**Blocks**  
Blocks, gọi nôm na là các khối, là cách mà Magento phân biệt các mảng chức năng trong hệ thống và tạo ra phong cách quản lý kiểu module để quản lý các chức năng này cả về góc độ giao diện trực quan lẫn tính năng.

**Block - Những thuộc tính có thể có của một Block Magento**

Khai báo 1 Block trong file layout .xml của Magento:  
<block type="..." name="..." as="..." template="...">  
         <action> </action>  
</block>

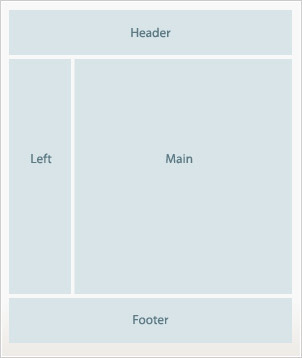
* **type:** loại Block, có thể là Block trong Magento, hay block trong một module nhất định
* **name:** Tên tham chiếu đến Block, tên này phải duy nhất trên toàn website, các block khác referrent ( Tham chiếu ) đến nó thông qua tên.
* **as:** Tên mật danh của Block hay Block con của nó, tên này được dùng để gọi Block bằng phương thức getChildHTML(ten\_mat\_danh) trong file Template tương ứng với Block.
* **template:** file có phần mở rộng .phtml, quy định nội dung bên trong của block, trong template có thể gọi đến một Block khác ( Block con hay bất kỳ Block nào đã được khai báo với Magento)
* **before (and) after:** đây là hai cách giúp sắp xếp vị trí của một Block nội dung trong một Block cấu trúc gồm nhiều Block nội dung. Sắp xếp Block lên vị trí cao nhất hay thấp nhất trong một Block cấu trúc.
* **action – <action> :**dùng để tác động lên các thay đổi trong một trang, một Block, và nhiều chức năng khác nữa.

Có 2 loại blocks và chúng phối hợp với nhau tạo ra giao diện trực quan mà bạn thấy trong Magento:

**Có 2 loại Block cơ bản trong Magento theo chức năng:**  
1. Structure Block (Block cấu trúc)

- Hiểu nôm na cho dễ hình, bạn có thể xem block cấu trúc như bộ khung cho hầu hết các trang trong website Magento. Mỗi Block cấu trúc này sẽ chứa nhiều Block con bên trong nó, gọi là Block nội dung.

- Một Block cấu trúc sẽ không là Block con của một Block nào khác, ngoài trừ Block root.

[](https://1.bp.blogspot.com/-mvc6Wtx_z2E/U_jGjXZEIDI/AAAAAAAAEA4/Y0MsA8lrh0M/s1600/design_three.png)

2. Content Block ( Block nội dung )

- Hiểu đơn giản, Block nội dung sẽ là con của một Block cấu trúc nào đó, hay là con của một Block nội dung khác. Đến lược nó có thể là cha của Block nội dung khác nữa, cứ thế cho đến vô tận ( trên lý thuyết ).

- Vậy một trang trong Magento được tạo ra là kết quả của:

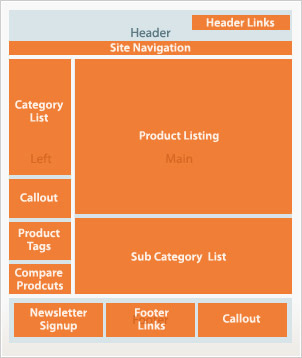
+ Số lượng Block cấu trúc được hiển thị hay ẩn đi

+ Vị trí của các Block nội dung bên trong các Block cấu trúc nào mà nó cầu hiển thị

+ Sự sắp xếp vị trí của các Block nội dung trong cùng 1 Block cấu trúc.

+ Sự sắp xếp vị trí các Block nội dung con bên trong một Block nội dung khác.

+ Ẩn hay hiện của một Block nội dung.

