



Bộ môn Công nghệ Phần mềm
Viện CNTT & TT
Trường Đại học Bách Khoa Hà Nội

CÔNG NGHỆ WEB TIỀN TIẾN

Bài 08B: WEB DI ĐỘNG

MỤC LỤC

1. Đặt vấn đề
2. Responsive
3. Thẻ meta viewport
4. CSS media
5. Giả lập trên trình duyệt
6. Relatives

MỤC LỤC

1. **Đặt vấn đề**
2. Responsive
3. Thẻ meta viewport
4. CSS media
5. Giả lập trên trình duyệt
6. Relatives

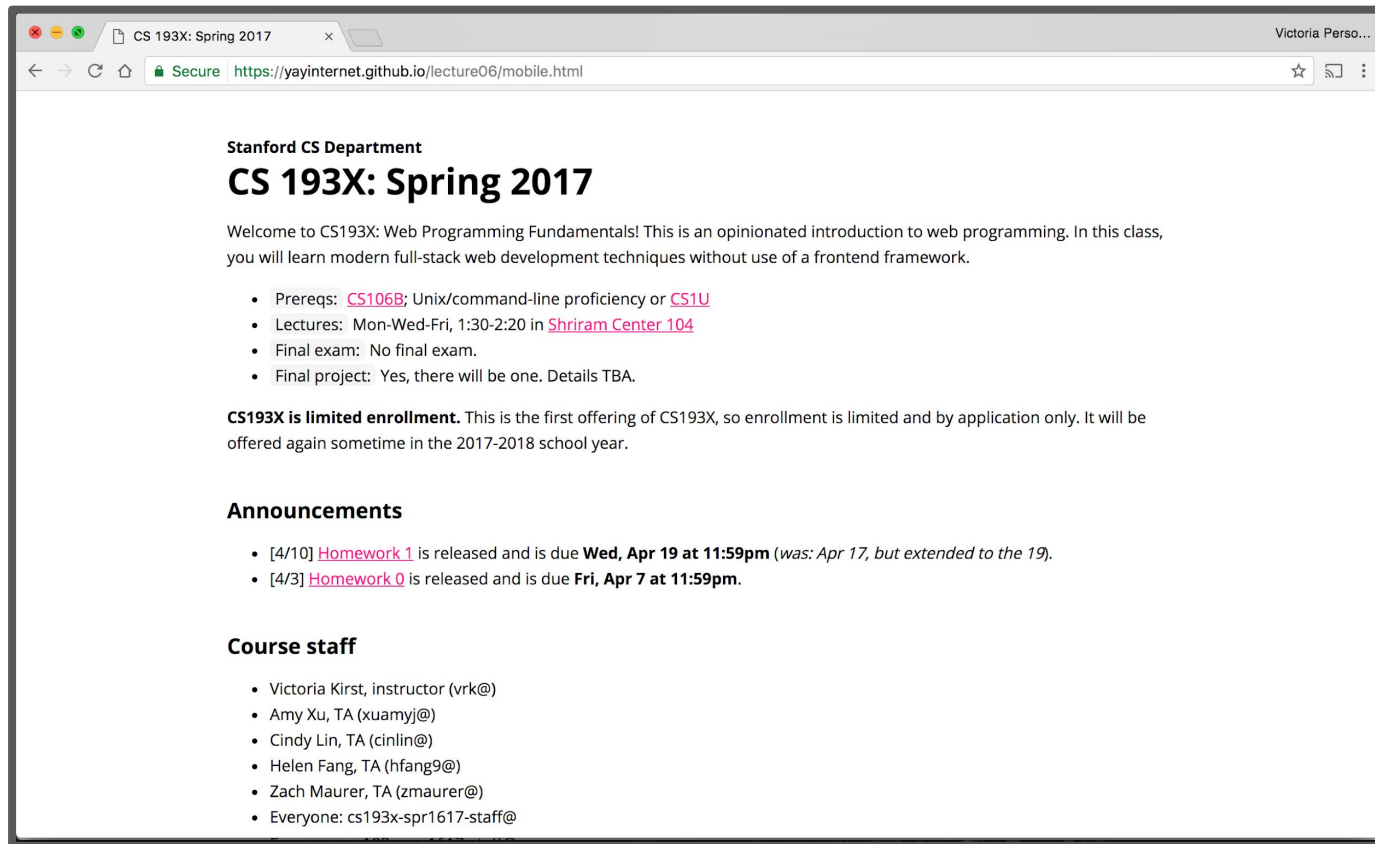
1. ĐẶT VẤN ĐỀ

Mặc dù đã có các ứng dụng thương mại điện tử, vì sao nhiều lúc chúng ta vẫn cần có các trang web để hiển thị nội dung (trên Web View chẳng hạn)?

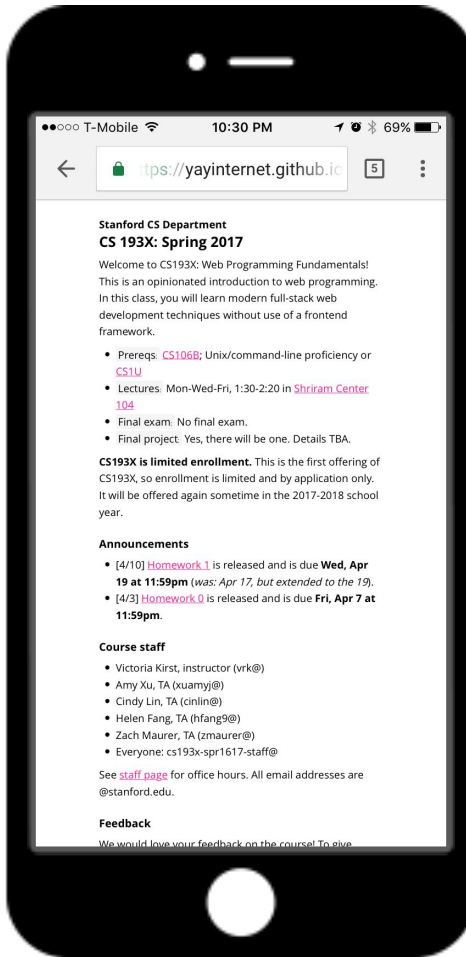
- Khi muốn hiển thị nội dung mà luôn cần phải thay đổi => cập nhật qua ứng dụng là không khả thi
- Nhiều trang web thanh toán yêu cầu phải truy cập vào website của họ để nhập mật khẩu.

