### [Tetris Battle 2P - Free online Tetris game at Tetris Friends](http://www.tetrisfriends.com/games/Battle2P/game.php)

www.tetrisfriends.com/games/Battle2P/game.php

[Dịch trang này](https://translate.google.com.vn/translate?hl=vi&sl=en&u=http://www.tetrisfriends.com/games/Battle2P/game.php&prev=search)

Play Tetris Battle 2P online for FREE! Play a two minute **game** against an opponent in this browser-based authentic Tetris **game** mode.

### [30 kịch bản lệnh trò chơi bạn có thể viết trong PHP, Phần 1: Tạo 10 ...](http://www.ibm.com/developerworks/vn/library/os-php-gamescripts1/)

www.ibm.com/developerworks/vn/library/os-php-gamescripts1/

Nhưng khi chúng ta làm việc với các trò chơi may rủi (**games** of chance), chúng ta cần cái gì đó tốt hơn một chút. **PHP** cung cấp một máy sinh số ngẫu nhiên tốt ...

### [30 game scripts you can write in PHP, Part 1: Creating 10 fundamental ...](https://www.ibm.com/developerworks/library/os-php-gamescripts1/)

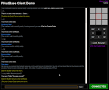
https://www.ibm.com › Learn › Open source

[Dịch trang này](https://translate.google.com.vn/translate?hl=vi&sl=en&u=https://www.ibm.com/developerworks/library/os-php-gamescripts1/&prev=search)

18 thg 11, 2008 - We won't spend a lot of time talking about **PHP** best practices or **game** design techniques. These scripts are designed to be simple to ...

### [Hình ảnh cho game php](https://www.google.com.vn/search?q=game+php&espv=2&tbm=isch&tbo=u&source=univ&sa=X&ved=0ahUKEwj2-ZizyuTSAhWBgbwKHcORCj8QsAQIMw)











[Hình ảnh khác cho game php](https://www.google.com.vn/search?q=game+php&espv=2&tbm=isch&tbo=u&source=univ&sa=X&ved=0ahUKEwj2-ZizyuTSAhWBgbwKHcORCj8Q7AkIQA)Báo cáo hình ảnh

### [HTML Game Example - W3Schools](https://www.w3schools.com/graphics/game_intro.asp)

https://www.w3schools.com/graphics/game\_intro.asp

[Dịch trang này](https://translate.google.com.vn/translate?hl=vi&sl=en&u=https://www.w3schools.com/graphics/game_intro.asp&prev=search)

... with lots of examples of how to use HTML, CSS, JavaScript, SQL, **PHP**, and XML. ... Learn how to make **games**, using nothing but HTML and JavaScript.

### [해어 윤미 - Link đây http://g2.ggdrive.cn/game.php?FQID ... - Facebook](https://www.facebook.com/permalink.php?story_fbid=312749449097394&id=1457458664543085)

https://www.facebook.com/permalink.php?story\_fbid=312749449097394&id...

Link đây http://g2.ggdrive.cn/**game**.**php**?FQID=3 anh em chơi sv 3 nhé free 500 triệu vàng sv có xúc tiến nhé hết vàng có thể đi chiến trường...

### [Game - The Kingdom of Loathing](https://www.kingdomofloathing.com/game.php)

https://www.kingdomofloathing.com/game.php

[Dịch trang này](https://translate.google.com.vn/translate?hl=vi&sl=en&u=https://www.kingdomofloathing.com/game.php&prev=search)

A Very Special Message from Jick: As the number of KoL players increases, so does the cost of hosting. I very much want to keep the **game** free, so please ...

### [Tibia - Free Massively Multiplayer Online Role-Playing Game ...](https://secure.tibia.com/mmorpg/free-multiplayer-online-role-playing-game.php)

https://secure.tibia.com/.../free-multiplayer-online-role-playing-game.p...

[Dịch trang này](https://translate.google.com.vn/translate?hl=vi&sl=en&u=https://secure.tibia.com/mmorpg/free-multiplayer-online-role-playing-game.php&prev=search)

Tibia is a free massively multiplayer online role-playing **game** (MMORPG). Join this fascinating **game**that has thousands of fans from all over the world!

### [Lưu trữ dữ liệu trong game: php javascript mysql! Cộng đồng phát ...](http://vietgamedev.net/blog/104/l%C6%B0u-tr%E1%BB%AF-d%E1%BB%AF-li%E1%BB%87u-trong-game-php-javascript-mysql/)

vietgamedev.net/blog/104/lưu-trữ-dữ-liệu-trong-game-php-javascript-mysql/

12 thg 4, 2012 - Lưu trữ dữ liệu trong **game**: **php** javascript mysql!. Trong quá trình làm game Cuộc Chiến Không Gian để trở thành game app của VietGamDev ...

### [How to make a browser game for dummies - Part 1 - PHP / HTML ...](https://www.youtube.com/watch?v=dWzzRkxrTMA)

[**▶ 37:43**](https://www.youtube.com/watch?v=dWzzRkxrTMA)

https://www.youtube.com/watch?v=dWzzRkxrTMA

30 thg 7, 2014 - Tải lên bởi Jeroen Beekman

Hey guys, Today we are going to start working on a browser **game** in the likes of Travian and Tribal Wars. We ...

## **Kết quả tìm kiếm**

### [PHP Tutorial: 2 Player Tic-Tac-Toe Game (no database required ...](https://jadendreamer.wordpress.com/2012/01/31/php-tutorial-2-player-tic-tac-toe-game-no-database-required/)

https://jadendreamer.wordpress.com/.../php-tutorial-2-player-tic-tac-toe...

[Dịch trang này](https://translate.google.com.vn/translate?hl=vi&sl=en&u=https://jadendreamer.wordpress.com/2012/01/31/php-tutorial-2-player-tic-tac-toe-game-no-database-required/&prev=search)

31 thg 1, 2012 - In the first part of my **games** tutorial series we created a **PHP** Hangman **game** and then in the second installment we converted Hangman to use ...

### [Play Game - Enercities](http://www.enercities.eu/game.php)

www.enercities.eu/game.php

[Dịch trang này](https://translate.google.com.vn/translate?hl=vi&sl=en&u=http://www.enercities.eu/game.php&prev=search)

EnerCities is a 3D **Game** and needs the Unity Web Player to work. 100% Safe; Fast download (3.1mb); Enjoy amazing 3D graphics! Unity Website ...

