

**CHUYÊN ĐỀ TIN HỌC**  
**2014**

# **RESPONSIVE**

## **WEB DESIGN**

**LÂM NGUYỄN NHỰT QUANG**  
**11A2-P**

Năm 2010, trang web A List Apart cho đăng tải một bài viết với tiêu đề "Responsive Web Design" (tạm dịch: "thiết kế web có độ phản hồi cao"). Bài viết này nói về việc áp dụng nhiều bố cục trang web cho các loại kích cỡ màn hình khác nhau chứ không chỉ thiết kế một giao diện cố định như người ta vẫn thường làm trước đó. Kể từ đây, giới lập trình web bắt đầu để ý hơn đến responsive web design (RWD). Cộng với sự phát triển của các chuẩn HTML5 và CSS3, RWD đã trở thành một thứ quan trọng mà quản trị viên hay chủ sở hữu website cần phải nghĩ tới trong bối cảnh ngày càng nhiều thiết bị di động với đủ các kích cỡ, đủ loại độ phân giải màn hình khác nhau được tung ra thị trường. Vậy RWD là gì và ảnh hưởng của nó đến chúng ta như thế? Trong bài viết này mình xin chia sẻ một ít thông tin với các bạn.

## KHÁI NIỆM

Trước khi tìm hiểu về Responsive Web Design, trước tiên chúng ta hãy cùng tìm hiểu xem người ta làm như thế nào để có được một website hoàn chỉnh với đủ màu sắc, hình ảnh chứ không phải là hàng nghìn dòng chữ khô khan.

### Người ta bố cục trang web như thế nào?

Trước hết, để viết ra được một trang web, người ta sẽ dùng đến ngôn ngữ HTML (Hyper Text Markup Language) và lưu tập tin dưới dạng \*.html. File này thực chất chỉ là một tập tin văn bản, tuy nhiên nó bao gồm rất nhiều "thẻ" (tag) được chuẩn hóa để khi trình duyệt đọc vào nó biết phải làm gì với tag. Tag thường được viết theo một cặp, tức là tag mở và tag đóng. Ví dụ, chúng ta có tag <p> để bắt đầu một đoạn văn bản mới, sau khi viết văn bản xong thì sẽ đóng lại bằng </p>.

Chẳng hạn với đoạn code:

```
<html>
<head></head>
<body>
<p>Đây là một đoạn văn bản mới. Bạn có thể <strong>in đậm</strong>, in <em>chữ
nghiêng</em> và làm nhiều thứ khác nữa.</p>
<p>Đây là một đoạn văn bản thứ hai.</p>
</body>
</html>
```

Trong số những tag của HTML có một cặp tag đặc biệt là <div></div> (div là division, tức là "chia cắt"). Nó dùng để chia trang web thành nhiều phần khác nhau, và mỗi phần sẽ đại diện cho một bộ phận của web. Lấy ví dụ ở Tinh tế, chúng ta có một div để hiển thị phần đầu trang (chỗ có logo tinhte), một div nữa nằm bên dưới để hiển thị menu,

một div khác để hiển thị nội dung và div cuối cùng dành cho việc hiển thị sơ đồ trang. Các div còn có thể lồng vào nhau nữa.



*Mô hình phân tầng DIV của trang tinhte.vn*

Một thứ ngôn ngữ thứ hai được sử dụng trong thiết kế web là CSS (Cascading Style Sheet). CSS có thể nhúng như là một phần của file HTML hoặc nằm trong file riêng. Người ta sẽ dùng CSS để viết ra những dòng mã quy định kích thước các div, màu nền, ảnh nền, khung viền, nói chung là để trang trí các thành phần đồ họa trên web. Ngoài ra, CSS còn dùng để bố cục trang (layout), tức sắp xếp vị trí các div lại tùy theo mục đích của người thiết kế. Ví dụ, bạn có thể dùng CSS để bảo một div phải nằm ép sát vào cạnh phải của trình duyệt, div còn lại nằm kế bên và cả hai kéo dài đến cuối trang chứ không nằm cạnh nhau một cách vô trật tự như khi chưa xài CSS.

### **Vậy Responsive Web Design là gì?**

Responsive web design (tạm dịch thiết kế thích nghi) là một khái niệm khuyến cáo những người thiết kế và người phát triển phải đáp ứng và thích nghi được với môi trường (hay thiết bị) của người dùng về kích thước màn hình, platform, trạng thái xoay hay đứng.

Trong ngành thiết kế responsive architecture không phải là khái niệm xa lạ, bạn có bao giờ để ý đến những cánh cửa tự động đóng mở, nó sẽ tự động mở ra một không gian vừa đủ cho một người hoặc nhóm người bước vào. Một hệ thống điều chỉnh điều hòa và ánh sáng tự động điều tiết nhiệt độ và ánh sáng tùy theo số lượng người đang có trong phòng...

Responsive web design không phải là ý tưởng hoàn toàn mới, nó kế thừa và tương tự như responsive architecture, trang web khi được thiết kế phải có khả năng tự động điều chỉnh để thích nghi với nhiều nhóm người sử dụng khác nhau.

Nói ngắn gọn, RWD là một kỹ thuật thiết kế web cho phép trang web tự động điều chỉnh kích thước khung, ảnh, giao diện,... theo từng kích thước màn hình khác nhau.