CÂU HỎI: THEO EM CÒN CÓ TRƯỜNG HỢP NÀO NỮA CẦN PHẢI SỬ DỤNG WEB MOBILE?

Khi ta xây dựng ra một trang web như sau



CÂU HỎI: NÓ SẼ TRÔNG NHƯ NÀO KHI XEM TRÊN ĐIỆN THOẠI



Không tệ lắm... nhưng khá nhỏ và khó đọc

MỤC LỤC

1. Đặt vấn đề
2. **Responsive**
3. Thẻ meta viewport
4. CSS media
5. Giả lập trên trình duyệt
6. Relatives

2. Phong cách thiết kế Responsive

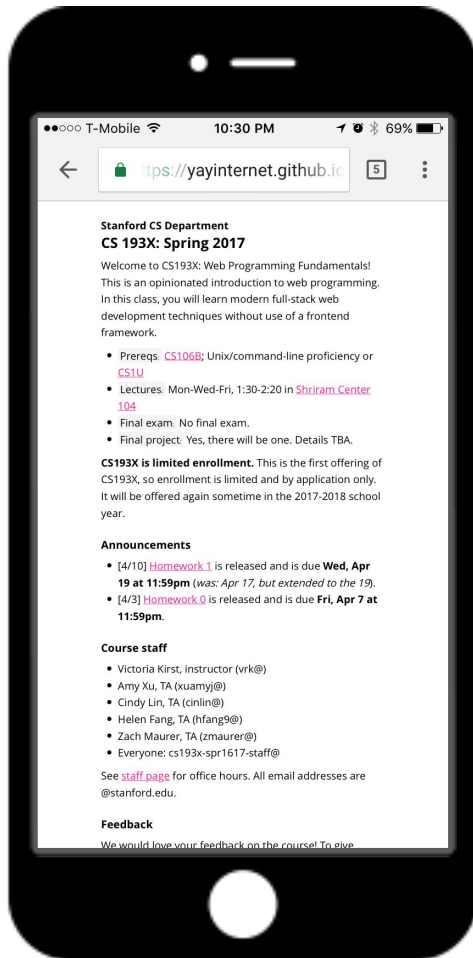
Chúng ta cần tạo ra những định dạng bằng CSS theo cách sao cho trang web hiển thị đẹp để trên nhiều kích thước màn hình khác nhau

- **Màn hình máy tính 27"**
- **Macbook Air**
- **Samsung Galaxy S7**
- **iPhone 7**
- **iPad**

CÂU HỎI: LÀM THẾ NÀO ĐỂ ĐẠT ĐƯỢC ĐIỀU ĐÓ

Liệu ta có phải tạo ra file CSS
cho nhiều kích thước khác nhau

Kích thước màn hình điện thoại



Trừ khi có các cấu hình khác khác thông qua thẻ **HTML** hoặc **CSS**, các trình duyệt trên điện thoại hiển thị các trang web ở chiều rộng màn hình máy bàn (~ 1000px), sau đó "thu nhỏ" cho đến khi toàn bộ trang vừa với màn hình.

(Đó là lý do tại sao đôi khi bạn nhận được các trang web có phong chữ nhỏ xíu trên điện thoại của mình: các trang web này không hỗ trợ cho điện thoại di động.)

([Đọc thêm ở đây](#))

MỤC LỤC

1. Đặt vấn đề
2. Responsive
3. **Thẻ meta viewport**
4. CSS media
5. Giả lập trên trình duyệt
6. Relatives

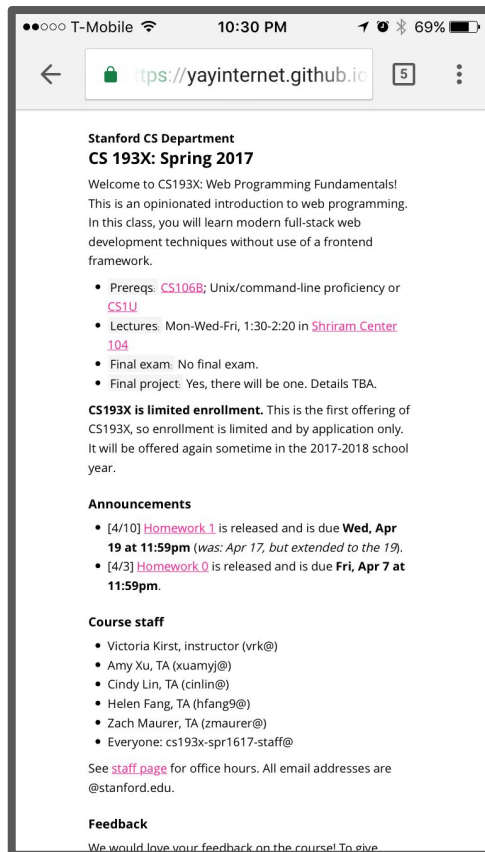
Thẻ Meta viewport

Để ngăn trình duyệt điện thoại hiển thị trang ở chiều rộng màn hình rồi sau đó thu nhỏ, hãy sử dụng thẻ **meta viewport**:

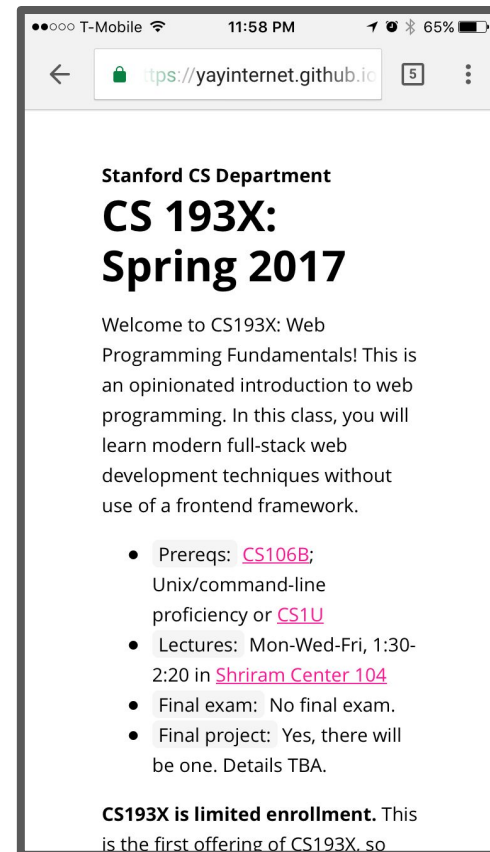
```
<meta name="viewport"  
content="width=device-width, initial-scale=1">
```

Nó nằm trong phần <head> của file **HTML**.
(Cùng cấp với <title>, <link>, và các thành phần metadata khác.)

The Meta viewport



KHÔNG dùng thẻ
meta viewport



CÓ thẻ **meta**
viewport

Thẻ Meta viewport

```
<meta name="viewport"  
content="width=device-width, initial-scale=1">
```

- **name=viewport**: "Trình duyệt, tôi sẽ cho biết tôi muốn khung nhìn trông như thế nào."
- **width=device-width**: "Chiều rộng của khung nhìn phải luôn bắt đầu bằng chiều rộng của thiết bị."
- **initial-scale=1**: "Bắt đầu ở mức thu phóng 100%."

Thẻ Meta viewport

```
<meta name="viewport"  
content="width=device-width, initial-scale=1">
```

(Hãy luôn sử dụng đoạn mã **HTML** này trong trang web của bạn)

MỤC LỤC

1. Đặt vấn đề
2. Responsive
3. Thẻ meta viewport
4. **CSS media**
5. Giả lập trên trình duyệt
6. Relatives

Các điều chỉnh khác

Thẻ **meta viewport** giúp chúng ta đạt được một giao diện gần như rất ổn, nhưng vẫn cần một vài điều chỉnh.

Ví dụ: **margin** căn lề có vẻ quá lớn trên thiết bị di động. Chúng ta có thể đặt một giá trị khác cho thiết bị di động không?



CSS media

Có thể dùng **CSS media query** để thay đổi cách hiển thị dựa trên đặc tính của thiết bị:

```
@media (max-width: 500px) {  
  article {  
    margin: 0 2px;  
  }  
}
```

Ngoài ra có thể tạo ra nhiều loại media query khác như [tại đây](#)



MỤC LỤC

1. Đặt vấn đề
2. Responsive
3. Thẻ meta viewport
4. CSS media
5. **Giả lập trên trình duyệt**
6. Relatives

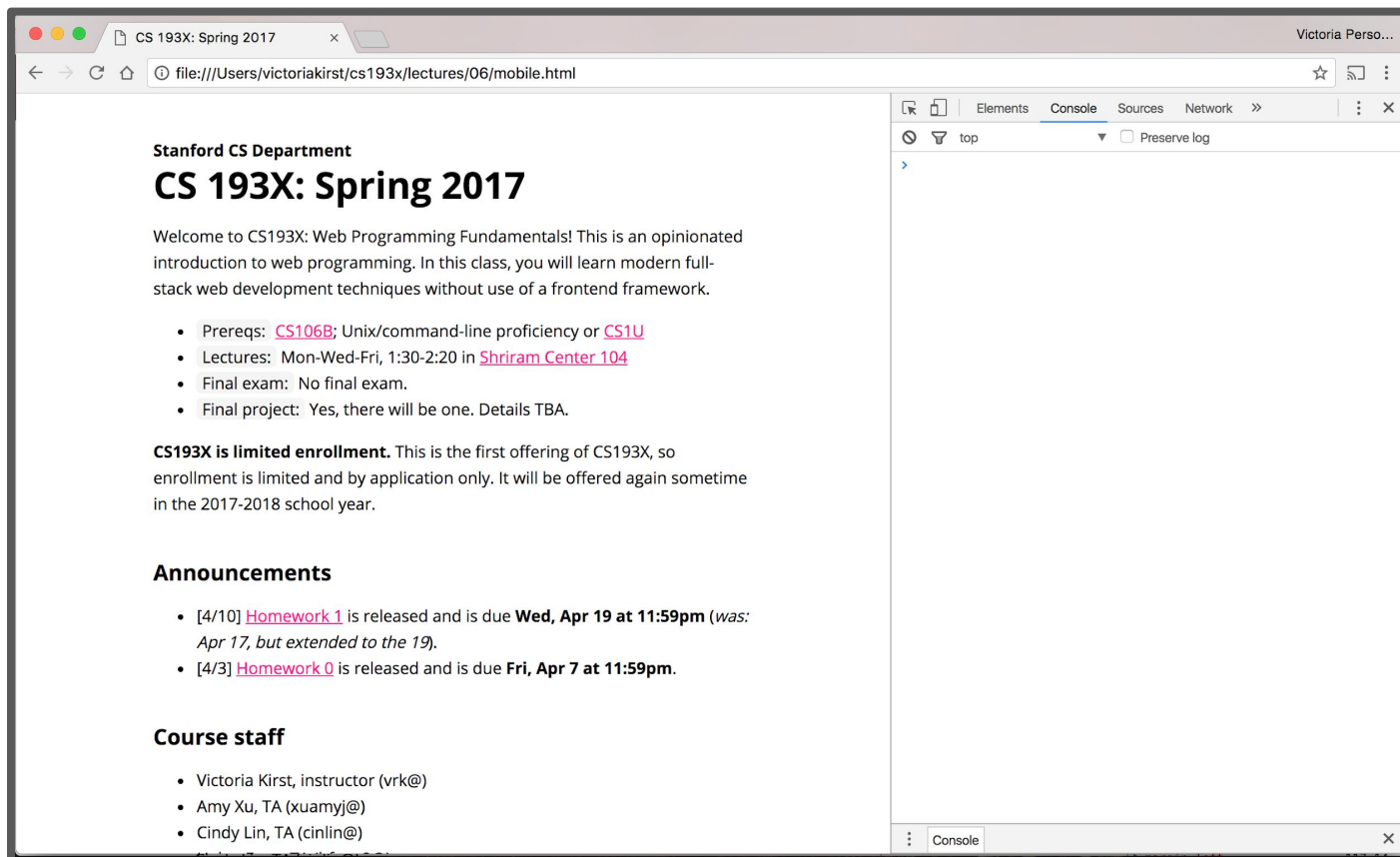
Kiểm thử giao diện?