### [Estimation Games - Math is Fun](http://www.mathsisfun.com/numbers/estimation-game.php)

www.mathsisfun.com/numbers/estimation-game.php

[Dịch trang này](https://translate.google.com.vn/translate?hl=vi&sl=en&u=http://www.mathsisfun.com/numbers/estimation-game.php&prev=search)

Find the answer fast ... you also get points for being close! Try every **game**, and don't worry if you make mistakes, you will get better with practice.

### [How to create a Snake game in PHP (using ncurses) Code Example ...](http://code.runnable.com/UoBqWfxBADcTAAA8/how-to-create-a-snake-game-in-php-using-ncurses-)

code.runnable.com/.../how-to-create-a-snake-game-in-php-using-ncurs...

[Dịch trang này](https://translate.google.com.vn/translate?hl=vi&sl=en&u=http://code.runnable.com/UoBqWfxBADcTAAA8/how-to-create-a-snake-game-in-php-using-ncurses-&prev=search)

How to create a Snake **game** in **PHP** (using ncurses) Code Example - Runnable.

### [Game Guide - Grim Dawn - ARPG from Crate Entertainment](http://www.grimdawn.com/game.php)

www.grimdawn.com/game.php

[Dịch trang này](https://translate.google.com.vn/translate?hl=vi&sl=en&u=http://www.grimdawn.com/game.php&prev=search)

The official website and community for Grim Dawn, an action role playing **game** currently being developed by Crate Entertainment.

### [share code game php - SinhvienIT](http://sinhvienit.net/forum/tags.php?tag=share+code+game+php)

sinhvienit.net/forum/tags.php?tag=share+code+game+php

Bài viết về share code **game php**, hình ảnh về share code **game php**, video về share code **game php**, hướng dẫn về share code **game php**, những thông tin liên ...

### [Layers of Time Game - OLogy](http://www.amnh.org/ology/features/layersoftime/game.php)

www.amnh.org/ology/features/layersoftime/game.php

[Dịch trang này](https://translate.google.com.vn/translate?hl=vi&sl=en&u=http://www.amnh.org/ology/features/layersoftime/game.php&prev=search)

Loading: 100%. CONFIGURING. You have put the layers in the right order! button label. normal italic. level\_panel\_title. loading... Let's Get. Started! Loading: 100 ...

### [The ReDistricting Game](http://www.redistrictinggame.org/game.php)

www.redistrictinggame.org/game.php

[Dịch trang này](https://translate.google.com.vn/translate?hl=vi&sl=en&u=http://www.redistrictinggame.org/game.php&prev=search)

Mission 1: Fundamentals Learn the basics of redistricting in the State of Jefferson. Basic - Advanced - Learn More - Quotes. Mission 2: Partisan Gerrymander

### [www.heroesland.com/game.php](http://www.heroesland.com/game.php)

Không có mô tả cho kết quả này do robots.txt của trang web này

[Tìm hiểu thêm](https://support.google.com/webmasters/answer/156449?hl=vi)

### [game php free download - SourceForge](https://sourceforge.net/directory/?q=game+php)

https://sourceforge.net/directory/?q=game+php

[Dịch trang này](https://translate.google.com.vn/translate?hl=vi&sl=en&u=https://sourceforge.net/directory/%3Fq%3Dgame%2Bphp&prev=search)

**game php** free download. php strategy game engine This is an engine for creating a php browser game, like ikariam, travian and ogame. Whit mysql ...

content 1

# **30 kịch bản lệnh trò chơi bạn có thể viết trong PHP, Phần 1: Tạo 10 kịch bản lệnh cơ sở**

PHP là một ngôn ngữ lập trình dễ sử dụng, dễ học, có mặt rộng rãi mọi nơi. Nó cũng rất thích hợp để phát triển các kịch bản lệnh đơn giản mà bạn có thể sử dụng để giúp bạn trong tất cả các loại trò chơi. Cho dù bạn chưa hay đã từng chơi các trò chơi một mình đơn giản chỉ dùng bút và giấy, các trò chơi phức tạp, có phân vai, bên bàn với một nhóm người, hay các trò chơi trực tuyến bất kỳ loại nào, thì loạt bài viết này này vẫn sẽ có một ý nghĩa gì đó đối với bạn. Mỗi bài trong loạt bài "[30 kịch bản lệnh trò chơi bạn có thể viết trong PHP](http://www.ibm.com/developerworks/vn/views/opensource/libraryview.jsp?search_by=30+k%E1%BB%8Bch+b%E1%BA%A3n+l%E1%BB%87nh+tr%C3%B2+ch%C6%A1i+b%E1%BA%A1n+c%C3%B3+th%E1%BB%83+vi%E1%BA%BFt+trong+PHP)" sẽ trình bày 10 kịch bản lệnh trong khoảng 300 từ hoặc ít hơn (*3d10* là viết tắt từ “roll three 10-sided dice” – lăn súc sắc 10 mặt ) đủ đơn giản đối với ngay cả một nhà phát triển mới vào nghề, nhưng cũng đủ ích lợi đối với một tay chơi dày dạn. Mục đích là cung cấp cho bạn một thứ gì đó mà bạn có thể sửa đổi cho phù hợp với nhu cầu của bạn, sao cho bạn có thể gây ấn tượng cho bạn bè và người chơi bằng cách làm nổ tung máy tính xách tay của bạn trong ván tiếp theo.

[Xem thêm bài trong loạt bài này](http://www.ibm.com/developerworks/vn/views/opensource/libraryview.jsp?search_by=30+k%E1%BB%8Bch+b%E1%BA%A3n+l%E1%BB%87nh+tr%C3%B2+ch%C6%A1i+b%E1%BA%A1n+c%C3%B3+th%E1%BB%83+vi%E1%BA%BFt+trong+PHP) | [PDF:](http://www.ibm.com/developerworks/vn/library/os-php-gamescripts1/os-php-gamescripts1-pdf.pdf) (560 KB) |

[Duane O'Brien](http://www.ibm.com/developerworks/vn/library/os-php-gamescripts1/#authorN1002F), Phát triển PHP, Java, 自由职业者

31 07 2009

* expand**Mục lục**



Phát tiển và triển khai ứng dụng tiếp theo của bạn trên nền tảng đám mây IBM Bluemix.