## LỢI ÍCH

Như đã nói ở trên, RWD dùng để bố cục lại giao diện trang web cho tương thích với nhiều loại kích cỡ màn hình khác nhau. Hãy thử tưởng tượng xem, nếu màn hình trên điện thoại của bạn đã hẹp, diện tích rất hạn chế rồi mà bạn còn phải thấy một đồng quảng cáo trên trang web thì sẽ thế nào? Nếu là mình, mình sẽ phát điên lên. May mắn là mấy anh kỹ thuật đã làm RWD nên chúng ta không cảm thấy phiền phức.

**Responsive Web Design của Tinh tế**

The image displays three versions of the 'Tinh tế' website, demonstrating its responsive design across different screen sizes:

- 320 x 480:** A narrow mobile view with a single column layout, featuring a 'Star Trek Into Darkness' article and a 'Sharp Smartphone' advertisement.
- 360 x 640:** A slightly wider mobile view, maintaining a single column layout with the same content.
- 1140 x 1280 (Xoay ngang):** A wide desktop view where the layout expands into multiple columns, showing a full navigation bar, multiple article thumbnails, and a large 'Sharp Smartphone' advertisement.

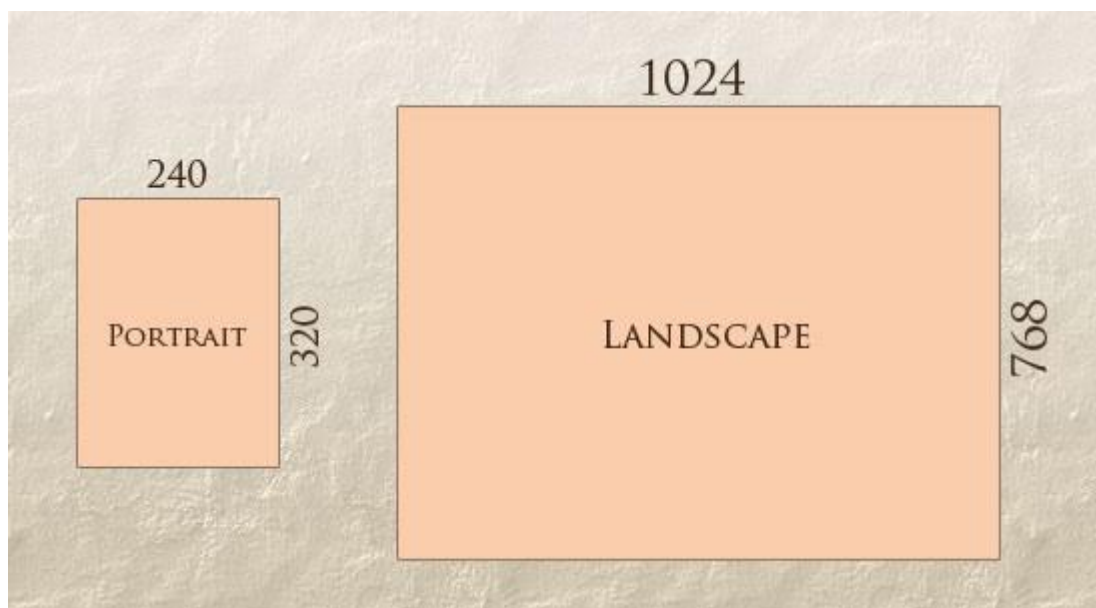
*Tính tế sẽ phản hồi lại với các kích thước màn hình/trình duyệt khác nhau bằng những bố cục khác nhau. Hai cái bên trái là giao diện mobile, cái bên phải là giao diện chuẩn*

Hoặc bạn cũng có thể nghĩ đến tình huống như thế này: giờ đây các trình duyệt web di động đều đã hỗ trợ HTML và CSS đầy đủ chứ không còn cùi bắp như xưa, do đó toàn bộ trang web có thể hiển thị một cách đầy đủ trên màn hình di động. Tuy nhiên, nếu không áp dụng RWD, trang web khi xem trên thiết bị di động sẽ trở nên nhỏ xíu, bạn bắt buộc phải dùng kính lúp, còn không thì phải zoom vào và kéo qua lại, lên xuống liên tục để đọc nội dung. Đây là trải nghiệm tiêu cực và nó khiến bạn nghĩ xấu về website, và điều tất nhiên là bạn chẳng thèm quay lại web đó nữa.

Nói tóm lại, **RWD là một xu hướng thiết kế hoàn toàn có lợi bởi nó đảm bảo bạn sẽ luôn luôn có những trải nghiệm tốt nhất, đẹp nhất khi xem trang web dù bạn có đang dùng thiết bị nào đi nữa. Nó giúp nhà lập trình web tận dụng tối đa không gian để trình diễn những nội dung cho chúng ta xem theo cách thoải mái và thích thú nhất có thể.**