Câu hỏi thực tế: **Nên kiểm tra bố cục trên thiết bị di động như thế nào?**

- Đăng mã **HTML + CSS** tại một địa chỉ Internet rồi điều hướng cho điện thoại đi đến URL?
- Có cách nào để kết nối điện thoại với máy tính của bạn không?
- Có thể truy cập website bằng trình giả lập **Android/iOS** không?
- Khác?

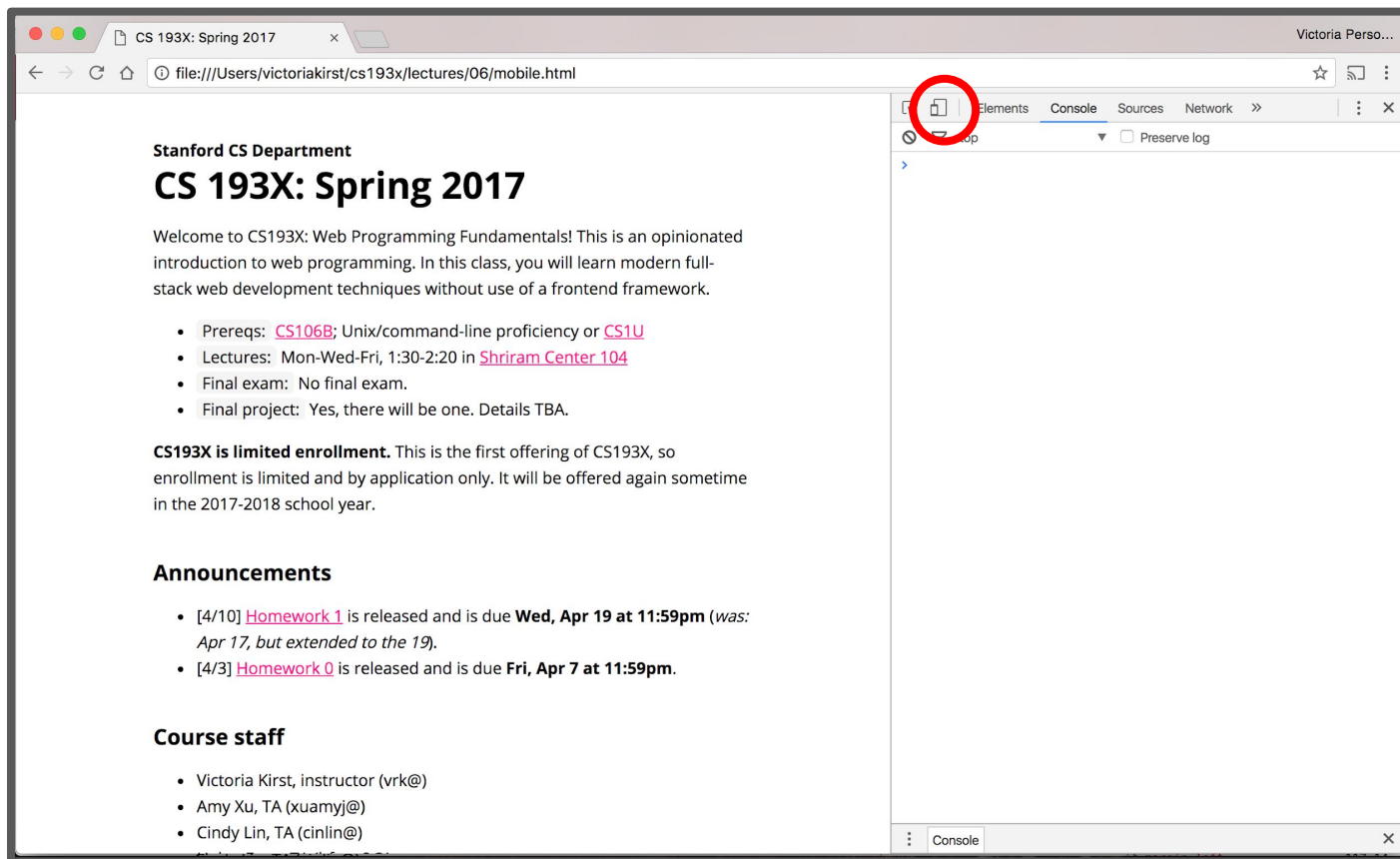
Chế độ Chrome device

Bạn có thể xem cách một trang web được hiển thị trên di động thông qua chế độ Chrome device:



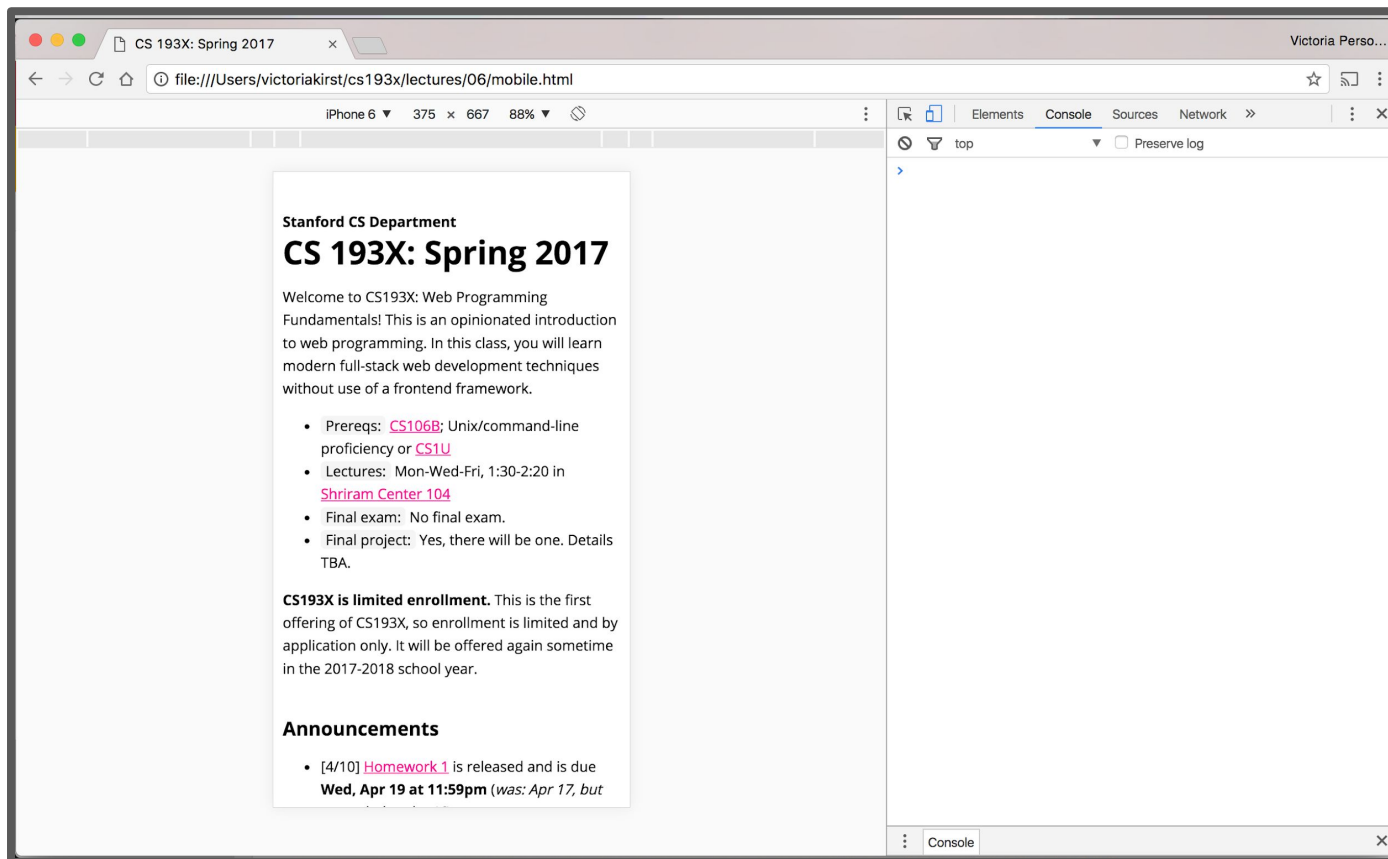
Chế độ Chrome device

Bằng cách nhấn vào nút ở góc trên màn hình:



Chế độ Chrome device

Từ đó ta có thể nhìn thấy giao diện web được hiển thị ra sao trên điện thoại



Chế độ Chrome device

Ưu điểm của Chrome device:

- Cực kỳ tiện lợi
- Độ chính xác rất cao

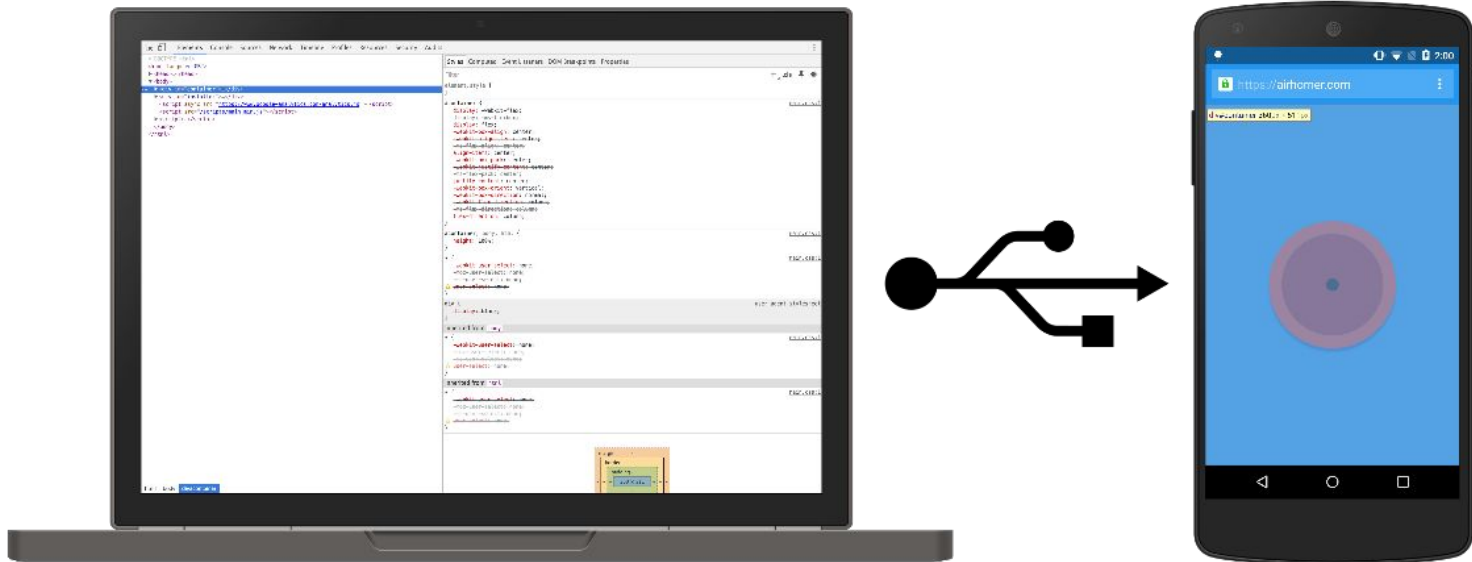
Nhược điểm của Chrome device:

- Không phải lúc nào cũng chính xác - nhất là trên **iPhone**
- Thỉnh thoảng vẫn bị lỗi không hiển thị được
- Không mô phỏng được các vấn đề về hiệu năng

Nhìn chung, cuối cùng thì vẫn phải kiểm tra trên thiết bị thật.