[**Bắt đầu thử miễn phí**](https://developer.ibm.com/sso/bmregistration?lang=en_VN&ca=dwvietnam-_-bluemix-_-os-php-gamescripts1-_-sidebar)

## **Bắt đầu**

Với tư cách là người thạo trò chơi/người kể chuyện và cũng là người phát triển, tôi thường tự mình viết vài tiện ích nhỏ và kịch bản lệnh để giúp cho chạy, lập kế hoạch, và chơi các trò chơi. Đôi khi tôi cần một ý tưởng nhanh. Vào lúc khác, tôi lại cần một đống tên cho các Nhân vật Không-phải-là-Người-chơi (NPCs - Non-Player Characters). Thỉnh thoảng, tôi cần phải vùi đầu vào các con số, tính toán tỷ lệ cược, hoặc tích hợp một số trò đố chữ vào một trò chơi. Nhiều việc trong số đó trở nên dễ quản lý hơn với một ít công sức làm kịch bản lệnh sẵn từ trước.

Bài này sẽ khám phá 10 kịch bản lệnh cơ bản, có thể được sử dụng trong các loại trò chơi. Kho lưu trữ mã chứa toàn bộ nguồn cho từng kịch bản lệnh chúng ta sẽ thảo luận, và bạn có thể thấy các kịch bản lệnh này hoạt động như thế nào tại trang web [chaoticneutral](http://chaoticneutral.net/3d10).

Chúng ta sẽ điểm qua các kịch bản lệnh này khá nhanh. Các chủ đề về tìm kiếm một máy chủ chứa hoặc thiết đặt một máy chủ sẽ không được trình bày. Có nhiều công ty chủ chứa các trang web cung cấp PHP, và bộ cài đặt XAMPP rất dễ sử dụng nếu bạn muốn thiết lập máy chủ cho riêng bạn. Chúng ta sẽ không dành nhiều thời gian nói về các thói quen thực hành PHP tốt nhất hoặc các kỹ thuật thiết kế trò chơi hay nhất. Các kịch bản lệnh này được thiết kế đơn giản để dễ hiểu, sử dụng đơn giản, và nhanh chóng nắm bắt được.

[**Về đầu trang**](http://www.ibm.com/developerworks/vn/library/os-php-gamescripts1/#ibm-pcon)

## **Trò gieo súc sắc cơ sở**

Nhiều trò chơi và các hệ thống trò chơi cần đến súc sắc. Hãy bắt đầu với một cái gì đó đơn giản: gieo một quân súc sắc sáu mặt. Về bản chất, không có khác biệt giữa việc gieo một con súc sắc sáu mặt và việc nhặt ngẫu nhiên một số từ 1 đến 6. Trong PHP, đây là điều đơn giản: echo rand(1,6);.

Trong nhiều trường hợp, như thế cũng đã là ổn. Nhưng khi chúng ta làm việc với các trò chơi may rủi (games of chance), chúng ta cần cái gì đó tốt hơn một chút. PHP cung cấp một máy sinh số ngẫu nhiên tốt hơn, đó là: mt\_rand(). Không cần đi sâu vào chi tiết về sự khác biệt giữa hai cái đó, cứ an tâm khi giả thiết rằng mt\_rand là một máy sinh số ngẫu nhiên tốt hơn và nhanh hơn: echo mt\_rand(1,6);. Chúng ta về tổng thể sẽ hài lòng hơn nếu đặt những cái này vào trong một hàm.

##### **Liệt kê 1. Sử dụng hàm sinh số ngẫu nhiên mt\_rand()**

function roll () {  
 return mt\_rand(1,6);  
}  
  
echo roll();

Sau đó chúng ta có thể chuyển kiểu súc sắc chúng ta muốn dùng tới hàm này dưới dạng một tham số.

##### **Liệt kê 2. Chuyển kiểu súc sắc thành một tham số**

function roll ($sides) {  
 return mt\_rand(1,$sides);  
}  
  
echo roll(6); // roll a six-sided die  
echo roll(10); // roll a ten-sided die  
echo roll(20); // roll a twenty-sided die

Từ đây, chúng ta có thể tiếp tục gieo nhiều súc sắc một lần, trả về một mảng kết quả, hoặc gieo nhiều con súc sắc các loại khác nhau cùng một lúc, tùy vào nhu cầu của chúng ta. Nhưng kịch bản lệnh đơn giản này có thể được sử dụng cho hầu hết các nhiệm vụ.

[**Về đầu trang**](http://www.ibm.com/developerworks/vn/library/os-php-gamescripts1/#ibm-pcon)

## **Máy sinh tên ngẫu nhiên**

Nếu bạn đang chạy các trò chơi, viết các truyện, hoặc tạo ra một nhóm các nhân vật cùng một lúc, đôi khi khó tìm thấy các tên mới. Chúng ta hãy xem xét một máy sinh tên ngẫu nhiên mà bạn có thể sử dụng để giải quyết vấn đề này. Để khởi động, chúng ta hãy tạo hai mảng đơn giản — một mảng các tên riêng, một mảng các tên họ.

##### **Liệt kê 3. Hai mảng đơn giản các tên riêng và tên họ**

$male = array(  
 "William",  
 "Henry",  
 "Filbert",  
 "John",  
 "Pat",  
);  
  
$last = array(  
 "Smith",  
 "Jones",  
 "Winkler",  
 "Cooper",  
 "Cline",  
);

Sau đó chúng ta có thể chỉ cần chọn ra một phần tử ngẫu nhiên từ mỗi mảng: echo $male[array\_rand($male)] . ' ' . $last[array\_rand($last)];. Để rút ra một nhóm tên mỗi lần, chúng ta có thể chỉ cần xáo trộn các mảng và rút ra nhiều hay ít tuỳ thích.

##### **Liệt kê 4. Xáo trộn các mảng tên**

shuffle($male);  
shuffle($last);  
for ($i = 0; $i <= 3; $i++) {  
 echo $male[$i] . ' ' . $last[$i];  
}

Khi nắm được khái niệm cơ bản này, chúng ta có thể tạo ra các tệp tin văn bản để lưu giữ các tên riêng và tên họ. Nếu chúng ta đặt một tên trên mỗi dòng trong tệp tin văn bản của chúng ta, chúng ta có thể dễ dàng chia tách nội dung các tệp tin theo ký tự xuống dòng để xây dựng các mảng nguồn của chúng ta.

##### **Liệt kê 5. Tạo các tệp tin văn bản cho các tên của chúng ta**

$male = explode('\n', file\_get\_contents('names.female.txt'));  
$last = explode('\n', file\_get\_contents('names.last.txt'));

Hãy xây dựng hoặc tìm ra một số tệp tin tốt chứa các tên (một vài tệp có trong [kho lưu trữ mã](http://www.ibm.com/developerworks/vn/library/os-php-gamescripts1/#download)) và chúng ta sẽ không bao giờ muốn tìm thêm các tên nữa.