[](https://2.bp.blogspot.com/-YRJRmVq2OlY/U_jGuPKMHpI/AAAAAAAAEBA/E0UVb4ALjXM/s1600/design_four.png)

**Nếu phân cấp Block theo Class ( lớp ) mà Block sử dụng, hay Block Type, thì trong Magento ta có các loại Block sau:**

**core/template:**

Khối này cho phép mẫu được xác định bởi thuộc tính khuôn mẫu của nó. Phần lớn khối được xác định trong cách bố trí là loại hoặc phụ của lõi / mẫu.

**page/html:**

Đây là một loại phụ của lõi / mẫu và xác định khối gốc. Tất cả các khối khác là khối con của khối này.

**page/html\_head:**

Xác định phần đầu HTML của trang có chứa các phần tử để bao gồm JavaScript, CSS, vv

**page/html\_header:**

Xác định phần tiêu đề của trang có chứa biểu trưng trang web, các liên kết hàng đầu, v.v.

**page/template\_links:**

Khối này được sử dụng để tạo một danh sách liên kết. Liên kết có thể nhìn thấy trong phần chân trang và phần đầu trang sử dụng loại khối này.

**core/text\_list:**

Các Block như content, left, right đều là core/text\_list. Khi Block này được render, tất cả các block con của nó ( bên trong trực tiếp hay <reference> đến nó ) cũng được render ra tự động mà không cần gọi method  getChildHtml().  
**Ví dụ: tạo một Block core/text\_list tên "postscript" rỗng trong tất cả các trang website**  
<reference name="root">  
 <block type="core/text\_list" name="postscript" as="postscript" translate="label">  
 <label>Page Postscript</label>  
 </block>  
</reference>  
Tại một trang cms page <cms\_index\_index> ta tham chiếu đến Block core/text\_list "postscript" ở trên, và thêm vào đó 1 Static Block tên "block\_home\_postscript". Và một Block page/html\_wrapper bao ngoài Static Block này dược render ra thành <div class="block\_home\_postscript std"></div>  
<cms\_index\_index>  
  <reference name="postscript">  
 <block type="page/html\_wrapper" name="i\_block\_home\_postscript">  
 <action method="setElementClass"><value>block\_home\_postscript std</value></action>  
 <block type="cms/block" name="block\_home\_postscript">  
 <action method="setBlockId"><block\_id>block\_home\_postscript</block\_id></action>  
 </block>  
 </block>  
  </reference>   
  
</cms\_index\_index>

**page/html\_wrapper:**  
Block này sẽ tạo 1 <div></div> (default) bao lấy các Block con được render bên trong nó. Có thể dùng Tag name khác ngoài <div> tag thông qua <action> setHtmlTagName

**Ví dụ: thực hiện code này trong file  layout\_name.xml**  
<block type="page/html\_wrapper" name="i\_block\_cms\_page\_top" before="-">  
 <action method="setElementClass"><value>block\_cms\_page\_top std</value></action>  
 <block type="cms/block" name="block\_cms\_page\_top">  
 <action method="setBlockId"><block\_id>block\_cms\_page\_top</block\_id></action>  
 </block>  
</block>  
Kết quả sẽ tạo <div class="block\_cms\_page\_top std"> </div> bao lấy Block name="block\_cms\_page\_top" bên trong

**page/html\_breadcrumbs:**

Khối này định nghĩa các breadcrumbs trên trang.

**page/html\_footer:**  
Xác định vùng footer của trang chứa liên kết chân trang, thông báo bản quyền vv

**core/messages:**

Khối này hiện thông báo lỗi / thành công / thông báo.  
  
**page/switch:**  
Khối này có thể được sử dụng cho việc thay đổi ngôn ngữ hoặc người dùng

Đây là danh sách các loại khối thông dụng nhất. Có rất nhiều loại khối khác được sử dụng trong việc triển khai chủ đề hiện đại.