**Chi tiết hơn nữa, hãy cũng điểm qua một vài tính năng hết sức linh động của RWD:**

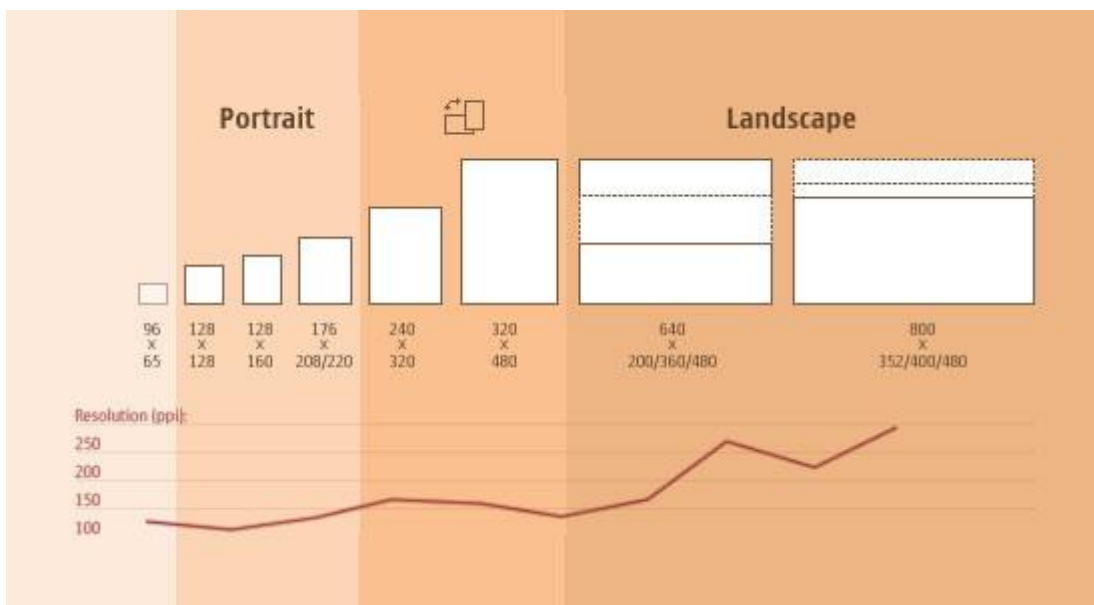
**Độ phân giải màn hình**  
Càng nhiều thiết bị, càng nhiều độ phân giải khác nhau, sự phổ biến của những thiết bị như iPhone, iPad và các smartphone có thể chuyển từ dạng xem ngang và đứng một cách nhanh chóng.



Thêm nữa khi thiết kế cho cả màn hình nằm ngang và màn hình đứng, chúng ta phải tính đến hàng trăm kích thước màn hình khác nhau. Ta có thể nhóm một số kích thước



lại với nhau và thiết kế cho từng nhóm một, và thiết kế riêng cho từng nhóm này nếu cần thiết. Bên cạnh đó bạn cũng phải biết rằng khá nhiều người dùng không bao giờ maximize trình duyệt của mình.