[**Về đầu trang**](http://www.ibm.com/developerworks/vn/library/os-php-gamescripts1/#ibm-pcon)

## **Máy sinh kịch bản**

Theo cùng một nguyên tắc cơ bản mà chúng ta đã sử dụng để làm máy sinh tên, chúng ta có thể làm ra cái gọi là *máy sinh kịch bản* Điều này là hữu ích trong các trò chơi nhập vai hay các tình huống khác mà chúng ta cần có một tập hợp giả ngẫu nhiên các tình huống, có thể được sử dụng để đóng vai diễn, ứng tác, viết, v.v... Một trong những trò chơi yêu thích của tôi là Paranoia, có một thứ tên là “mission blender” (N.D: máy trộn ngẫu nhiên, sinh ra các sứ mệnh (nhiệm vụ) khác nhau để luyện trí óc) trong GM Pack của nó. Mission blender có thể được sử dụng để lắp ghép thành một sứ mệnh đầy đủ dựa trên kết quả gieo súc sắc. Chúng ta hãy lắp ghép máy sinh kịch bản của chính chúng ta.

Lấy kịch bản sau đây: Bạn thức dậy thấy bị lạc trong rừng. Bạn biết bạn phải đến New York, nhưng bạn không biết lý do. Bạn nghe thấy tiếng chó sủa và âm thanh không thể nhầm lẫn của những kẻ săn tìm thù địch ngay gần đó. Bạn bị lạnh, run, và không có vũ khí. Mỗi câu trong kịch bản đó đưa ra một khía cạnh cụ thể của kịch bản:

* "Bạn thức dậy thấy bị lạc trong rừng" — Việc này tạo ra một khung cảnh (setting).
* "Bạn biết bạn phải đến New York" — Điều này mô tả một mục tiêu.
* "Bạn nghe thấy tiếng chó sủa" — Việc này đưa ra một bên đối kháng.
* "Bạn bị lạnh, run, và không có vũ khí" — Điều này làm gia tăng sự phức tạp.

Đúng như chúng ta đã tạo ra các tệp tin văn bản cho tên riêng và tên họ của chúng ta, bắt đầu bằng việc tạo các tệp tin văn bản tương ứng cho các khung cảnh, các mục tiêu, các bên đối kháng, và sự phức tạp. Các tệp tin mẫu được gộp trong kho lưu trữ mã. Một khi chúng ta có các tệp tin này, mã để sinh ra một kịch bản trông rất giống với mã để sinh ra các tên.

##### **Liệt kê 6. Sinh ra một kịch bản**

$settings = explode("\n", file\_get\_contents('scenario.settings.txt'));  
$objectives = explode("\n", file\_get\_contents('scenario.objectives.txt'));  
$antagonists = explode("\n", file\_get\_contents('scenario.antagonists.txt'));  
$complications = explode("\n", file\_get\_contents('scenario.complications.txt'));  
  
shuffle($settings);  
shuffle($objectives);  
shuffle($antagonists);  
shuffle($complications);  
  
echo $settings[0] . ' ' . $objectives[0] . ' ' . $antagonists[0] . ' '   
. $complications[0] . "<br />\n";

Chúng ta có thể thêm các yếu tố vào kịch bản của chúng ta bằng cách bổ sung các tệp tin văn bản mới, hoặc chúng ta có thể muốn gia tăng thêm nhiều sự phức tạp. Chúng ta càng thêm vào các tệp tin văn bản cơ bản, thì sau này các kịch bản của chúng ta càng thêm đa dạng.

[**Về đầu trang**](http://www.ibm.com/developerworks/vn/library/os-php-gamescripts1/#ibm-pcon)

## **Trình lập cỗ bài và trình xáo trộn bài**

Nếu bạn chơi bài và quan tâm đến việc viết bất kỳ kịch bản lệnh nào liên quan đến lá bài, chúng ta muốn ghép cùng nhau một trình lập cỗ bài với trình trộn bài lập sẵn. Để bắt đầu, chúng ta hãy lập ra một cỗ bài cơ sở gồm các lá bài tiêu chuẩn. Chúng ta cần phải lập hai mảng — một mảng để giữ các lá bài hoa (suits) và một mảng để giữ các lá bài mặt người (quân J, Q, K). Việc này sẽ cho phép có sự linh hoạt sau này nếu chúng ta muốn thêm vào các hoa hoặc kiểu lá bài mới.

##### **Liệt kê 7. Lập một cỗ bài cơ sở**

$suits = array (  
 "Spades", "Hearts", "Clubs", "Diamonds"  
);  
  
$faces = array (  
 "Two", "Three", "Four", "Five", "Six", "Seven", "Eight",  
 "Nine", "Ten", "Jack", "Queen", "King", "Ace"  
);

Sau đó lập ra một mảng cỗ bài để giữ tất cả các trị số của lá bài (card values). Chúng ta có thể làm việc này một cách đơn giản bằng cách sử dụng một cặp vòng lặp foreach.

##### **Liệt kê 8. Lập một mảng cỗ bài**

$deck = array();  
  
foreach ($suits as $suit) {  
 foreach ($faces as $face) {  
 $deck[] = array ("face"=>$face, "suit"=>$suit);  
 }  
}

Một khi chúng ta đã có một mảng cỗ bài được lập, chúng ta có thể trộn bài dễ dàng và rút ngẫu nhiên một lá bài.

##### **Liệt kê 9. Trộn cỗ bài và rút ngẫu nhiên một lá bài**

shuffle($deck);  
  
$card = array\_shift($deck);  
  
echo $card['face'] . ' of ' . $card['suit'];

Từ đây, chỉ còn một quãng đường ngắn để rút ra một tay bài gồm một bộ các lá bài hoặc lập ra một multideck shoe (N.D – tên một lối chơi bài).

[**Về đầu trang**](http://www.ibm.com/developerworks/vn/library/os-php-gamescripts1/#ibm-pcon)

## **Trình tính toán khả năng: Rút lá bài**

Do chúng ta đã lập ra cỗ bài theo cách đã làm ở trên, theo vết lá bài mặt người (face) và lá bài hoa (suit) riêng rẽ, chúng ta có thể sử dụng cỗ bài theo cách lập chương trình để làm một điều gì đó chẳng hạn như tính toán khả năng nhận được một lá bài cụ thể. Bắt đầu bằng cách rút ra hai tay bài mỗi tay năm lá bài.

