**COMMANDS TRONG PLUGINS**

* Ở mỗi plugins thông thường đều có commands (lệnh) để người dùng có thể truy cập phần bên trong của plugins.
* Để tạo commands riêng cho plugins cũng giống như tạo event, có 2 kiểu:
  + Làm trực tiếp trong class chính:
  + Làm ở bên ngoài class chính như là một class riêng biệt:
* Đối với trường hợp làm trực tiếp trong class chính:
  + Tạo một phương thức ghi đè (Override) onCommand(), có 4 tham số: CommandSender, Command, String, String[].

|  |
| --- |
| @Override  Public boolean onCommand(CommandSender sender, Command command, String label, String[] args){  //Đoạn code thực thi khi mà người chơi thực thi câu lệnh  return true;  } |

* + Giải thích các tham số:
    - CommandSender: là class để nhận diện ra người nào đã thực thi câu lệnh hoặc thứ gì đó đã bắt phải thực thi câu lệnh.
    - Command: là nội dung thực thi câu lệnh
    - String : là nội dung của câu lệnh
    - String[] : là nội dung của câu lệnh nhưng nó có nhiều options khác nhau, nhiều tham số khác nhau của một câu lệnh.
  + Ví dụ, đây là một câu lệnh khi người dùng thực thi câu lệnh này, người chơi sẽ chết :

|  |
| --- |
| @Override  Public boolean onCommand(CommandSender sender, Command command, String label, String[] args){  If(command.getName().equalsIgnoreCase(“die”) {  If(sender instanceof Player) {  Player p = (Player) sender;  p.setHealth(0);  p.sendMessage(ChatColor.RED + “You have chosen to die. We miss you!”);  }  }  Return true;  } |

* + Giải thích câu lệnh:
    - Nếu câu lệnh là ‘die’ nó sẽ tiếp túc kiểm tra nếu sender (là người thực thi câu lệnh) này là người chơi thì tiến hành set máu bằng 0 (chết) và thông báo tin nhắn cho người chơi đấy.
    - Ngoài ra việc sender khác (không phải người chơi) có thể thực thi dòng lệnh này, có thể là ConsoleCommandSender, BlockCommandSender.
  + Ngoài ra để commands hoạt động thì chúng ta phải đăng ký lệnh ở rong file “plugin.yml”, tạo một documents về commands :

|  |
| --- |
| Commands :  die :  description : An easy way to kill yourself  usage: /<command> |

* Đối với trường hợp bên ngoài class chính:
  + Tương tự như bên trong class chính thì cũng tạo một phương thức onCommand() nhưng khác biệt một chút là phải implements lại CommandExecutor
  + Ví dụ, câu lệnh làm cho người chơi bất tử:

|  |
| --- |
| Public class GodCommand implements CommandExecutor {    @Override  Public boolean onCommand(CommandSender sender, Command command, String label, String[] args) {  If(sender instanceof Player) {  Player p = (Player) player;  If(p.isInvunerable()) {  p.setInvulnerable(false);  p.sendMessage(“God mode disabled!”);  } else {  p.setInvulnerable(true);  p.sendMessage(“God mode enabled!”);  }  }  }  } |

* + Khác biệt hơn nữa chúng ta phải nhận diện lệnh ở trong class chính, thông qua phương thức getCommand(“[tên command]”).setExecutor([class thuộc commands]);

|  |
| --- |
| getCommand(“god”).setExecutor(new GodCommand()); |

* + Sau khi nhận diện lệnh ở class chính xong, chúng ta phải thực hiện cho plugins nhân diện lệnh (giống như đó là câu lệnh mặc định) ở trong plugins của chúng ta và thêm lệnh ở trong file “plugin.yml”:

|  |
| --- |
| Commands:  god:  description: Become invincible  usage: /<command> |

* + Lệnh trong class GodCommand() được thực thi khi người chơi thực hiện câu lệnh “god”
  + Aliases trong “plugin.yml” ở mỗi câu lệnh dùng để ngoài câu lệnh “god” thì chúng ta có thể định nghĩa thêm các câu lệnh khác cùng công dụng với câu lệnh “god”.