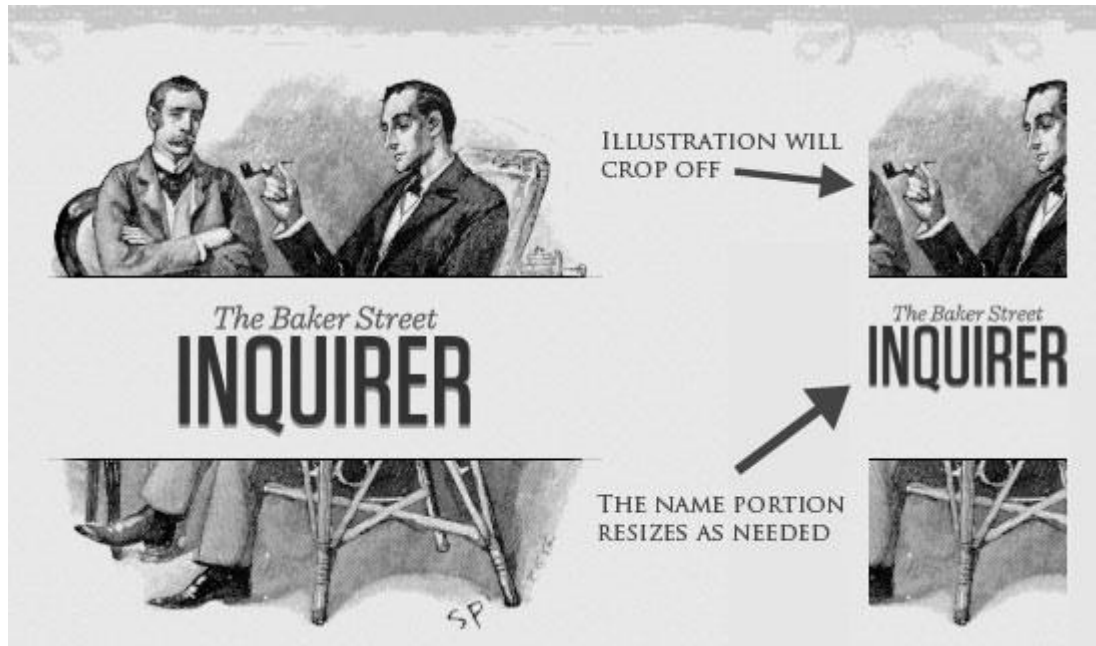
## Flexible

## layout

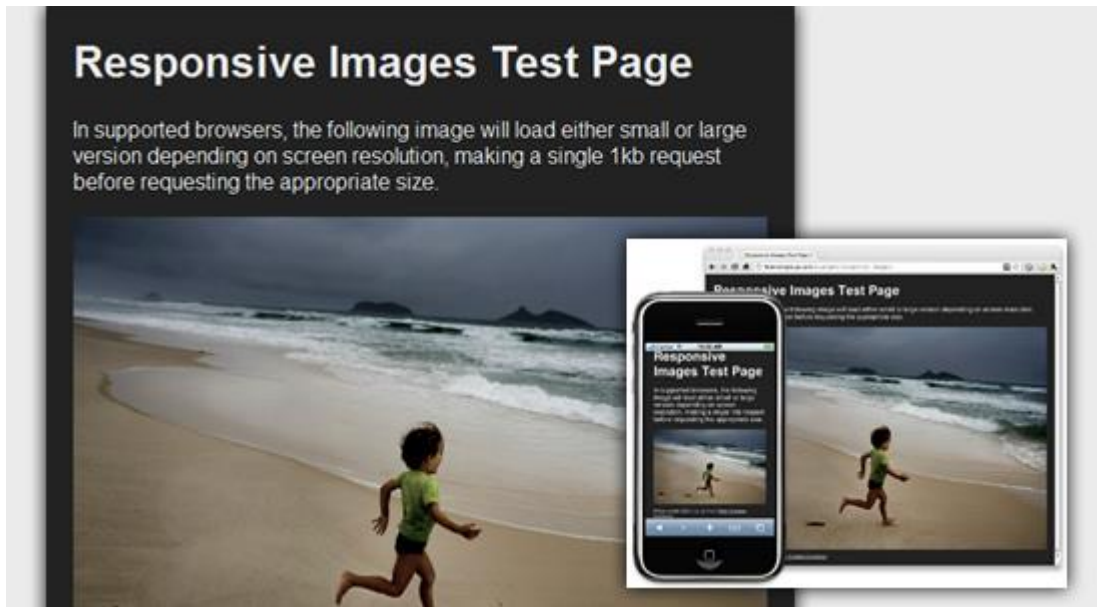
Một vài năm trước, khi flexible layout còn là thứ gì đó xa xỉ đối với website. Thứ duy nhất flexible trong thiết kế là số lượng cột và text. Ảnh có thể dễ dàng làm “vỡ” cấu trúc website. Việc thiết kế trên một khoảng kích thước tính bằng pixel khiến người thiết kế lúng túng khi trang web chuyển giữa các kích thước khung hình khác nhau.



Hình ảnh cần phải được tự động điều chỉnh và không được phép làm vỡ cấu trúc website, vì ta không làm một sản phẩm in ấn. Khi kích thước ảnh không bị fix ở một giá trị nào đó, nó mang lại nhiều lợi ích hơn mong đợi, ý tưởng tuyệt vời cho những thiết bị có thể xoay ngang và đứng. Trang web trên được thiết kế bằng cách sử dụng markup một cách thông minh, fluid grid, fluid image.



**Filament Group's Responsive Images**  
 Kỹ thuật này được giới thiệu bởi Filament Group, theo đó thay vì resize lại hình ảnh, sẽ load hẳn một ảnh khác khi cần, cách này có thể tiết kiệm được việc load ảnh lớn ko cần thiết với các thiết bị nhỏ.

**Media**

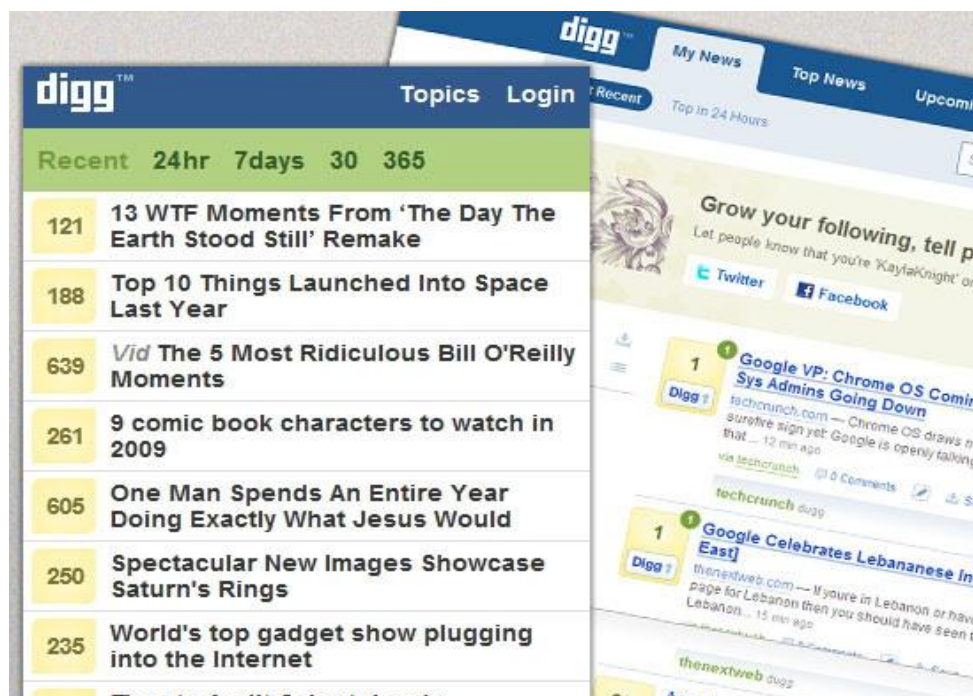
CSS 3 vẫn hỗ trợ đầy đủ media type từ CSS2.1 như screen, print, handheld, và đã bổ sung khá nhiều khai báo mới như max-width, device-width, orientation, color. Mai mắn thay những thiết bị chạy android, ipad đều không sử dụng trình duyệt IE!!! nên bạn an tâm khi sử dụng những thuộc tính css 3.