##### **Liệt kê 10. Rút hai tay bài mỗi tay năm lá bài**

$hands = array(1 => array(), 2=>array());  
  
for ($i = 0; $i < 5; $i++) {  
 $hands[1][] = implode(" of ", array\_shift($deck));  
 $hands[2][] = implode(" of ", array\_shift($deck));  
}

Sau đó chúng ta có thể tìm trong cỗ bài để xem có bao nhiêu lá bài còn lại và khả năng để rút được một lá bài cụ thể là bao nhiêu. Còn lại bao nhiêu lá bài là một việc dễ dàng. Đó chỉ là việc đếm có bao nhiêu phần tử trong mảng $deck. Để tính được khả năng rút được một lá bài cụ thể, chúng ta cần một hàm duyệt qua toàn bộ cỗ bài và đánh giá các lá bài còn lại để xem chúng có khớp với yêu cầu không.

##### **Liệt kê 11. Tính toán khả năng rút được một lá bài cụ thể**

function calculate\_odds($draw, $deck) {  
 $remaining = count($deck);  
 $odds = 0;  
 foreach ($deck as $card) {  
 if ( ($draw['face'] == $card['face'] && $draw['suit'] ==   
$card['suit'] ) ||  
 ($draw['face'] == '' && $draw['suit'] == $card['suit'] ) ||  
 ($draw['face'] == $card['face'] && $draw['suit'] == '' ) ) {  
 $odds++;  
 }  
 }  
 return $odds . ' in ' $remaining;  
}

Bây giờ chúng ta có thể chọn lá bài mà chúng ta đang thử rút. Để làm việc này dễ dàng hơn, chuyển vào một mảng trông hệt như một lá bài. Chúng ta có thể tìm kiếm một lá bài cụ thể.

##### **Liệt kê 12. Tìm kiếm một lá bài cụ thể**

$draw = array('face' => 'Ace', 'suit' => 'Spades');  
  
echo implode(" of ", $draw) . ' : ' . calculate\_odds($draw, $deck);

Hoặc chúng ta có thể tìm một lá bài có một mặt người hoặc có hoa đã cho.

##### **Liệt kê 13. Tìm kiếm một lá bài có một mặt người hoặc có hoa cho trước**

$draw = array('face' => '', 'suit' => 'Spades');  
$draw = array('face' => 'Ace', 'suit' => '');

[**Về đầu trang**](http://www.ibm.com/developerworks/vn/library/os-php-gamescripts1/#ibm-pcon)

## **Trình chia bài poker đơn giản**

Bây giờ chúng ta đã có một trình lập cỗ bài và một thứ để tính ra khả năng rút được các lá bài cụ thể, chúng ta có thể ghép thành một trình chia bài thực sự đơn giản để thực hành tay bài poker (poker hands). Để phục vụ cho thí dụ này, chúng ta lập ra một trình chia bài để rút năm lá bài. Trình chia bài sẽ cung cấp năm lá bài từ cỗ bài. Bạn chỉ rõ các lá bài nào mà bạn muốn chui thông qua số hiệu lá bài, và trình chia bài sẽ thay các lá bài này bằng các lá bài mới từ cỗ bài. Chúng ta sẽ không phải bận lòng với việc đặt ra các giới hạn rút bài hay các luật chơi riêng, mặc dù bạn có thể thấy đó là một bài luyện tập cá nhân đáng làm.

Như đã trình bày trong phần trước, tạo ra và trộn một cỗ bài, sau đó tạo ra một tay bài gồm năm lá bài. Hiển thị các lá bài bằng chỉ số mảng của chúng để bạn có thể xác định được những lá bài nào trả về. Bạn phải làm điều đó bằng cách sử dụng các hộp kiểm để để chỉ các lá bài nào bạn muốn thay thế.

##### **Liệt kê 14. Sử dụng hộp kiểm để chỉ ra các lá bài bạn đang thay thế**

foreach ($hand as $index =>$card) {  
 echo "<input type='checkbox' name='card[" . $index . "]'>   
" . $card['face'] . ' of ' . $card['suit'] . "<br />";  
}

Sau đó, đánh giá mảng đầu vào array $\_POST['card'] để xem các lá bài đã được đánh dấu kiểm để thay thế.

##### **Liệt kê 15. Đánh giá đầu vào**

$i = 0;  
while ($i < 5) {  
 if (isset($\_POST['card'][$i])) {  
 $hand[$i] = array\_shift($deck);  
 }  
}

Sử dụng kịch bản lệnh này, bạn có thể thử luyện tay — chơi chữ có chủ ý — qua việc nêu rõ các cách tốt nhất để xử lý một bộ các lá bài cụ thể.

[**Về đầu trang**](http://www.ibm.com/developerworks/vn/library/os-php-gamescripts1/#ibm-pcon)

## **Người chơi trò Hangman**

Hangman về bản chất là một trò chơi đoán từ. Cho trước một từ có độ dài nhất định, chúng ta có một số lần hạn chế để đoán các chữ cái. Nếu bạn đoán đúng một chữ cái có xuất hiện trong từ, tất cả các vị trí xuất hiện của chữ đó được điền vào. Sau một số lần đoán sai định từ trước (thường là sáu), bạn sẽ bị thua. Để lắp ghép thành một trò chơi hangman thô sơ, chúng ta cần bắt đầu với một danh sách các từ. Bây giờ, chúng ta hãy tạo cho nó một mảng đơn giản.

##### **Liệt kê 16. Tạo một danh sách các từ**

$words = array (  
 "giants",  
 "triangle",  
 "particle",  
 "birdhouse",  
 "minimum",  
 "flood"  
);

Sử dụng các kỹ thuật đã trình bày trước đây, chúng ta có thể di chuyển những từ này tới một tệp tin văn bản danh sách các từ ở bên ngoài và nhập khẩu chúng vào theo ý muốn.

Khi chúng ta đã có một danh sách các từ, chúng ta cần chọn ngẫu nhiên một từ, hiển thị khoảng trống thay cho mỗi chữ, và bắt đầu đoán. Chúng ta cần theo vết các lần đoán đúng và sai từ lần đoán này đến lần khác. Chúng ta sẽ làm điều này với chi phí rẻ bằng cách chỉ cần sắp tuần tự mảng các lần đoán và truyền qua mỗi lần đoán. Nếu ta muốn giữ không cho người chơi gian lận bằng cách xem thẳng trang mã nguồn, thì phải làm một cái gì đó đảm bảo an ninh hơn một chút.

