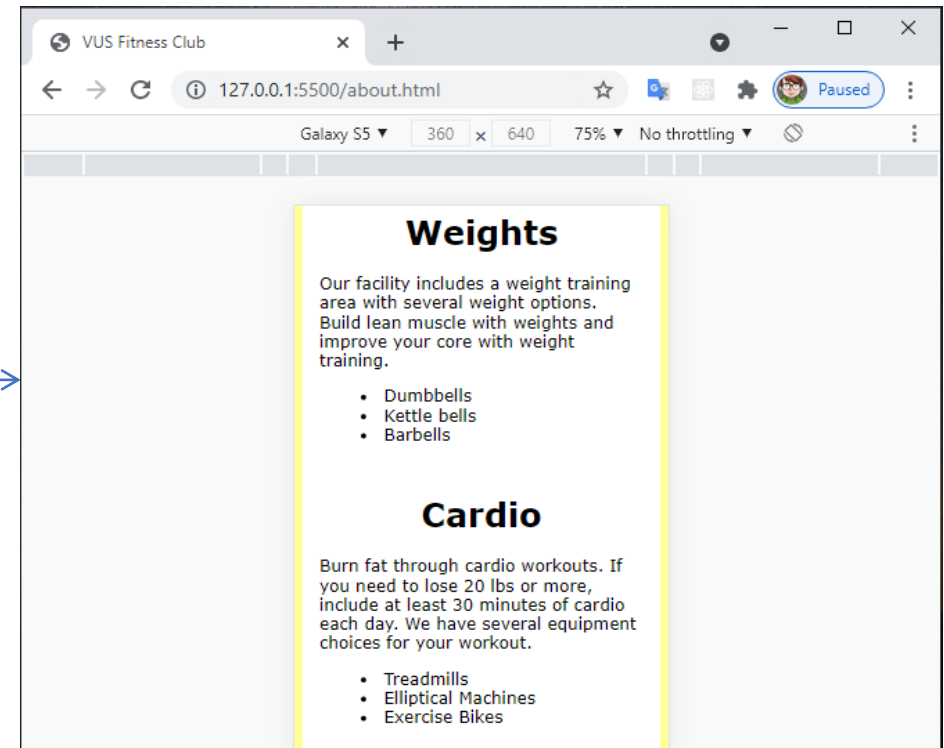
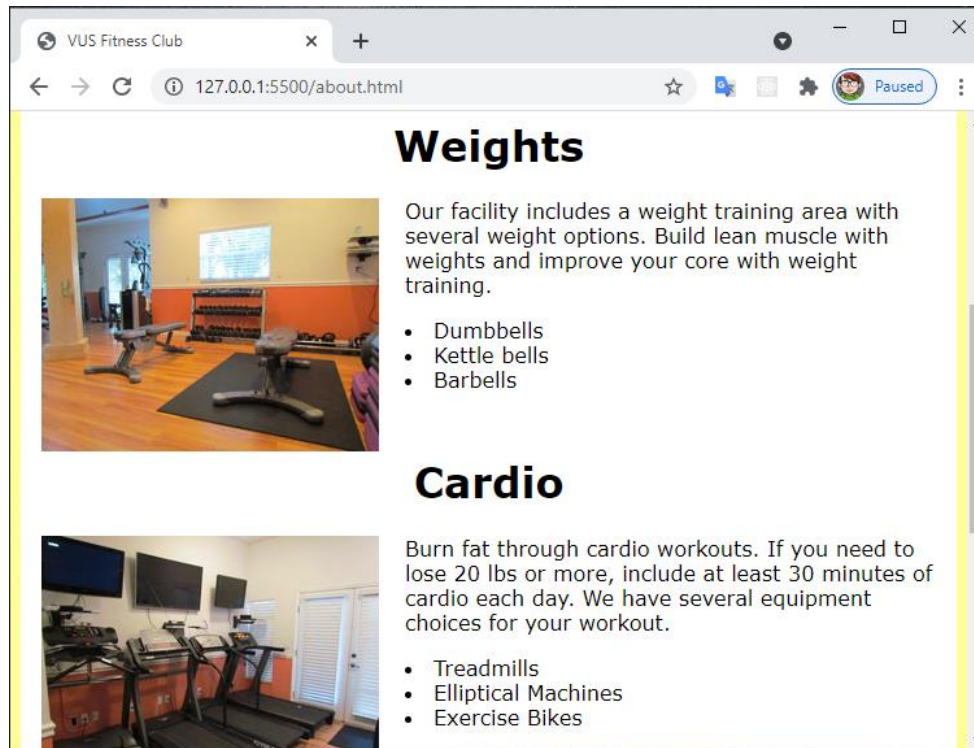




Thiết kế định kiểu CSS cho Tablet Viewports

❑ Phân bổ ẩn/hiện nội dung cho Tablet Viewport:

♦ **Tablet** màn hình lớn hơn **mobile**, **phù hợp** để **hiển thị hình ảnh giống desktop**. Sửa lại định kiểu CSS của lớp **equip** thành **display: inline;** để hiển thị hình ảnh trên **tablet**.