**Queries**



## Ẩn hoặc hiển thị nội dung

Rõ ràng việc chỉ hiển thị những nội dung thật sự cần thiết trên những thiết bị có kích thước nhỏ là cần thiết, kinh nghiệm cho thấy trên các thiết bị di động bạn chỉ nên có một thanh navigation đơn giản, nhấn mạnh nội dung, xem dạng list nhiều dòng tốt hơn là chia thành nhiều cột



## Touchscreen vs Cursors

Màn hình cảm ứng ngày càng phổ biến. Những thiết bị có màn hình cảm ứng thường có kích thước nhỏ, rõ ràng bạn phải thấy được rằng việc tương tác qua ngón tay cần một khoảng không gian lớn hơn như khi dùng chuột, màn hình cảm ứng cũng chẳng hề có trạng thái hover khi gì bạn chỉ có chạm và chạm



Không những thế, trong vài năm trở lại đây, thị trường di động trở thành một thị trường đầy tiềm năng và được nhiều hãng công nghệ lớn quan tâm. Nó dẫn đến thị phần bị chia nhỏ bởi nhiều hãng phần mềm khác nhau: Apple iOS, Google Android, Microsoft Windows Phone, Samsung Tizen,... Sự cạnh tranh này tuy đem lại nhiều lợi ích cho người tiêu dùng như nhiều tính năng mới được phát triển, giá thành điện thoại giảm,... và cả lập trình viên cũng được hỗ trợ nhiều hơn bởi các công ty này cần một hệ sinh thái ứng dụng cho nền tảng của họ. Nhưng đối với các lập trình viên, đặc biệt là lập trình viên các ứng dụng đám mây, thiết kế một ứng dụng đa nền tảng là hết sức khó khăn bởi mỗi hệ điều hành lại sử dụng một ngôn ngữ riêng, một trình biên dịch riêng,... Và thực sự, RWD cùng các công nghệ web như HTML5 đã xóa bỏ rào cản này. Cũng giống như trên máy tính, chúng ta có thể truy cập Facebook, Google trên bất kỳ hệ điều hành nào mà ta muốn, chỉ cần chúng ta có một trình duyệt web được cài sẵn trên máy tính của mình. **Và như vậy nhờ có RWD, ta có thể sử dụng web như một nền tảng “Code một lần và chạy ở bất cứ đâu”, tương tự như một số ngôn ngữ như Java, thậm chí còn mạnh mẽ hơn thế và nó đồng nghĩa với tiết kiệm thời gian, chi phí cho người lập trình, còn đối với người tiêu dùng, chúng ta có thể tự do lựa chọn hệ điều hành mình yêu thích, mà không cần phải quan tâm về vấn đề ứng dụng.**

# CÁCH THỨC HOẠT ĐỘNG

## Nhận biết các loại thiết bị khác nhau và áp dụng RWD

Để nhận biết thiết bị nào đang truy cập vào trang web, người ta có rất nhiều cách. Thứ nhất, mỗi trình duyệt đều có một chuỗi kí tự cũ thể để xác định tên, bộ nguồn dựng trang của chính nó. Các website được phép đọc chuỗi này để biết được rằng người dùng đang dùng trình duyệt nào, thậm chí là thiết bị nào, tên gì để vào web. Ví dụ, UserAgent của **IE 10.0** là

Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; WOW64; Trident/6.0)