Lập ra các mảng để giữ các chữ cái và các lần đoán đúng/sai của chúng ta. Với lần đoán đúng, chúng ta sẽ điền vào một mảng các chữ như các khóa và dấu chấm câu như là các giá trị.

##### **Liệt kê 17. Lập các mảng để lưu giữ các chữ cái và các lần đoán**

$letters = array('a','b','c','d','e','f','g','h','i','j','k','l','m','n','o',  
'p','q','r','s','t','u','v','w','x','y','z');  
$right = array\_fill\_keys($letters, '.');  
$wrong = array();

Bây giờ chúng ta cần một ít mã để đánh giá các lần đoán và hiển thị từ theo tiến triển của trò chơi đoán chữ này.

##### **Liệt kê 18. Đánh giá các lần đoán và hiển thị sự tiến triển**

if (stristr($word, $guess)) {  
 $show = '';  
 $right[$guess] = $guess;  
 $wordletters = str\_split($word);  
 foreach ($wordletters as $letter) {  
 $show .= $right[$letter];  
 }  
   
} else {  
 $show = '';  
 $wrong[$guess] = $guess;  
 if (count($wrong) == 6) {  
 $show = $word;  
 } else {  
 foreach ($wordletters as $letter) {  
 $show .= $right[$letter];  
 }  
 }  
}

Trong [kho lưu trữ mã nguồn](http://www.ibm.com/developerworks/vn/library/os-php-gamescripts1/#download), chúng ta có thể thấy cách thức xếp tuần tự các mảng của mỗi lần đoán và chuyển chúng từ lần đoán này đến lần khác.

[**Về đầu trang**](http://www.ibm.com/developerworks/vn/library/os-php-gamescripts1/#ibm-pcon)

## **Trình trợ giúp trò chơi ô chữ**

Tôi biết nó là xấu, nhưng đôi lúc khi bạn đang chơi trò đoán ô chữ, bạn bỗng vướng mắc khi cố gắng tìm một từ có năm chữ cái mà bắt đầu bằng chữ C và kết thúc bằng chữ T. Bằng cách sử dụng cùng một danh sách các từ chúng ta đã lập cho Hangman, có thể dễ dàng tìm kiếm những từ khóa vừa khớp với một mẫu nhất định. Trước tiên, hãy thiết lập một cách để chuyển tải các từ. Để làm cho việc này dễ dàng, hãy thay các chữ thiếu bằng các dấu chấm: $guess = "c...t";. Vì các biểu thức chính quy coi một dấu chấm câu như là một ký tự duy nhất, chúng ta có thể dễ dàng duyệt qua hết danh sách các từ, tìm kiếm từ vừa khớp.

##### **Liệt kê 19. Duyệt qua danh sách các từ**

foreach ($words as $word) {  
 if (preg\_match("/^" . $\_POST['guess'] . "$/",$word)) {  
 echo $word . "<br />\n";  
 }  
}

Tùy vào chất lượng danh sách các từ của chúng ta và độ chính xác của mỗi phán đoán của chúng ta, ta sẽ có thể nhận được một danh sách từ hợp lý để sử dụng tìm các so khớp khả dĩ. Bạn sẽ phải tự quyết định liệu “chest” (rương, hòm; ngực) hay “cheat” (gian lận) là khớp hơn với “một từ gồm 5 chữ cái, có nghĩa là 'không chơi đúng luật.'"

[**Về đầu trang**](http://www.ibm.com/developerworks/vn/library/os-php-gamescripts1/#ibm-pcon)

## **Mad Libber**

Mad Libs là một trò chơi với các từ mà người chơi lấy một câu chuyện ngắn và thay thế các từ khoá biểu thị từ loại bằng một từ thuộc từ loại đó để tạo ra một phiên bản mới, ngớ ngẩn của câu chuyện đã cho. Lấy văn bản sau: “Tôi đang dạo chơi trong công viên thì nhìn thấy một hồ nước. Tôi nhảy xuống và uống phải quá nhiều nước. Tôi phải đi viện.” Bắt đầu bằng việc thay thế các từ loại bằng các thẻ từ thế chỗ. Bắt đầu và kết thúc mỗi thẻ từ thế chỗ bằng một dấu gạch dưới để tránh các trùng khớp ngẫu nhiên.

##### **Liệt kê 20. Thay thế các từ loại bằng các thẻ từ thế chỗ**

$text = "I was \_VERB\_ing in the \_PLACE\_ when I found a \_NOUN\_.   
I \_VERB\_ed in, and \_VERB\_ed too much \_NOUN\_. I had to go to the \_PLACE\_.";

Tiếp theo, hãy tạo ra vài danh sách từ cơ sở. Đối với ví dụ này, chúng ta sẽ không đi quá xa.

##### **Liệt kê 20. Tạo vài danh sách từ cơ sở**

$verbs = array('pump', 'jump', 'walk', 'swallow', 'crawl', 'wail', 'roll');  
$places = array('park', 'hospital', 'arctic', 'ocean', 'grocery', 'basement',   
 'attic', 'sewer');  
$nouns = array('water', 'lake', 'spit', 'foot', 'worm',   
 'dirt', 'river', 'wankel rotary engine');

Bây giờ chúng ta có thể lặp lại nhiều lần duyệt qua văn bản để thay thế các từ thế chỗ như yêu cầu.

##### **Liệt kê 22. Đánh giá văn bản**

while (preg\_match("/(\_VERB\_)|(\_PLACE\_)|(\_NOUN\_)/", $text, $matches)) {  
 switch ($matches[0]) {  
 case '\_VERB\_' :  
 shuffle($verbs);  
 $text = preg\_replace($matches[0], current($verbs), $text, 1);  
 break;  
 case '\_PLACE\_' :  
 shuffle($places);  
 $text = preg\_replace($matches[0], current($places), $text, 1);  
 break;  
 case '\_NOUN\_' :  
 shuffle($nouns);  
 $text = preg\_replace($matches[0], current($nouns), $text, 1);  
 break;  
 }  
}  
  
echo $text;

Rõ ràng, đây là một ví dụ đơn giản và thô thiển. Các danh sách từ của chúng ta càng chính xác bao nhiêu, và chúng ta đầu tư càng nhiều thời gian vào văn bản cơ sở của chúng ta bao nhiêu, thì kết quả mà ta nhận được càng tốt hơn bấy nhiêu. Chúng ta đã sử dụng các tệp tin văn bản để tạo ra các danh sách tên và danh sách từ cơ sở. Sử dụng cùng một nguyên tắc này, chúng ta có thể tạo ra các danh sách từ, phân chia theo từ loại và sử dụng chúng để tạo thêm các Mad Libs khác nhau.