Chế độ gỡ rối từ xa trên Chrome

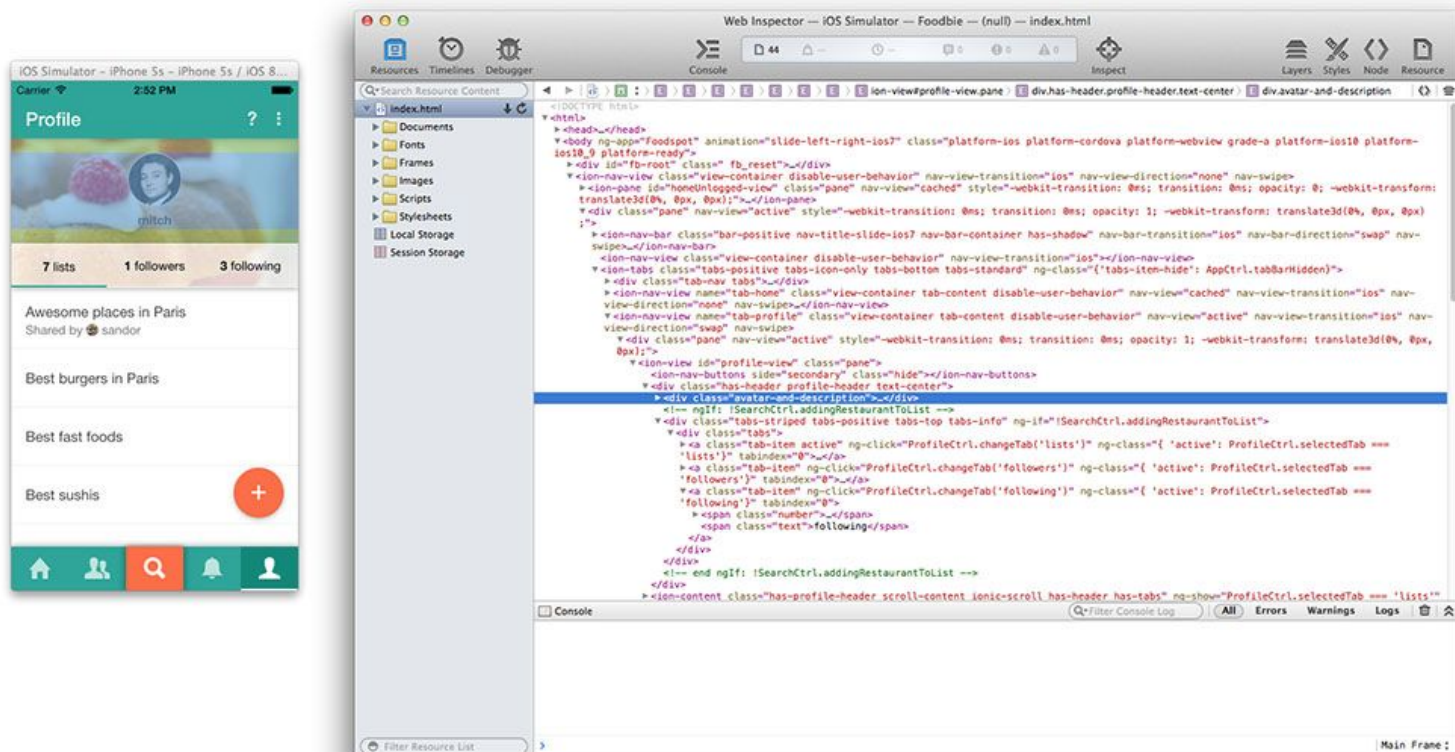
Nếu ta có điện thoại **Android**, có thể gỡ rối trang web trên điện thoại thông qua [Chrome remote debugging](#).



(Bạn cũng có thể tải một máy chủ chạy cục bộ trên máy tính xách tay, rồi truy cập bằng điện thoại thông qua [port forwarding](#).
Chưa cần thiết sử dụng máy chủ thật.)

Chế độ gỡ rối từ xa trên Safari

Nếu ta có điện thoại **iPhone**, có thể gỡ rối trang web trên điện thoại thông qua Safari remote debugging.



MỤC LỤC

1. Đặt vấn đề
2. Responsive
3. Thẻ meta viewport
4. CSS media
5. Giả lập trên trình duyệt
6. **Relatives**

Relative

Nếu có thể, sử dụng các đơn vị tương đối (**relative units**) (như phần trăm) thay vì các đơn vị tuyệt đối (như **px**).

Ưu điểm:

- Nhiều khả năng hoạt động trên các kích thước màn hình khác nhau
- 10%/80%/10% thay vì 122px/926px/122px

CÂU HỎI: LIỆU CÓ NÊN DÙNG ĐƠN VỊ TƯƠNG ĐỐI CHO FONT-SIZE?

Phần trăm

Có thể khai báo kích thước **font** chữ theo phần trăm:

```
<body>  
  <h1>This is 60px</h1>  
  <p>This is 15px</p>  
</body>
```

```
body {  
  font-size: 30px;  
}  
  
h1 {  
  font-size: 200%;  
}  
  
p {  
  font-size: 50%;  
}
```

This is 60px

This is 15px

([CodePen](#))

Phần trăm

Phần trăm trên kích thước **font** chữ hoạt động chính xác giống như phần trăm trên chiều rộng và chiều cao, do có liên quan đến phần tử **parent**:

```
<div>  
  This is 60px  
  <p>This is 45px</p>  
</div>
```

```
body {  
  font-size: 30px;  
}  
  
div {  
  font-size: 200%;  
}  
  
p {  
  font-size: 75%;  
}
```

This is 60px

This is 45px

([CodePen](#))

Phần trăm

Phần trăm trên kích thước **font** chữ hoạt động chính xác giống như phần trăm trên chiều rộng và chiều cao, do có liên quan đến phần tử **parent**:

```
<div>
  This is 60px
  <p>This is 45px</p>
</div>
```

```
body {
  font-size: 30px;
}

div {
  font-size: 200%;
}

p {
  font-size: 75%;
}
```