còn của **Safari trên iPad** là

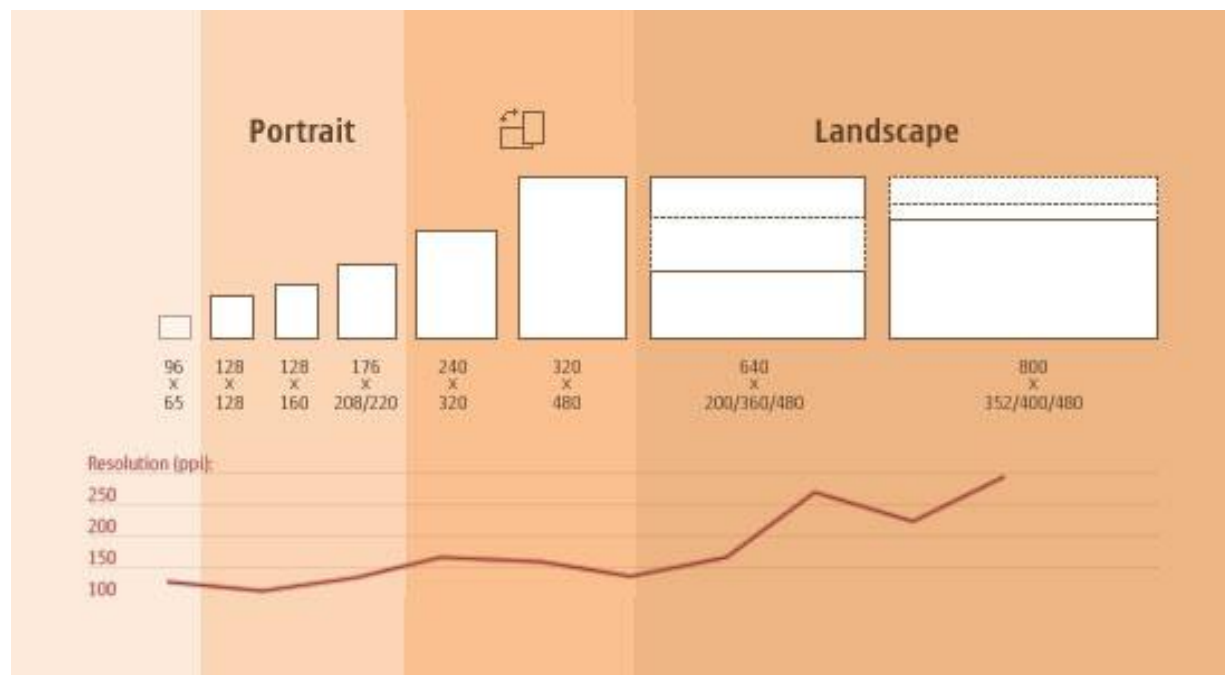
*Mozilla/5.0 (iPad; CPU OS 6\_0 like Mac OS X) AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0 Mobile/10A5355d Safari/8536.25.*

Khi biết được máy nào đang truy cập website, người ta có thể dễ dàng xác định kích cỡ màn hình và nhiều thông tin khác liên quan đến thiết bị, từ đó dùng mã tĩnh chỉnh lại bố cục web ngay khi người dùng mới truy cập vào. Khi bạn truy cập vào một số trang web nhất định, nó cũng dùng UserAgent để biết bạn đang dùng máy nào và đưa ra đề xuất tương ứng (ví dụ, vào <https://www.google.com/intl/en/chrome/browser/> bằng máy Mac thì nút download là dành cho Mac, vào bằng máy Windows thì nút download lại cho phép tải file exe của Windows).

Ngoài ra, việc nhận biết thiết bị để RWD còn có thể được tích hợp vào ngay trong file CSS và đây là cách mà người ta thường dùng. Nhà thiết kế web có thể đặt ra các "luật", ví dụ như nếu thiết bị truy cập vào web có độ phân giải chiều ngang thấp hơn 700px thì không tải thành phần A, sắp xếp lại thành phần B và C. Còn nếu thiết bị có màn hình từ 700px trở lên, thành phần A sẽ xuất hiện, B và C giữ nguyên vị trí ban đầu. Ngoài ra lập trình viên còn có kết hợp thêm một số yếu tố khác như hướng xoay của màn hình, tỉ lệ dài:ngang nữa.

Bạn hãy thử lấy điện thoại của mình vào Tinh tế xem, phần bài viết nổi bật, banner quảng cáo ở trên đầu và loạt quảng cáo ở bên tay phải sẽ biến mất, nội dung vừa khít với màn hình của bạn. Còn khi chuyển qua dùng PC hay máy tính bảng, tự nhiên những thành phần đó lại xuất hiện ra. Một trong những "luật" mà các anh kỹ thuật của Tinh tế đặt ra đó là nếu chiều dài màn hình nhỏ hơn 700 pixel thì hiển thị giao diện mobile, còn trên 700 pixel thì hiển thị kiểu đầy đủ. Ngoài ra Tinh tế còn kết hợp thêm một số

yếu tố khác để bố cục nên khi bạn dùng một chiếc smartphone Full-HD (1080 x 1920) duyệt thì vẫn thấy giao diện mobile của Tinh tế.



Một mẫu sơ đồ phân chia bố cục RWD

## HIỆU NĂNG

Năm ngoái, Guy Podjarny, trưởng nhóm kiến trúc sản phẩm của công ty Akamai (một đơn vị chuyên nghiên cứu và tối ưu hóa tốc độ cho các giải pháp điện toán) đã thử nghiệm 347 trang web RWD được trình diễn trên <http://mediaqueri.es> bằng cách dùng Chrome trên nhiều thiết bị khác nhau, cộng với công cụ WebPageTest. Podjarny nhận thấy rằng **kích thước tải về cũng như tốc độ load** của các trang web này **không khác mấy** khi sử dụng trên nhiều thiết bị khác nhau và màn hình với độ phân giải khác nhau. Sau đó Podjarny đã trình bày phát hiện của mình tại một hội nghị chuyên về thiết kế.

Vậy thử nghiệm trên có ý nghĩa gì? Nó cho thấy rằng mặc dù trang web đã phản hồi lại với kích thước màn hình bằng cách thu gọn nội dung, ẩn bớt những thành phần không cần thiết, thu nhỏ cỡ ảnh... nhưng điều đó không đồng nghĩa với chuyện web sẽ tải nhanh hơn. Nó cũng không đảm bảo rằng dung lượng tải về sẽ nhỏ hơn, ít chiếm băng thông hơn. Việc tối ưu hóa này hoàn toàn nằm trong tay lập trình viên và nhà thiết kế web.