Thiết kế định kiểu CSS cho Tablet Viewports

❑ Phân bố ẩn/hiện nội dung cho Tablet Viewport:

♦ Ở trang **About Us**, các hình ảnh đã hiển thị đúng tablet viewport nhưng trên trình duyệt, tên các dịch vụ hiện có (Weights, Cardio, Personal Training) **hiển thị chưa cân đối**. Sửa CSS lại để nội dung hiển thị hợp lý, cân đối hơn cho trang Web.

```
<h1>Weights</h1>

<p>Our facility includes a weight training area with the following equipment:
<ul class="items">
  <li>Dumbbells</li>
  <li>Kettle bells</li>
  <li>Barbells</li>
</ul>

<h1>Cardio</h1>

<p>Burn fat through cardio workouts. If you need to lose weight, cardio is the best way to go.
<ul class="items">
  <li>Treadmills</li>
  <li>Elliptical Machines</li>
  <li>Exercise Bikes</li>
</ul>
```

```
h1 {
  text-align: center; → left
}
```

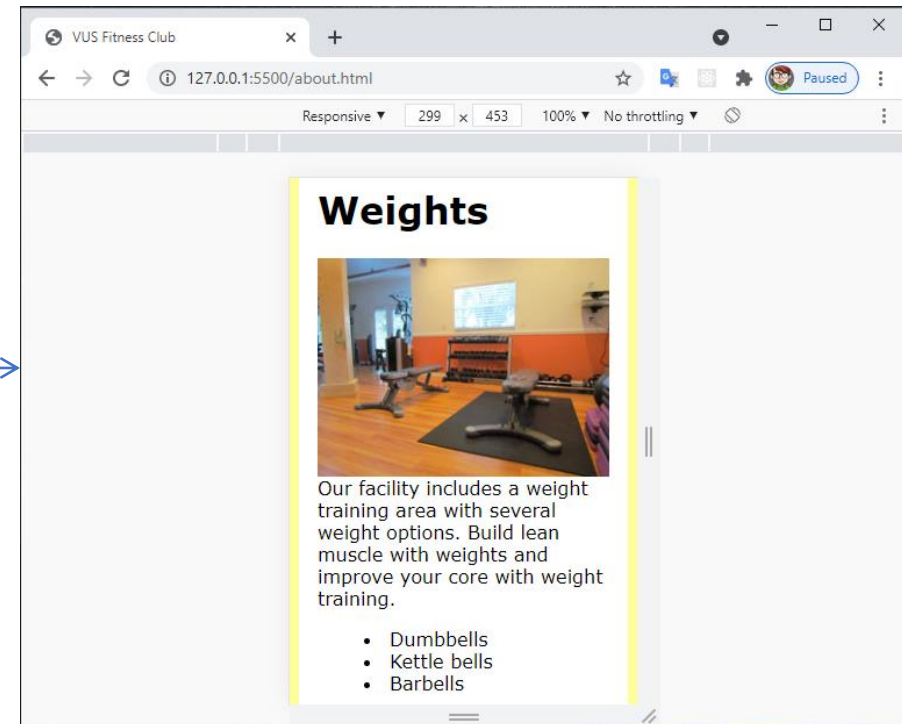
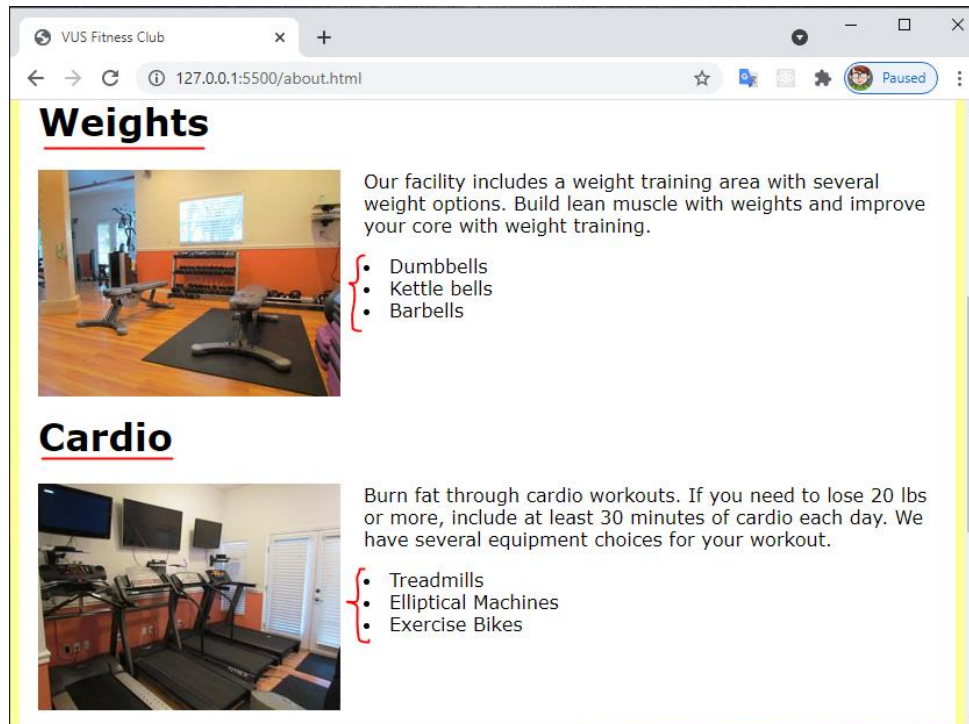
```
.items {
  display: block;
  list-style-position: inside;
  margin-bottom: 5em;
}
```



Thiết kế định kiểu CSS cho Tablet Viewports

❑ Phân bổ ẩn/hiện nội dung cho Tablet Viewport:

♦ Ở trang **About Us**, các hình ảnh đã hiển thị đúng tablet viewport nhưng trên trình duyệt, tên các dịch vụ hiện có (Weights, Cardio, Personal Training) **hiển thị chưa cân đối**. Sửa CSS lại để nội dung hiển thị hợp lý, cân đối hơn cho trang Web.





