**Web front-end (15) - HTML - Hiển thị ảnh linh hoạt**

**1.1.1       Hiển thị ảnh linh hoạt (responsive image)**

Responsive (tính từ): có nghĩa là khả năng thích ứng, có tính linh hoạt, tùy chỉnh (adapt).

Là khả năng trang web có thể tự thay đổi giao diện (gồm nội dung và bố cục – content and layout) tùy theo kích thước màn hình ứng với mỗi thiết bị duyệt web, như điện thoại (smartphone), máy tính bảng (tablet), máy tính, hay tivi thì trang web sẽ luôn có giao diện phù hợp, đẹp, dễ đọc.

Ví dụ, với hình ảnh, nếu người dùng đang sử dụng màn hình nhỏ, mà trang web lại tải về một hình ảnh lớn thì sẽ làm chậm quá trình hiển thị, lãng phí tài nguyên thiết bị; ngược lại nếu người dùng đang sử dụng màn hình lớn mà lại tải về một hình ảnh nhỏ, độ phân giải thấp thì sẽ làm cho hình ảnh bị mờ, không đẹp.

Kỹ thuật hay được sử dụng để tạo ra trang web có khả năng tùy chỉnh là dùng CSS, JavaScript hoặc xử lý bên phía server.

Tuy nhiên, phần này làm cho hình ảnh có tính chất tùy chỉnh. Ý tưởng là: chúng ta sẽ chuẩn bị nhiều loại hình ảnh ứng với các kích thước màn hình khác nhau; khi hiển thị trang web, trình duyệt sẽ tự động chọn hình ảnh cho phù hợp, dựa vào các yếu tố như kích thước, độ phân giải của màn hình; tốc độ đường truyền, có lưu bộ đệm (cache) hay không.

Để có hình ảnh tùy chỉnh, cần đáp ứng bốn tình huống sau:

– Cung cấp ảnh chất lượng (dung lượng cao) cho các màn hình có độ phân giải (resolution) cao

– Với mỗi ảnh, chuẩn bị nhiều tập tin với kích thước (dimension) khác nhau, để dùng cho mỗi kích thước màn hình

– Cung cấp nhiều phiên bản của một ảnh, với nội dung và kiểu dáng (portrait, landscape) khác nhau

– Cung cấp nhiều định dạng (format) khác nhau của một ảnh, để trình duyệt lựa chọn cho phù hợp.

**Hiển thị ảnh cho màn hình có độ phân giải cao**

Mọi thứ bạn thấy trên màn hình đều được tạo ra từ các ô màu, hình vuông, kích thước rất nhỏ, gọi là điểm ảnh hay phần tử ảnh (picture element), thường gọi là pixel. Các pixel do thiết bị tạo ra, nên gọi là device pixel (hoặc hardware pixel, physical pixel).

Số device pixel trên một inch gọi là độ phân giải của màn hình (pixel per inch), được gọi tắt là ppi.

Các pixel được tổ chức theo kiểu lưới (grid) (bitmap)

***Device-pixel-ratios***

Tuy nhiên, các thiết bị hiện nay có độ phân giải ngày càng cao, nghĩa là số pixel trên một inch ngày càng lớn (kích thước một pixel ngày càng nhỏ, nhìn ảnh mịn hơn)

Nếu đặt ảnh vào màn hình theo tỉ lệ 1:1 ( một ảnh có độ rộng 100 pixel sẽ được đặt vừa khít trên 100 pixel của thiết bị) thì sẽ thấy ảnh bị thu nhỏ lại. ( vì pixel đã nhỏ lại)

Để giải quyết tình trạng trên, các thiết bị sử dụng thêm một đơn vị đo khác gọi là pixel-tham-chiếu (reference pixel) hay pixel CSS. Đơn vị đo này được sử dụng trong quá trình hiển thị nội dung ra trình duyệt (layout).

Pixel-tham-chiếu còn có tên gọi khác là point (PT) trong iOS, hoặc device independent pixel (DP, DiP) trong Android hoặc CSS pixel trong CSS.