Kích thước màn hình nhỏ hơn, ít nội dung hơn, nhưng...

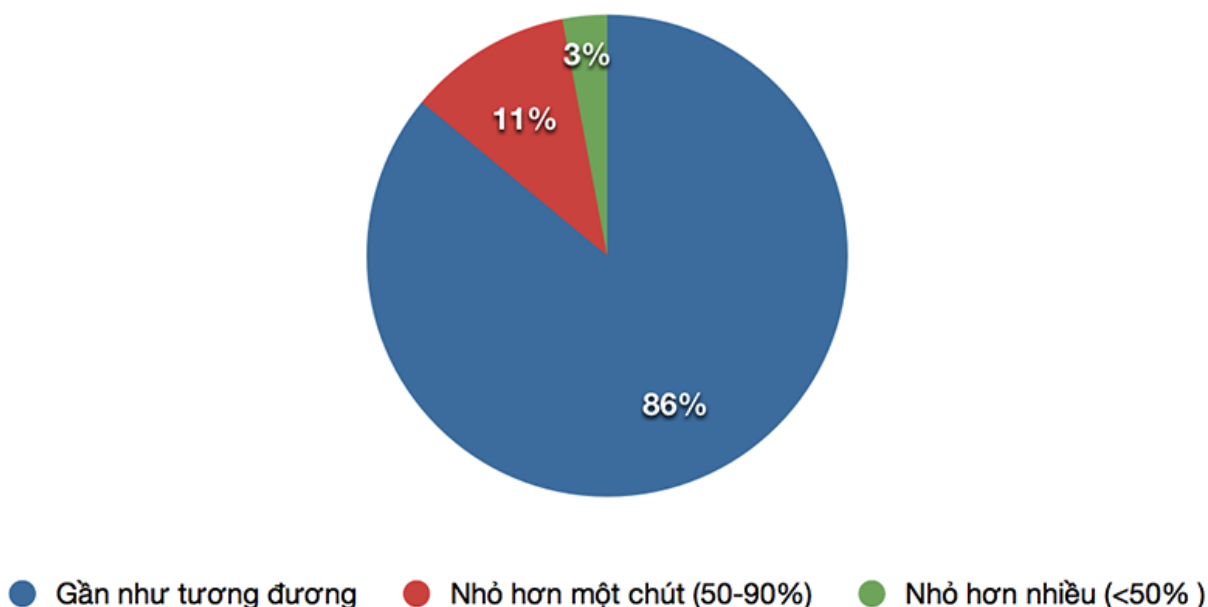


Tốc độ	Số HTTP request	Dung lượng
4.349s	38	226 KB

Tốc độ	Số HTTP request	Dung lượng
4.396s	39	240 KB

**Ghi chú:** HTTP Request là những yêu cầu mà trình duyệt gửi đến máy chủ để tải hoặc làm một thứ gì đó. Máy chủ sẽ trả lời (response) lại tương ứng với request. Ví dụ: máy tính của bạn request để tải một trang web, server sẽ trả lời lại: "Đây, đây là file mà anh cần". Ngoài ra còn có request để tải ảnh, tải file XML,... Có thể thấy ở kết quả trên, số HTTP khác nhau giữa hai loại giao diện chỉ là 1 request, không đáng kể.

Còn bên dưới là biểu đồ so sánh về kích thước trang của các web RWD trên hai loại độ phân giải do Akamai thực hiện. Đến 86% trang web khi tải giữa hai loại màn hình không cho thấy sự khác biệt rõ rệt về dung lượng trang, tức là chúng ta chỉ tiết kiệm được một khoảng dung lượng không đáng kể khi duyệt web bằng mobile.

**Kích thước trang, so sánh giữa màn hình 320 x 480 với 1600 x 1200**

Có thể bạn sẽ nói rằng "Trời ơi lo gì, bây giờ kết nối Internet trên di động đã nhanh lắm rồi!". Đúng là kết nối 3G, 4G ngày càng nhanh hơn, nhưng nó vẫn còn ở khoảng cách rất xa so với mạng Internet băng thông rộng. Hiện tại, việc tải một trang web ở trên máy tính vẫn nhanh hơn nhiều so với thiết bị di động, nhất là các trang có nhiều hình ảnh. Theo một nghiên cứu của Google, người dùng "mong đợi tốc độ tải web trên di động cũng nhanh như khi tải trên máy tính", và đó là điều mà lập trình viên phải nghĩ đến khi làm RWD.

Hoặc khi ứng dụng HTML5 và RWD để thiết kế ứng dụng offline cho mobile, dù không còn lệ thuộc vào tốc độ internet, tuy nhiên ứng dụng vẫn chưa đạt được tốc độ như ứng dụng viết thuần bằng ngôn ngữ của máy (native apps). Một ví dụ điển hình là ứng dụng Facebook cho Android, ban đầu ứng dụng này được viết bằng HTML5 tuy nhiên do hiệu năng kém nên Facebook đang dần chuyển sang native code của Android là Java.