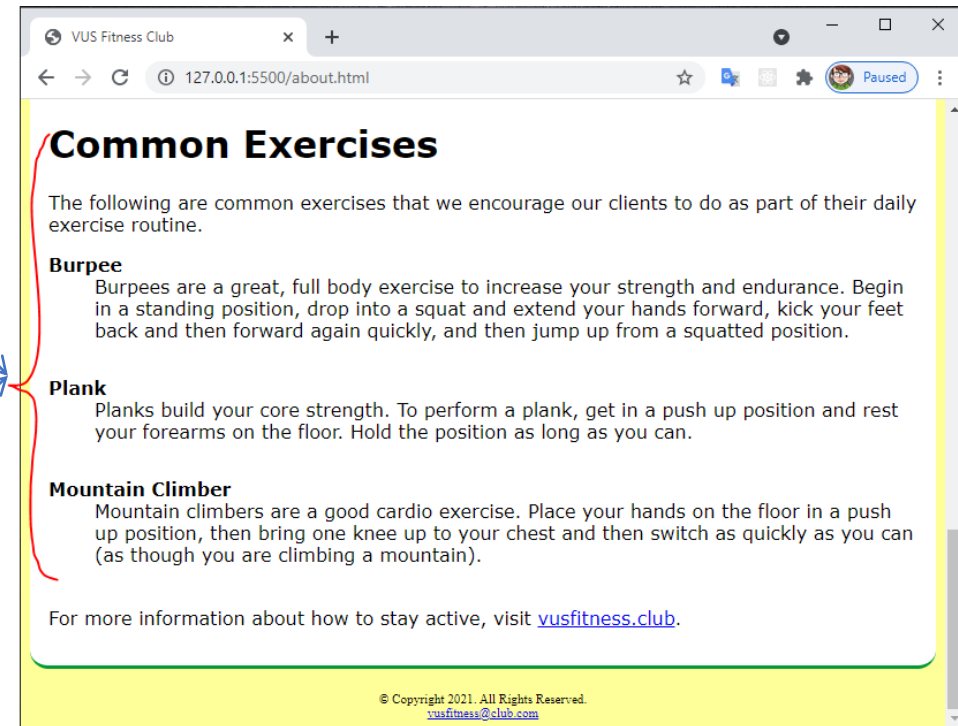
Thiết kế định kiểu CSS cho Tablet Viewports

❑ Hiện thị lớp tablet:

♦ Ở trang **About Us**, một phần tử **div** với **class="tablet"** đang được hiển thị dưới danh sách không có thứ tự cuối cùng. Thẻ này dùng để mô tả **common exercises** và đã bị ẩn đi khi hiện thực **Mobile-First**. Lúc này, ta **sửa lại các CSS** để **hiển thị phù hợp cho tablet**.

```
/* Style hides the tablet class */  
  
.tablet {  
  display: none; → block  
}
```

```
dt {  
  font-weight: bold;  
  margin-top: 0.7em;  
}
```

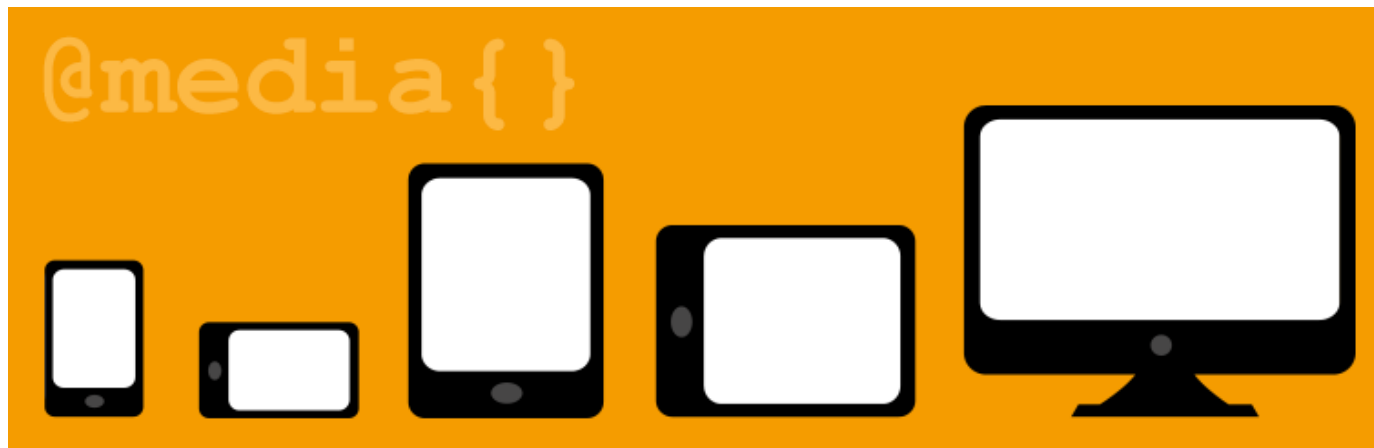




Thiết kế định kiểu CSS cho Desktop Viewports

❑ Giới thiệu:

- ♦ Bây giờ, bạn sẽ **tạo một bố cục desktop** bằng cách **sử dụng thiết kế responsive** và **tạo truy vấn phương tiện** (media queries) cho chế độ xem trên màn hình desktop.
- ♦ **Các nguyên tắc thiết kế desktop** tương tự áp dụng cho thiết kế responsive:
 - Sử dụng điều hướng đơn giản, trực quan
 - Hình ảnh, kiểu chữ rõ ràng và áp dụng cùng một bảng màu.
- ♦ Vì **màn hình desktop** thường **rộng hơn** so với màn hình cho các **thiết bị khác**, **thiết kế** cho **viewport desktop** cũng **tạo cơ hội cho bố cục nhiều cột**.





Thiết kế định kiểu CSS cho Desktop Viewports

❑ Tạo Media Query cho Desktop Viewport:

♦ Tiếp tục tạo media query cho desktop viewport để để bàn sẽ chứa các định kiểu CSS để thay đổi giao diện trang web cho người dùng máy tính để bàn (desktop).