This is 60px

This is 45px

p có cỡ chữ 75% so với phần tử parent, vốn có cỡ 200% của 30px.

p sẽ có cỡ: $.75 * 2 * 30 = 45\text{px}$

([CodePen](#))

Đơn vị em

Nhưng thay vì tỷ lệ phần trăm, kích thước font chữ tương đối thường được xác định theo em:

- em đại diện cho kích thước font chữ được tính toán của phần tử
 - 1em = kích thước font chữ kế thừa
 - 2em = 2 lần kích thước font chữ kế thừa

Nói cách khác,

font-size: 1em; tương tự như font-size: 100%;

Đơn vị em: 0.5em

```
<body>  
  <h1>This is 60px</h1>  
  <p>This is 15px</p>  
</body>
```

```
body {  
  font-size: 30px;  
}  
  
div {  
  font-size: 2em;  
}  
  
p {  
  font-size: .5em;  
}
```

This is 60px

This is 15px

([CodePen](#))

Đơn vị em: 0.75em

```
<div>  
  This is 60px  
  <p>This is 45px</p>  
</div>
```

```
body {  
  font-size: 30px;  
}  
  
div {  
  font-size: 2em;  
}  
  
p {  
  font-size: .75em;  
}
```

This is 60px

This is 45px

([CodePen](#))

Đơn vị em: 0.75em

```
<div>  
  This is 60px  
  <p>This is 45px</p>  
</div>
```

```
body {  
  font-size: 30px;  
}  
  
div {  
  font-size: 2em;  
}  
  
p {  
  font-size: .75em;  
}
```

This is 60px

This is 45px

phần tử **parent** có cỡ chữ là 2em, Tức 60px. Vậy 0.75em là $0.75 \times 60 = 45\text{px}$.

```
<body>
  This is
  <h1>
    <strong>120px</strong>
  </h1>
</body>
```

```
body {
  font-size: 30px;
}

strong {
  font-size: 2em;
}
```

This is

120px

Tại sao trong thẻ **** lại có cỡ **120px** thay vì **60px**?

```
<body>
  This is
  <h1>
    <strong>120px</strong>
  </h1>
</body>
```

```
body {
  font-size: 30px;
}

strong {
  font-size: 2em;
}
```

This is

120px

([CodePen](#))

```
h1 {                                user agent stylesheet
  display: block;
  font-size: 2em;
  -webkit-margin-before: 0.67em;
  -webkit-margin-after: 0.67em;
  -webkit-margin-start: 0px;
  -webkit-margin-end: 0px;
  font-weight: bold;
}
```

Trong **Chrome Inspector**, ta thấy cỡ chữ mặc định cho **h1** là **2em**. Vậy **$30 * 2 * 2 = 120px$** .

Đơn vị rem

Nếu **không** muốn cỡ chữ tương đối bị ảnh hưởng bởi phần tử **parent**, hãy sử dụng **rem**:

- **rem** đại diện cho cỡ chữ của phần tử gốc
 - **1rem** = cỡ chữ của phần tử gốc (thẻ **html**)
 - **2rem** = 2 lần cỡ chữ của phần tử gốc

Đơn vị rem

```
<body>
  <div>
    This is 60px
    <p>This is 22.5px</p>
  </div>
</body>
```

```
html {
  font-size: 30px;
}

div {
  font-size: 2rem;
}

p {
  font-size: .75rem;
}
```

This is 60px

This is 22.5px

([CodePen](#))

Đơn vị rem

```
<body>
  <div>
    This is 60px
    <p>This is 22.5px</p>
  </div>
</body>
```

```
html {
  font-size: 30px;
}

div {
  font-size: 2rem;
}

p {
  font-size: .75rem;
}
```

This is 60px

This is 22.5px

font-size đặt cho thẻ
html, không phải thẻ body
(hoặc các thẻ khác)

([CodePen](#))

Đơn vị rem

```
<body>
  <div>
    This is 60px
    <p>This is 22.5px</p>
  </div>
</body>
```

```
html {
  font-size: 30px;
}
```

```
div {
  font-size: 2rem;
}
```

```
p {
  font-size: .75rem;
}
```

This is 60px

This is 22.5px

.75em tính theo giá trị cỡ chữ của thẻ **html**, là 30px, vậy $30 \times .75 = 22.5\text{px}$.