Tuy nhiên, có thể thấy RWD yêu cầu khi người sử dụng truy cập trang web, dù lượng dữ liệu ít hay nhỏ, thì lượng code tải về vẫn như nhau, tuy nhiên như đã nêu ở trên, RWD cho phép điều khiển cho trình duyệt load các phần hình ảnh khác nhau dựa trên kích cỡ và độ phân giải màn hình vì vậy nếu một trang web có lượng hình ảnh lớn, thì lúc này RWD lại tỏ rõ ưu điểm của mình về mặt hiệu năng.



## ĐỘ PHỨC TẠP

Thật ra trước đây các lập trình viên cũng có nghĩ đến một biện pháp khác khi mà RWD chưa phổ biến, đó là xây dựng một phiên bản di động dành cho web (ví dụ: m.tinhte.vn). Cách này cũng tốt, tác dụng gần như tương đương với RWD. Tuy nhiên, nó là một trang tập tin HTML riêng, một file CSS riêng được viết riêng cho thiết bị di động, hình ảnh cũng được thiết lập với kích thước nhỏ hơn. Và mặc dù tốn công hơn nhưng các trang web m. đơn giản hơn việc tích hợp RWD vào chung với trang web chính. Một số cuộc thảo luận trên mạng cũng có nói rằng ngay cả một website tối ưu hóa cực kì tốt với RWD cũng tải chưa nhanh bằng trang web mobile riêng.

Ngoài ra, một số dịch vụ online cũng có hỗ trợ chuyển RSS thành một trang web riêng biệt. Lập trình viên có thể tận dụng điều này để thiết kế web cho thiết bị di động mà không phải đầu tư quá nhiều công sức. Họ chỉ cần làm cho RSS của mình đầy đủ nhất có thể là xong. Người dùng truy cập từ các smartphone, tablet sẽ không thấy giao diện chính mà chỉ thấy các dòng cập nhật mới nhất, tin tức mới nhất. Như vậy cũng đã đủ đối với một số trang web rồi.

Responsive web design thì ngược lại, nó vốn dĩ là phức tạp hơn bởi nhà thiết kế web đang cố gắng nhiều trải nghiệm xem khác nhau chứ không nhắm đến một loại thiết bị cụ thể nào cả. Điều đó có nghĩa là trình duyệt trên máy mobile phải đảm đương một file HTML lớn, một tập tin CSS cũng lớn không kém. Nếu không được tích hợp tốt, RWD có thể làm cho việc duyệt web di động trở nên chậm chạp hơn mặc dù bố cục rất tốt.

# ỨNG DỤNG

Thực sự, RWD nghe thoáng qua có vẻ khá phức tạp, tuy nhiên hiện nay đã có khá nhiều công cụ hỗ trợ cho việc thiết kế & ứng dụng RWD, dưới đây là một số công cụ mà bạn nên điểm qua:

## FRAMEWORK:

- Bootstrap (<http://getbootstrap.com/>)
- Foundation (<http://foundation.zurb.com/>)

## EDITOR:

- Brackets (<http://brackets.io/>), miễn phí
- Sublime Text (<http://www.sublimetext.com/>), dùng thử vô thời hạn

## TOOLS:

- Adobe Edge (<http://html.adobe.com/edge/>)

Đặc biệt, nếu bạn muốn sử dụng RWD, HTML để thiết kế ứng dụng cho các hệ điều hành di động như Android, iOS,... thì dưới đây là một số công cụ hết sức hữu ích:

- Marmalade (<http://www.madewithmarmalade.com/>)
- Adobe PhoneGap

# KẾT

Có thể nói Responsive Web Design là một công nghệ hết sức đáng quan tâm, và nếu trong tương lai vấn đề hiệu năng được cải thiện, RWD cùng với HTML5 sẽ dần thay thế cách lập trình truyền thống. Tuy chuyên đề chỉ mới giới thiệu sơ lược về RWD mà chưa đi sâu hướng dẫn nhưng qua đó mong các bạn sẽ có được cái nhìn tổng quan hơn về công nghệ web và có định hướng trong việc lựa chọn ngôn ngữ cũng như môi trường lập trình phù hợp với mình.

Thân mến,

Quang

Nguồn tham khảo:

**Responsive Web Design là gì và nó giúp ích như thế nào cho việc duyệt web trên thiết bị di động**

<http://www.tinhte.vn/threads/responsive-web-design-la-gi-va-no-giup-ich-nhu-the-nao-cho-viec-duyet-web-tren-thiet-bi-di-dong.2101375/>

Sixdivision - dịch bởi Duy Luân

**Khái niệm và ứng dụng của Responsive Web Design**

<http://www.adcvietnam.net/cong-nghe/khai-niem-va-ung-dung-cua-responsive-web-design.htm>

ADCVietnam sưu tầm

## MỤC LỤC

<b>KHÁI NIỆM .....</b>	<b>1</b>
Người ta bố cục trang web như thế nào?.....	1
<b>LỢI ÍCH .....</b>	<b>3</b>
<b>CÁCH THỨC HOẠT ĐỘNG .....</b>	<b>10</b>
Nhận biết các loại thiết bị khác nhau và áp dụng RWD .....	10
<b>HIỆU NĂNG .....</b>	<b>11</b>
<b>ĐỘ PHỨC TẠP .....</b>	<b>14</b>
<b>ỨNG DỤNG .....</b>	<b>15</b>
<b>KẾT.....</b>	<b>16</b>