Canh độ rộng
cho **container**
khi gấp desktop
viewport

```
@media only screen and (min-width: 769px) {  
    /* Style specifies a width for container */  
    #container {  
        width: 80%;  
    }  
    /* Style specifies margin and padding for the unordered list  
    within the nav */  
    nav ul {  
        margin: 0;  
        padding-left: 0.50%;  
        padding-right: 0.50%;  
    }  
    /* Style for navigation list items specifies a border radius,  
    border, margin, width, and padding */  
    nav li { ...  
    }  
    /* Style for navigation links specifies display and padding */  
    nav li a { ...  
    }  
}
```

Thêm CSS chỉnh
navigation khi gấp
desktop viewport



Thiết kế định kiểu CSS cho Desktop Viewports

❑ Tạo Media Query cho Desktop Viewport:

♦ Tiếp tục tạo media query cho desktop viewport để để bàn sẽ chứa các định kiểu CSS để thay đổi giao diện trang web cho người dùng máy tính để bàn (desktop).

Chỉ CSS cho
links trên menu
khi gặp desktop
viewport

```
@media only screen and (min-width: 769px) {  
    /* Style specifies a width for container */  
    > #container { ...  
    }  
    /* Style specifies margin and padding for the unordered list  
    within the nav */  
    > nav ul { ...  
    }  
    /* Style for navigation list items specifies a border radius,  
    border, margin, width, and padding */  
    nav li {  
        border-radius: 0;  
        border-top: solid 1px #000000;  
        border-bottom: solid 1px #000000;  
        margin-left: 0;  
        margin-right: 0;  
        padding: 0;  
        width: 20%;  
    }  
    /* Style for navigation links specifies display and padding */  
    nav li a {  
        display: inline-block;  
        padding: 0.7em;  
    }  
}
```

Thêm CSS chỉnh
spacing của
navigation khi gặp
desktop viewport



Thiết kế định kiểu CSS cho Desktop Viewports

❑ Tạo Media Query cho Desktop Viewport:

♦ Quan sát giao diện website VUS Fitness Club sau khi chỉnh sửa **CSS Media Queries** cho khu vực điều hướng, thay đổi kích cỡ từ **viewport tablet** sang **viewport desktop**.





Thiết kế định kiểu CSS cho Desktop Viewports

❑ Chỉnh sửa CSS phần nội dung cho Desktop Viewport:

♦ Loại bỏ border, thêm hiệu ứng box-shadow, và áp dụng top, bottom margins để tạo định kiểu CSS mới ở phần nội dung chính cho desktop viewports.

```
/* Style specifies border, box shadow, and margins for the main element */
```

```
main {  
  display: block;  
  font-family: Verdana, Arial, sans-serif;  
  font-size: 1em;  
  margin-top: 2.5em;  
  padding: 1em;  
  border-radius: 1em;  
  background-color: #FFFFFF;  
  border-top: solid 0.2em #009933;  
  border-bottom: solid 0.2em #009933;  
  clear: left;  
  border-style: none;  
  box-shadow: 0.5em 0.5em 0.5em #404040;  
  margin-bottom: 1em;  
  margin-top: 1em;  
}
```

```
/* Style specifies left and right padding for  
paragraph elements within the main element */
```

```
main p {  
  padding-left: 0.5em;  
  padding-right: 0.5em;  
}
```

```
/* Style for the nav specifies text properties */
```

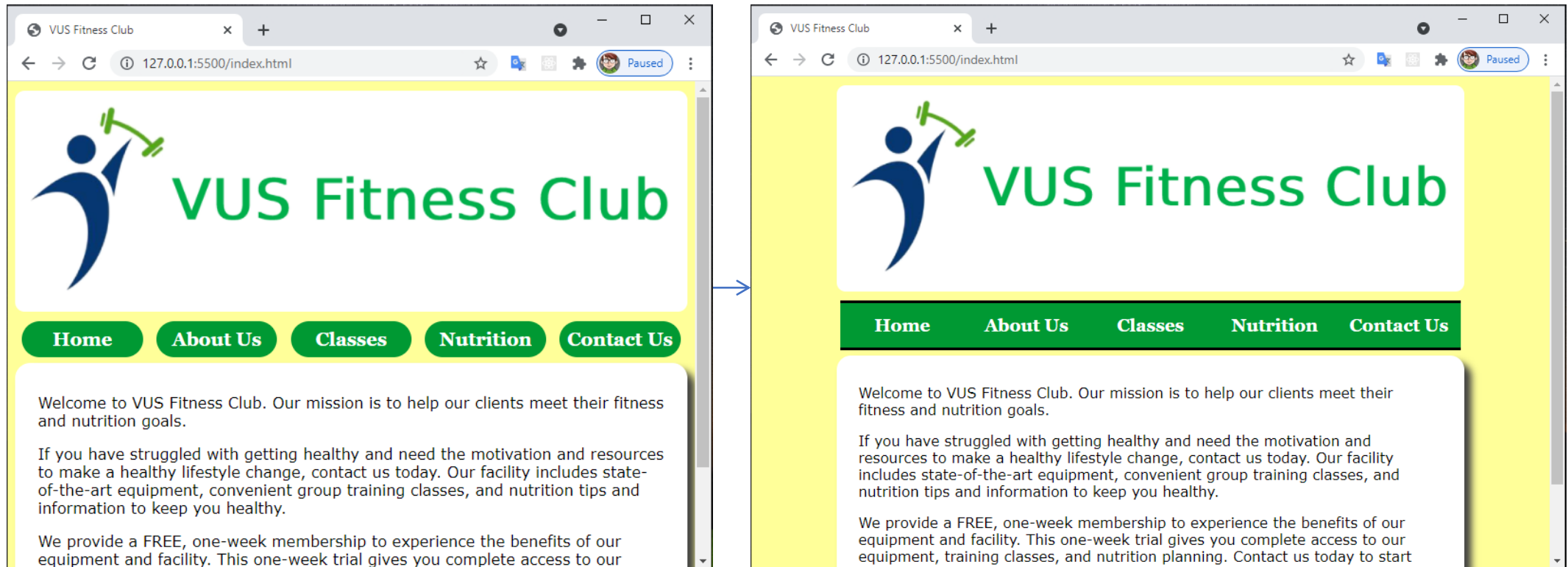
```
nav {  
  font-family: Georgia, 'Times New Roman', Times, serif;  
  font-size: 1.25em;  
  font-weight: bold;  
  text-align: center;  
  margin-top: -0.8em;  
}
```



Thiết kế định kiểu CSS cho Desktop Viewports

❑ Tạo Media Query cho Desktop Viewport:

♦ Quan sát giao diện website VUS Fitness Club sau khi chỉnh sửa **CSS Media Queries** cho phần nội dung chính, thay đổi kích cỡ từ **viewport tablet** sang **viewport desktop**.