[**Về đầu trang**](http://www.ibm.com/developerworks/vn/library/os-php-gamescripts1/#ibm-pcon)

## **Trình chơi xổ số**

Việc chọn đúng sáu số trong một trò xổ số, chẳng phải nói nhiều, là bất khả thi theo lý thuyết xác suất thống kê. Tuy nhiên, nhiều người vẫn trả tiền để chơi, và nếu bạn ưa thích các con số, nó có thể là thú tiêu khiển khi quan sát các xu hướng. Chúng ta hãy cùng tung ra một kịch bản lệnh cho phép ta theo dõi các số trúng thưởng và cung cấp sáu số được chọn ít nhất trong danh sách của chúng ta.

(Từ chối trách nhiệm: Việc này sẽ không giúp bạn trúng xổ số, vì vậy xin đừng chi tiền của bạn vào vé số. Đây chỉ là trò vui đùa).

Chúng ta ghi lưu các số vé trúng thưởng trong một tệp tin văn bản. Chúng ta phân tách từng số một bằng dấu phẩy và đặt mỗi bộ số trên một dòng riêng của nó. Khi chúng ta nhận được nội dung tệp tin, tách ra theo từng dòng, và tách mỗi dòng theo các dấu phẩy, chúng ta nhận được một thứ trông giống như Liệt kê 23.

##### **Liệt kê 23. Lưu các số vé trúng thưởng trong một tệp tin văn bản**

$picks = array(  
 array('6', '10', '18', '21', '34', '40'),  
 array('2', '8', '13', '22', '30', '39'),  
 array('3', '9', '14', '25', '31', '35'),  
 array('11', '12', '16', '24', '36', '37'),  
 array('4', '7', '17', '26', '32', '33')  
);

Rõ ràng, đó không hẳn là một tệp tin cơ sở để rút ra các số liệu thống kê. Nhưng đó là một sự khởi đầu, và đủ để minh họa các nguyên lý.

Thiết lập một mảng cơ sở để lưu giữ phạm vi được chọn. Chẳng hạn, nếu chúng ta chọn các số từ 1 đến 40 (ví dụ, $numbers = array\_fill(1,40,0);, sau đó duyệt qua từng lựa chọn, tăng tương ứng các giá trị ăn khớp nhau.

##### **Liệt kê 24. Duyệt qua từng lựa chọn của chúng ta**

foreach ($picks as $pick) {  
 foreach ($pick as $number) {  
 $numbers[$number]++;  
 }  
}

Cuối cùng, sắp xếp các số này theo giá trị. Thao tác này sẽ xếp các số được chọn ít nhất lên đầu mảng.

##### **Liệt kê 25. Sắp xếp các số theo giá trị**

asort($numbers);  
  
$pick = array\_slice($numbers,0,6,true);  
  
echo implode(',', array\_keys($pick));

Bằng cách bổ sung thêm đều đặn các số trúng thưởng thực tế vào tệp tin văn bản chứa danh sách các số được chọn của chúng ta, chúng ta có thể phát hiện xu thế chọn số trong một thời gian dài. Thật là thú vị khi thấy được một số chữ số thường xuất hiện ra sao.

[**Về đầu trang**](http://www.ibm.com/developerworks/vn/library/os-php-gamescripts1/#ibm-pcon)

## **Tóm tắt**

Bài viết này là một nguồn tài nguyên giúp bạn bước vào sử dụng PHP để giúp làm giàu thêm các trải nghiệm trò chơi của bạn. Trong Phần 2 của loạt bài “30 kịch bản lệnh trò chơi bạn có thể viết trong PHP”, chúng ta xây dựng tiếp dựa trên các mã đã có ở đây và đi vào các kịch bản lệnh liên quan sâu hơn, có thể mang lại cho bạn thậm chí nhiều ích lợi hơn nữa.

content 2

# Creating 10 fundamental scripts



Duane O'Brien

Published on November 18, 2008

[Facebook](http://www.facebook.com/sharer.php?u=http%3A%2F%2Fwww.ibm.com%2Fdeveloperworks%2Flibrary%2Fos-php-gamescripts1%2Findex.html&t=30%20game%20scripts%20you%20can%20write%20in%20PHP%2C%20Part%201%3A%20Creating%2010%20fundamental%20scripts)[Twitter](https://twitter.com/intent/tweet?url=http%3A%2F%2Fibm.co%2F2iWgCb5&text=30%20game%20scripts%20you%20can%20write%20in%20PHP%2C%20Part%201%3A%20Creating%2010%20fundamental%20scripts&via=developerWorks)[Linked In](http://www.linkedin.com/shareArticle?mini=true&url=http%3A%2F%2Fwww.ibm.com%2Fdeveloperworks%2Flibrary%2Fos-php-gamescripts1%2Findex.html&title=30%20game%20scripts%20you%20can%20write%20in%20PHP%2C%20Part%201%3A%20Creating%2010%20fundamental%20scripts)[Google+](https://plus.google.com/share?url=http%3A%2F%2Fwww.ibm.com%2Fdeveloperworks%2Flibrary%2Fos-php-gamescripts1%2Findex.html&t=30%20game%20scripts%20you%20can%20write%20in%20PHP%2C%20Part%201%3A%20Creating%2010%20fundamental%20scripts)E-mail this page

Comments

[3](https://www.ibm.com/developerworks/library/os-php-gamescripts1/#icomments)

### **Content series:**

## This content is part 1 of 3 in the series: **30 game scripts you can write in PHP**

As both a game master/storyteller and a developer, I frequently find myself writing little utilities and scripts to help me when running, planning, and playing games. Sometimes I need a quick idea. Other times, I just need a whole pile of names for Non-Player Characters (NPCs). Occasionally, I need to geek out on numbers, work out some odds, or integrate some word puzzles into a game. Many of these tasks become more manageable with a little bit of script work ahead of time.

This article will explore 10 fundamental scripts that can be used in various types of games. The code archive contains the full source for each script we will discuss, and you can see the scripts in action at [chaoticneutral](http://chaoticneutral.net/3d10).