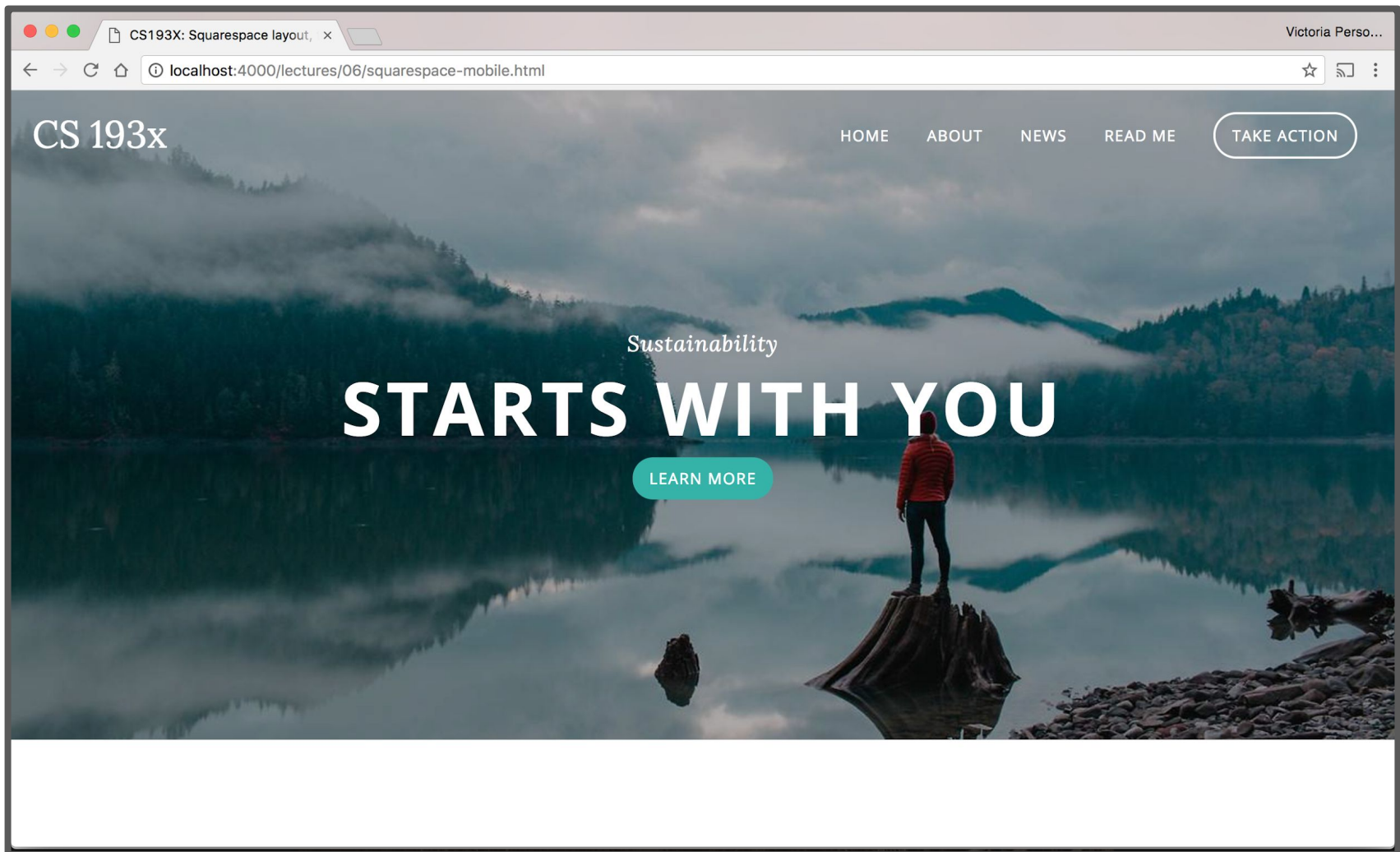
([CodePen](#))

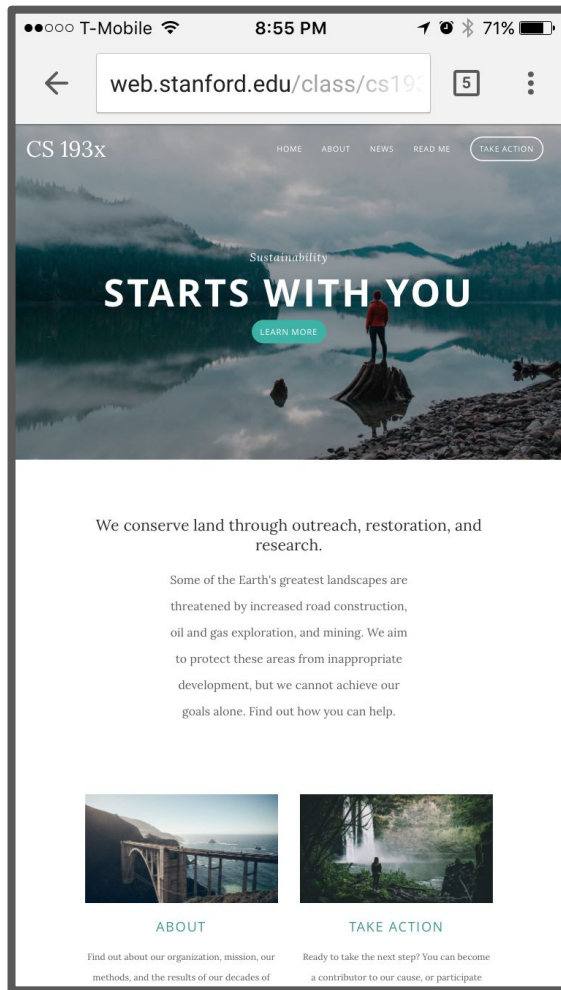
Relative

Sử dụng relative cho cả kích thước height và width:

- Nhìn chung **em** và **rem** hữu ích hơn so với phần trăm
 - Đặc biệt với cỡ chữ, **em** dễ hiểu hơn, và được ưa thích sử dụng hơn
- Dùng **rem** để tránh các tác động của các phần tử **parent**
- Bên cạnh **font-size**, có thể dùng **em/rem** cho:
 - **line-height**
 - **margin-top**
 - **margin-bottom**

Bố cục này sẽ trông như thế nào trên màn hình điện thoại với thẻ **meta viewport**





KHÔNG dùng thẻ
meta viewport



CÓ thẻ **meta**
viewport

CS 193x

☰

Sustainability


STARTS WITH YOU

LEARN MORE

We conserve land through outreach, restoration, and research.


Some of the Earth's greatest landscapes are threatened by increased road construction, oil and gas exploration, and mining. We aim to protect these areas from inappropriate development, but we cannot achieve our goals alone. Find out how you can help.

protect these areas from inappropriate development, but we cannot achieve our goals alone. Find out how you can help.



ABOUT

Find out about our organization, mission, our methods, and the results of our decades of advocacy.



Thực tế, phải dùng nhiều kĩ thuật khác nhau để đạt được bố cục đẹp nhất cho điện thoại

Kết luận

- Luôn dùng thẻ **meta viewport**
- Dùng **@media queries** để điều chỉnh với kích thước đặc thù của thiết bị, chẳng hạn chiều rộng màn hình
- Dùng **Chrome Device Mode** để mô phỏng việc hiển thị trang web trên màn hình điện thoại
- Với **height** và **width**, nên dùng phần trăm
- Với **fonts**, dùng **em** và **rem**
- Cố giảm **sự phụ thuộc** giữa các giao diện (Việc thay đổi chiều rộng của một vùng chứa buộc bạn phải thay đổi 15 thuộc tính khác của nó)

Đọc thêm [responsive web design](#)

Câu hỏi