Thiết kế định kiểu CSS cho Desktop Viewports

❑ Chỉnh sửa CSS phần nội dung cho Desktop Viewport:

- ♦ Ở trang **About Us**, tiếp tục **Loại bỏ border**, áp dụng **top, bottom margins** cho các danh sách thông qua lớp **items** để định kiểu phù hợp cho **desktop viewports**.

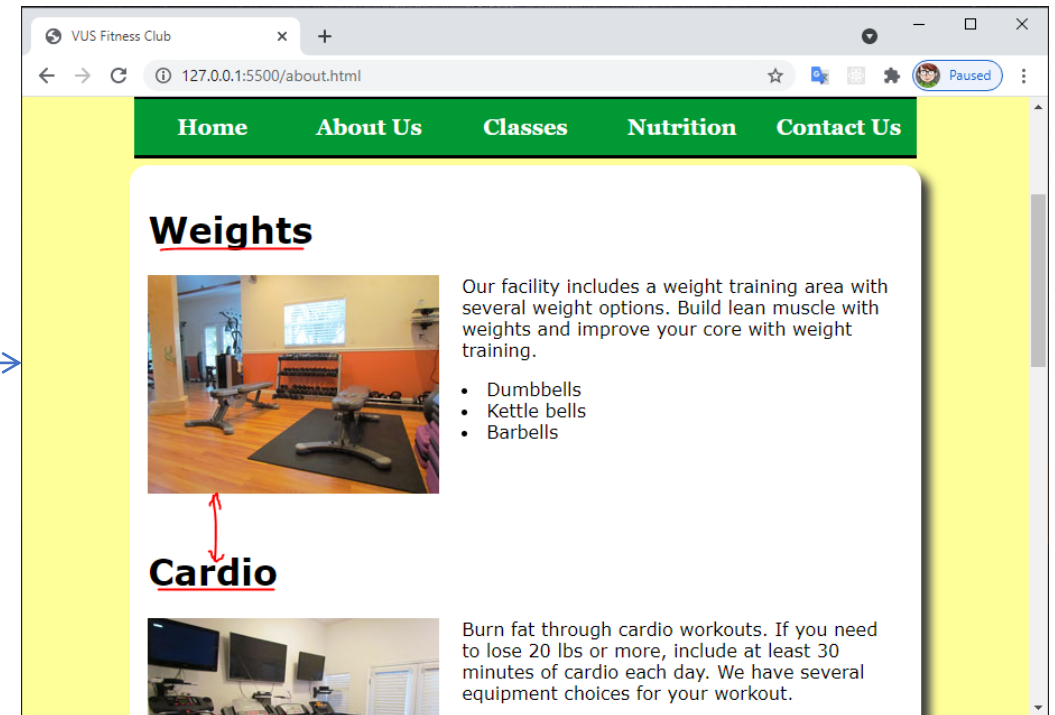
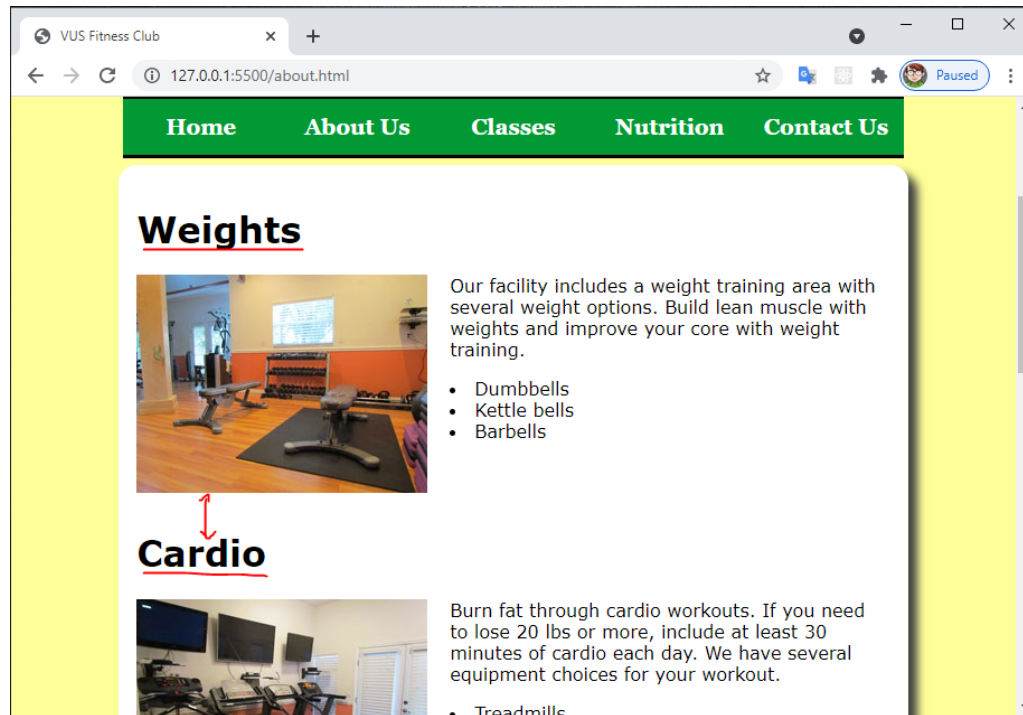
```
/* Style specifies a bottom margins for the items class */  
  
.items {  
    display: block;  
    list-style-position: inside;  
    margin-bottom: 5em; → 6em  
}
```



Thiết kế định kiểu CSS cho Desktop Viewports

❑ Tạo Media Query cho Desktop Viewport:

♦ Quan sát giao diện website VUS Fitness Club sau khi chỉnh sửa **CSS Media Queries** cho phần nội dung chính, thay đổi kích cỡ từ **viewport tablet** sang **viewport desktop**.