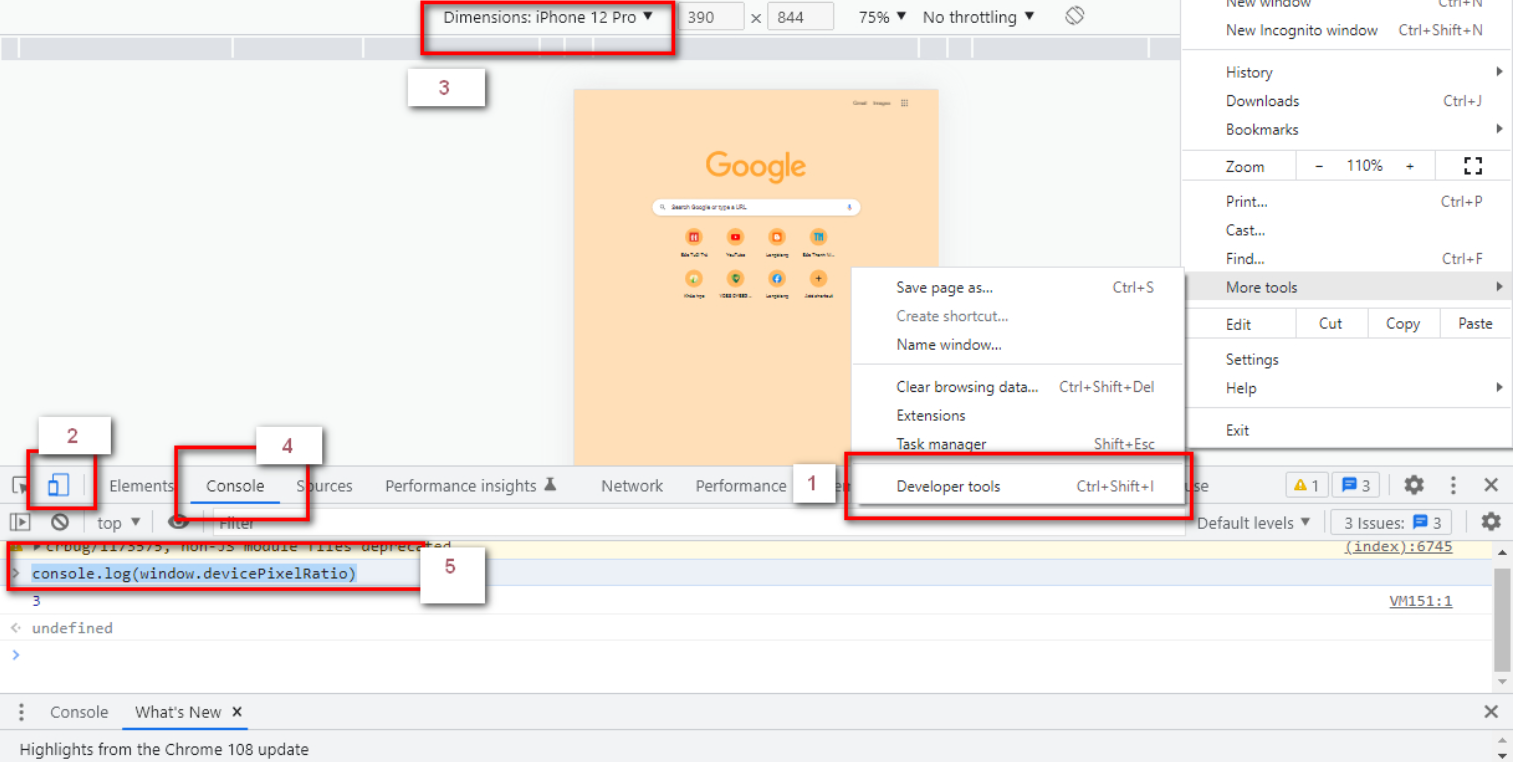
Ví dụ, iPhone 8 có kích thước màn hình là 750 x 1334 (device pixel) (độ rộng 750 ppi ~ chắc vậy), nhưng nó lại dùng lưới-hiển-thị (layout grid), có kích thước là 375 x 667 (pixel-tham-chiếu). Nghĩa là tỉ lệ giữa số điểm ảnh trên thiết bị và số điểm ảnh trên lưới-hiển-thị là 2:1 (hay 2x);

Hay nói cách khác, một điểm ảnh trên lưới-hiển-thị sẽ phủ lên 2 điểm ảnh trên màn hình thiết bị. Vậy, một cái hộp có độ rộng 100 pixel trong lưới-hiển-thị khi xuất hiện trên màn hình iPhone 8 sẽ được đặt trong 200 pixel.

Tỉ lệ giữa điểm ảnh của thiết bị và điểm ảnh của lưới-hiển-thị được gọi là device-pixel-ratios. Với các thiết bị cầm tay, device-pixel-ratios thường là 1.325x, 1.5x, 1.7x, 2.4x, 3x, thậm chí 4.x (ký hiệu ‘x’ được ngầm hiểu là device-pixel-ratios, gọi là x-descriptor).

Kết luận rút ra ở đây là nếu bạn muốn hiển thị một ảnh có độ rộng 200px trên một thiết bị có device-pixel-ratios là 1x thì bạn cần chuẩn bị một ảnh 200px, nhưng nếu hiển thị trên các thiết bị có device-pixel-ratios là 2x, 3x, thì bạn phải chuẩn bị ảnh có độ rộng là 400px và 600px tương ứng, nếu không ảnh sẽ bị mờ. Như vậy, tùy theo độ phân giải của màn hình, bạn cần lựa chọn ảnh phù hợp để tải xuống thiết bị.

Bạn có thể xem giá trị device-pixel-ratio của một số thiết bị bằng cách sau:



***Thuộc tính srcset***

Để lựa chọn ảnh phù hợp cho từng độ phân giải của màn hình, bạn sẽ sử dụng thuộc tính srcset cùng với phần tử img. Sử dụng thuộc tính srcset để liệt kê một danh sách các tập tin ảnh, khi thực thi mã HTML, trình duyệt sẽ lựa chọn ảnh phù hợp từ danh sách này.

Giá trị của thuộc tính srcset gồm các cặp image-URL và x-descriptor, mỗi cặp ngăn cách nhau bằng dấu phẩy (,). Ví dụ:

srcset="image-URL1 #x1, image-URL2 #x2"

Trong phần tử img, vẫn có thuộc tính src, đây là đường dẫn của ảnh cho trường hợp giá trị device-pixel-ratios của thiết bị là 1x. Thuộc tính alt cũng bắt buộc phải có. Ví dụ,

<img src="image-URL" alt="" srcset="image-URL1 #x1, image-URL2 #x2">

Nếu màn hình thiết bị ở chế độ phân giải chuẩn, tức device-pixel-ratios là 1x, thì sẽ hiển thị ảnh turkey-200px.jpg. Nếu màn hình có độ phân giải cao hơn thì sẽ hiển thị ảnh turkey-400px.jpg cho trường hợp device-pixel-ratios là 2x hoặc hiển thị ảnh turkey-600px.jpg cho trường hợp device-pixel-device là 3x.

<img

*src*="/images/turkey-200px.jpg" *alt*=""

*srcset*="/images/turkey-400px.jpg 2x,

        /images/turkey-600px.jpg 3x" >

**Khi nào thì nên dùng x-descriptor**

Dựa vào giá trị của x-descriptor, trình duyệt sẽ kiểm tra độ phân giải của màn hình để lựa chọn ảnh phù hợp, mà nó không quan tâm tới kích thước của màn hình (dài, rộng) cũng như kích thước của cửa sổ trình duyệt.

Vì vậy, chỉ nên sử dụng x-descriptor đối với các ảnh nhỏ, có kích thước cố định và không ảnh hưởng nhiều đến bố cục chung của trang web, như logo, biểu tượng mạng xã hội.

Việc chỉ dựa vào độ phân giải của màn hình để lựa chọn ảnh trong thực tế là chưa tối ưu. Chúng ta có thể sử dụng cách khác là lựa chọn ảnh dựa vào kích thước của màn hình (**w-descriptor)**, với màn hình nhỏ thì hiển thị ảnh nhỏ, màn hình lớn thì hiển thị ảnh lớn

**w-descriptor**

Để trang web có khả năng tự lựa chọn ảnh theo kích thước của cửa sổ trình duyệt (browser viewport) hay kích thước của màn hình (screen) (từ đây gọi chung là cửa sổ trình duyệt) chúng ta sẽ sử dụng thuộc tính srcset kết hợp với giá trị w-descriptor. w-descriptor là độ rộng của cửa sổ tính bằng pixel (actual pixel width). Kĩ thuật này được gọi là lựa chọn hình ảnh theo kích thước khung nhìn (viewport-based selection).

Chúng ta có thể sử dụng lệnh của JavaScript để xem kích thước hiện tại của cửa sổ trình duyệt

console.log(Math.max(document.documentElement.clientWidth || 0, window.innerWidth || 0))

Khi dịch mã nguồn để hiển thị trang web, trình duyệt sẽ kiểm tra độ rộng của nó, so khớp với giá trị w-descriptor trong thuộc tính srcset để lựa chọn hình ảnh cho phù hợp.