Thiết kế định kiểu CSS cho Desktop Viewports

❑ Tạo Media Query cho Desktop Viewport:

♦ **Chỉ** sửa độ rộng **viewport** mới cho các **media queries** máy tính bảng (tablet: sửa 481px thành 660px) và **máy tính để bàn** (desktop: sửa 769px thành 830px).

```
/* Media query for tablet viewport targets s
@media only screen and (min-width: 481px) {
  /* Style specifies a horizontal display
  nav li { ...
}

```

660px

```
@media only screen and (min-width: 769px) {
  /* Style specifies a width for container */
  #container { ...
}
/* Style specifies margin and padding for the un
nav ul { ...
}
/* Style for navigation list items specifies a b
nav li { ...
}
/* Style for navigation links specifies display
nav li a { ...
}

```

830px

```
nav li {
  background-color: #009933;
  border-radius: 2em;
  list-style-type: none;
  margin: 0.3em;
  padding: 0.4em;
  white-space: nowrap;
}
```

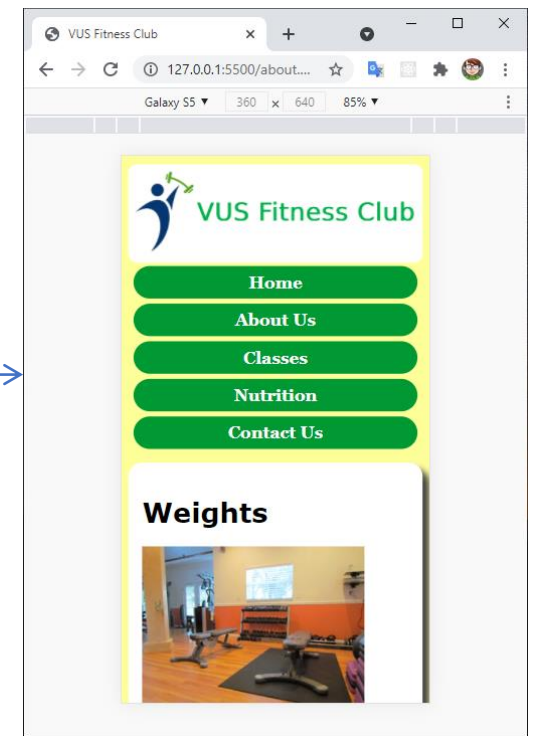
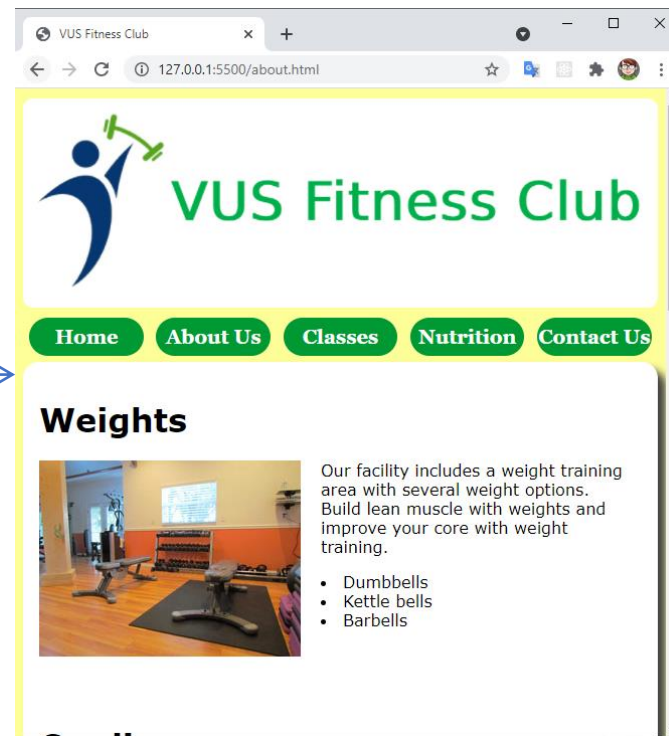
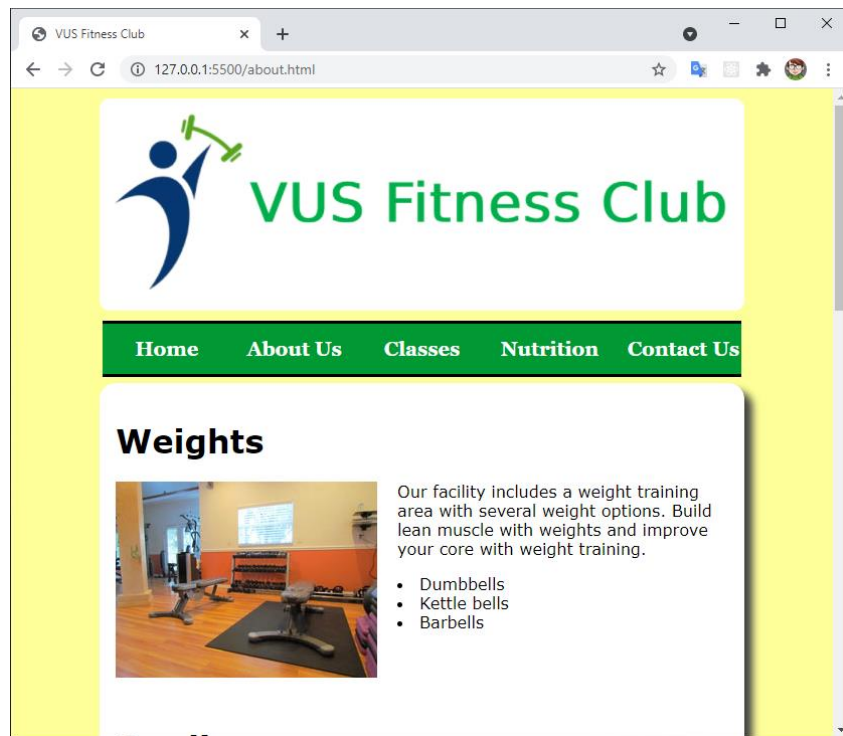
Tránh xảy ra
lỗi mất chữ
khi co giãn
màn hình.



Thiết kế định kiểu CSS cho Desktop Viewports

❑ Tạo Media Query cho Desktop Viewport:

♦ Quan sát giao diện website VUS Fitness Club sau khi chỉnh sửa **CSS Media Queries** cho phần nội dung chính, thay đổi kích cỡ từ **viewport desktop** sang **viewport tablet** sang **viewport mobile** và quan sát kết quả:





Thiết kế các lớp giả sử dụng Pseudo-Classes

❑ Giới thiệu Pseudo-Classes:

- ♦ Bạn đã từng **truy cập 1 website** và **chú ý thấy rằng liên kết để truy cập đó bị đổi màu khi** bạn **trở vào hoặc nhấn vào** chúng?
- ♦ **Màu sắc của liên kết thay đổi dựa vào việc sử dụng các lớp giả (pseudo-classes) trong định kiểu CSS.**
- ♦ **Pseudo-classes cho phép bạn thay đổi định kiểu của một liên kết dựa trên 4 trạng thái của liên kết: link, visited, hover và active.**

♦ **Cú pháp:**

```
selector: pseudo-class {  
    property: value;  
}
```

Pseudo-class	Dùng để định kiểu
:link	Liên kết chưa được truy cập lần nào hết
:visited	Liên kết đã được nhấn vào ít nhất 1 lần
:hover	Liên kết khi con trỏ chuột được lướt qua liên kết này
:active	Liên kết tại thời điểm mà nó được nhấn vào

- ♦ a:link, a:visited, a:hover, and a:active



Thiết kế các lớp giả sử dụng Pseudo-Classes

❑ Sử dụng Pseudo-Classes:

♦ Áp dụng các lớp giả **Pseudo-classes** để cải thiện các liên kết trên giao diện website **VUS Fitness Club**. Sau đó, quan sát các liên kết trên giao diện và tương tác thử với chúng.

na li a:visited
{..} dùng thử
để biết công
dụng. Sau đó,
bỏ CSS này đi!

```
/* Pseudo-classes for links */  
  
nav li a:link {  
    color: #FFFFFF;  
}  
  
nav li a:visited {  
    color: #FFFF99;  
}  
  
nav li a:hover {  
    color: #FFFF00;  
    font-style: italic;  
}  
  
nav li a:active {  
    color: #6600CC;  
}
```





Thiết kế độ dốc sử dụng Gradients

❑ Giới thiệu Gradients trong CSS3:

- ♦ Bạn **đã tích hợp một số thuộc tính CSS3 vào** trang web **VUS Fitness Club**, chẳng hạn như các góc tròn (**rounded corners**) và bóng hộp (**box-shadow**),...
- ♦ Ngoài ra, **CSS3 đã giới thiệu một số thuộc tính mới khác** để **nâng cao** hơn nữa trải nghiệm người dùng khi sử dụng trang web, một trong số đó là **gradient**.
- ♦ **Gradient** (độ dốc) là **sự chuyển dần từ màu này sang màu khác**. **CSS3 có hai loại gradient**: tuyến tính (**linear**) và xuyên tâm (**radial**).





Thiết kế độ dốc sử dụng Gradients

❑ Các loại gradients của CSS3:

♦ Các lớp giả **Linear gradients** có thể chuyển đổi từ nhiều góc độ khác nhau: **từ trên xuống dưới** (mặc định), **lên trên**, **sang trái**, **sang phải** hoặc **theo đường chéo**.

Hướng	Ví dụ
top to bottom (mặc định)	<pre>body { background: -moz-linear-gradient: (white, blue); background: -o-linear-gradient: (white, blue); background: -webkit-linear-gradient: (white, blue); background: linear-gradient: (white, blue); }</pre>
left to right (trái qua phải)	<pre>body { background: -moz-linear-gradient: (right, white, blue); background: -o-linear-gradient: (right, white, blue); background: -webkit-linear-gradient: (left, white, blue); background: linear-gradient: (to right, white, blue); }</pre>



Thiết kế độ dốc sử dụng Gradients

❑ Các loại gradients của CSS3:

♦ Các lớp giả **Linear gradients** có thể chuyển đổi từ nhiều góc độ khác nhau: **từ trên xuống dưới** (mặc định), **lên trên**, **sang trái**, **sang phải** hoặc **theo đường chéo**.

Hướng	Ví dụ
diagonal (đường chéo)	<pre>body { background: -moz-linear-gradient: (bottom right, white, blue); background: -o-linear-gradient: (bottom right, white, blue); background: -webkit-linear-gradient: (left top, white, blue); background: linear-gradient: (to bottom right, white, blue); }</pre>
specified angle (góc xác định)	<pre>body { background: -moz-linear-gradient: (180deg, white, blue); background: -o-linear-gradient: (180deg, white, blue); background: -webkit-linear-gradient: (180deg, white, blue); background: linear-gradient: (180deg, white, blue); }</pre>



Thiết kế độ dốc sử dụng Gradients

❑ Các loại gradients của CSS3:

- ♦ Các lớp giả **Radial gradients** được chỉ định bởi tâm của chúng. Màu **bắt đầu ở trung tâm** và chuyển đổi theo hướng xuyên tâm **sang màu** hoặc **màu khác**.
- ♦ Để tạo một **gradient xuyên tâm**, bạn phải chỉ định ít nhất hai màu.
- ♦ Ví dụ:

```
body {  
    background: -moz-radial-gradient(red, green, blue);  
    background: -o-radial-gradient(red, green, blue);  
    background: -webkit-radial-gradient(red, green, blue);  
    background: radial-gradient(red, green, blue);  
}
```

- ♦ **Linear gradients** được sử dụng phổ biến hơn nên chúng ta sẽ áp dụng kỹ thuật này vào dự án VUS Fitness Club.