Ví dụ sau đưa ra bốn hình ảnh, ứng với bốn kích thước của cửa sổ trình duyệt, tùy theo độ rộng của màn hình, hình ảnh tương ứng sẽ được hiển thị. Ví dụ: màn hình rộng 480px (hoặc nhỏ hơn) thì ảnh strawberries-480.jpg sẽ hiển thị; màn hình rộng 960px (hoặc nhỏ hơn) thì ảnh strawberries-960.jpg sẽ hiển thị. Chữ “w” đằng sau các con số là viết tắt của w-descriptor (width descriptor), nghĩa là độ rộng.

     <img srcset="strawberries-480.jpg 480w,

          strawberries-960.jpg 960w,

          strawberries-1280.jpg 1280w,

          strawberries-2400.jpg 2400w">

Lúc chạy để kiểm tra đoạn mã trên, bạn nhớ tắt chế độ “lưu ảnh bộ nhớ đệm” (Disable cache) của trình duyệt, kết quả sẽ chính xác hơn. Để tắt chế độ “lưu ảnh bộ đệm”, trong cửa sổ Developer tools, tab Network, bỏ dấu chọn ở mục Disable cache.

**Sử dụng thuộc tính sizes**

Khi sử dụng giá trị w-descriptor thì bạn cần sử dụng thuộc tính sizes để báo cho trình duyệt biết ảnh sẽ chiếm bao nhiêu % kích thước của cửa sổ trình duyệt. Mục đích của việc này là để giúp trình duyệt có thông tin về kích thước ảnh sẽ hiển thị trên trang, nhằm tăng tốc độ xây dựng layout.

Ví dụ sau sẽ luôn hiển thị banner chiếm 100% kích thước cửa sổ trình duyệt,

     <img *src*="strawberries-640.jpg"

*alt*="baskets of ripe strawberries"

*srcset*="strawberries-480.jpg 480w,

               strawberries-960.jpg 960w,

               strawberries-1280.jpg 1280w,

               strawberries-2400.jpg 2400w"

*sizes*="100vw">

Đơn vị đo của sizes là vw (viewport width), 100vw tương đương với 100%, 50vw tương đương 50% kích thước cửa sổ trình duyệt. Ví dụ sau sẽ hiển thị ảnh chiếm 50% cửa sổ trình duyệt,

Thuộc tính sizes có thể dùng ở dạng phức tạp hơn, tùy theo độ rộng của cửa sổ trình duyệt để lựa chọn giá trị size tương ứng, ví dụ,

<img *src*="strawberries-640.jpg" *alt*="baskets of ripe strawberries"

*srcset*="strawberries-240.jpg 240w,

          strawberries-480.jpg 480w,

          strawberries-672.jpg 672w"

*sizes*="(max-width: 480px) 100vw,

          (max-width: 960px) 70vw,

          240px">

Ở đoạn mã trên, nếu độ rộng cửa sổ trình duyệt bằng hoặc nhỏ hơn 480px thì hiển thị 100% độ rộng của cửa sổ, từ 481px đến 960px thì hiển thị 70%, các trường hợp còn lại hiển thị ảnh với độ rộng đúng bằng 240px.

**Art direction (phần tử picture)**

Trong một số trường hợp, việc thay đổi kích thước ảnh không có nhiều ý nghĩa, ví dụ một ảnh lớn với nhiều chi tiết, nếu hiển thị ở màn hình rộng thì bạn có thể nhìn rõ các chi tiết; tuy nhiên, khi hiển thị ở màn hình nhỏ thì bạn không thể phân biệt được các chi tiết, hiệu quả của hình ảnh không cao. Trong trường hợp này, bạn chỉ muốn hiển thị một điểm nhấn nào đó trong ảnh. Hoặc bạn cũng muốn hiển thị ảnh ở dạng đứng (portrait) và dạng nằm ngang (landscape) là hai ảnh khác nhau.

Để chọn ảnh theo độ rộng của cửa sổ trình duyệt, bạn sử dụng phần tử picture với cú pháp sau,

<picture>

     <source *media*="(min-width: 1024px)" *srcset*="icecream-large.jpg">

     <source *media*="(min-width: 760px)" *srcset*="icecream-medium.jpg">

     <img *src*="icecream-small.jpg" *alt*="hand holding ice cream cone and

     text that reads Savor the Summer">

</picture>

Ở đoạn mã trên, nếu cửa sổ trình duyệt có kích thước từ 1024px trở lên thì dùng ảnh icecream-large.jpg, nếu cửa sổ trình duyệt có kích thước từ 760 px tới 1023px thì dùng ảnh icecream-medium.jpg, nhỏ hơn thì dùng icecream-small.jpg.