Thiết kế độ dốc sử dụng Gradients

❑ Sử dụng Gradients của CSS3:

- ♦ Áp dụng các lớp giả **Linear gradients** để cải thiện background trên giao diện website **VUS Fitness Club** bằng cách thêm mã định kiểu CSS vào media queries.

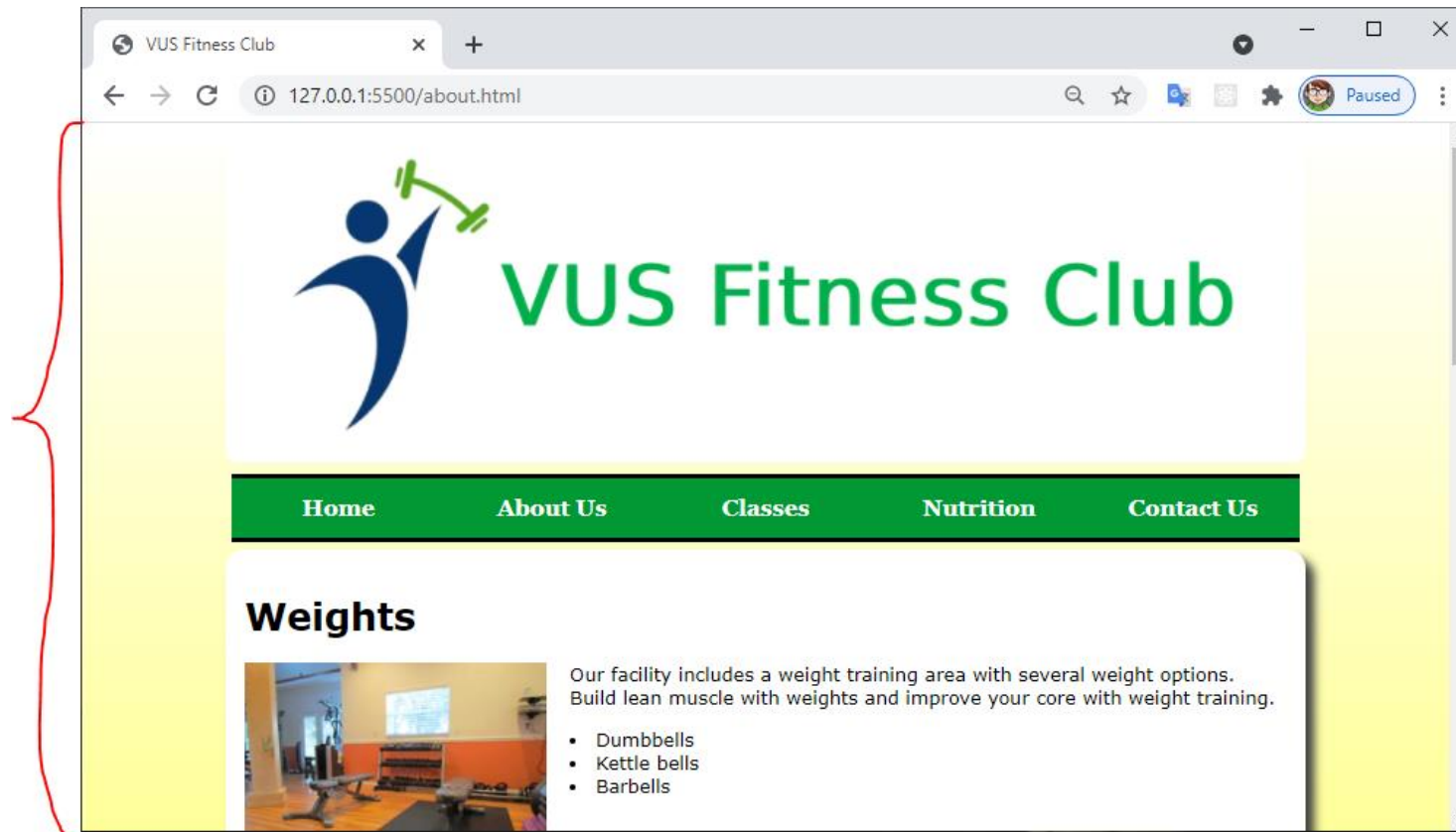
```
/* Style specifies a gradient for the body element */
body {
  background: -moz-linear-gradient(■ #FFFFFF, ■ #FFFF99);
  /* Firefox */
  background: -o-linear-gradient(■ #FFFFFF, ■ #FFFF99);
  /* Opera */
  background: -webkit-linear-gradient(■ #FFFFFF, ■ #FFFF99);
  /* Chrome */
  background: linear-gradient(■ #FFFFFF, ■ #FFFF99);
  background-attachment: fixed;
}
```



Thiết kế độ dốc sử dụng Gradients

❑ Sử dụng Gradients của CSS3:

♦ Áp dụng các lớp giả **Linear gradients** để cải thiện background trên giao diện website **VUS Fitness Club**. Sau đó, quan sát background trên giao diện để nhận biết kết quả.





Tổng kết nội dung bài học

- ☐ Sử dụng truy vấn Media Query
- ☐ Thiết kế định kiểu CSS cho Tablet Viewports
- ☐ Thiết kế định kiểu CSS cho Desktop Viewports
- ☐ Thiết kế các lớp giả sử dụng Pseudo-Classes
- ☐ Thiết kế độ dốc sử dụng Gradients

Let's
Recap