Learn more. Develop more. Connect more.

The new [developerWorks Premium](https://www.ibm.com/us-en/marketplace/developer-program?cm_sp=dwpremium-_-os-php-gamescripts1-_-sidebar) membership program provides an all-access pass to powerful development tools and resources, including 500 top technical titles (dozens specifically for open source developers) through Safari Books Online, deep discounts on premier developer events, video replays of recent O'Reilly conferences, and more. [Sign up today](https://www.ibm.com/us-en/marketplace/developer-program?cm_sp=dwpremium-_-os-php-gamescripts1-_-sidebar).

We will blaze through these scripts pretty quickly. The topic of finding a host or setting up a server will not be covered. There are many Web hosting companies that offer PHP, and the XAMPP installer is easy to use if you want to set up your own. We won't spend a lot of time talking about PHP best practices or game design techniques. These scripts are designed to be simple to understand, simple to use, and quick to pick up.

## A basic die roller

Many games and game systems need dice. Let's start with something simple: rolling a single six-sided die. Essentially, there's no difference between rolling a six-sided die and picking a random number between 1 and 6. In PHP, this is simple: echo rand(1,6);.

In many cases, that would be more or less fine. But when we deal with games of chance, we want something a little better. PHP provides a better random number generator: mt\_rand(). Without going into detail on the differences between the two, it is safe to assume that mt\_rand is a faster and better random number generator: echo mt\_rand(1,6);. We'll be happier overall if we put this in a function.

##### Listing 1. Using the mt\_rand() random number-generator function

|  |  |
| --- | --- |
| 1  2  3  4  5 | function roll () {  return mt\_rand(1,6);  }    echo roll(); |

Then we can pass the type of die we want to roll as a parameter to the function.

##### Listing 2. Passing the type of die as a parameter

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | function roll ($sides) {  return mt\_rand(1,$sides);  }    echo roll(6); // roll a six-sided die  echo roll(10); // roll a ten-sided die  echo roll(20); // roll a twenty-sided die |

From here, we can go on to rolling multiple die at once, returning an array of results, or rolling multiple die of different kinds all at once, depending on our needs. But this simple script can be used for most tasks.

## Random name generator

If you're running games, writing stories, or creating a bunch of characters all at once, sometimes it's hard to keep coming up with new names. Let's look at a simple random name generator you can use to solve this problem. Starting off, let's make two simple arrays — one with first names, one with last names.

##### Listing 3. Two simple arrays of first and last names

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | $male = array(  "William",  "Henry",  "Filbert",  "John",  "Pat",  );    $last = array(  "Smith",  "Jones",  "Winkler",  "Cooper",  "Cline",  ); |

Then we can just pick a random element from each array: echo $male[array\_rand($male)] . ' ' . $last[array\_rand($last)];. To pull a bunch of names at once, we can simply shuffle the arrays and pull as many as we like.

##### Listing 4. Shuffling the name arrays

|  |  |
| --- | --- |
| 1  2  3  4  5 | shuffle($male);  shuffle($last);  for ($i = 0; $i <= 3; $i++) {  echo $male[$i] . ' ' . $last[$i];  } |

Taking this basic concept, we can create text files to hold our first and last names. If we put one name per line in our text file, we can easily split the file contents on the newline character to build our source arrays.

##### Listing 5. Creating text files for our names

|  |  |
| --- | --- |
| 1  2 | $male = explode('\n', file\_get\_contents('names.female.txt'));  $last = explode('\n', file\_get\_contents('names.last.txt')); |

Build or find some good name files (a couple are included in the [code archive](https://www.ibm.com/developerworks/library/os-php-gamescripts1/#download)), and we will never want for names again.

## Scenario generator

Taking the same basic principles we used to make the name generator, we can make what's called a *scenario generator.* This is useful in role-playing games or other situations where we need to come up with a pseudo-random set of circumstances that can be used for role-play, improvisation, writing, etc. One of my favorite games, Paranoia, includes something called a "mission blender" in its GM Pack. The mission blender can be used to put together a full mission at the quick roll of a die. Let's put together our own scenario generator.

Take the following scenario: You wake up lost in the woods. You know you have to get to New York, but you don't know why. You can hear the barking of dogs and the unmistakable sound of hostile searchers nearby. You are cold, shivering, and without a weapon. Each sentence in that scenario introduces a specific aspect of the scenario:

* "You wake up lost in the woods" — This establishes the setting.
* "You know you have to get to New York" — This describes an objective.
* "You can hear the barking of dogs" — This introduces an antagonist.
* "You are cold, shivering, and without a weapon" — This adds a complication.

Just like we created text files for our first and last names, start by making a text file each for settings, objectives, antagonists, and complications. Sample files are included in the code archive. Once we have these files, the code to generate a scenario looks much the same as the code to generate names.

##### Listing 6. Generating a scenario

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | $settings = explode("\n", file\_get\_contents('scenario.settings.txt'));  $objectives = explode("\n", file\_get\_contents('scenario.objectives.txt'));  $antagonists = explode("\n", file\_get\_contents('scenario.antagonists.txt'));  $complications = explode("\n", file\_get\_contents('scenario.complications.txt'));    shuffle($settings);  shuffle($objectives);  shuffle($antagonists);  shuffle($complications);    echo $settings[0] . ' ' . $objectives[0] . ' ' . $antagonists[0] . ' '  . $complications[0] . "<br />\n"; |

We can add elements to our scenarios by adding new text files, or we may wish to add multiple complications. The more we add to our base text files, the more varied our scenarios will be over time.

## Deck builder and shuffler

If you play cards and are interested in working on any scripts that are card-related, we want to put together a deck builder with a built in shuffler. To start, let's build a basic deck of standard playing cards. We need to build up two arrays — one to hold suits and one to hold faces. This will allow flexibility later if we want to add new suits or card types.

##### Listing 7. Building a basic deck of playing cards

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | $suits = array (  "Spades", "Hearts", "Clubs", "Diamonds"  );    $faces = array (  "Two", "Three", "Four", "Five", "Six", "Seven", "Eight",  "Nine", "Ten", "Jack", "Queen", "King", "Ace"  ); |

Then build a deck array to hold all the card values. We can do this simply using a pair of foreach loops.

##### Listing 8. Building a deck array

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | $deck = array();    foreach ($suits as $suit) {  foreach ($faces as $face) {  $deck[] = array ("face"=>$face, "suit"=>$suit);  }  } |

Once we have a deck array built, we can easily shuffle the deck and draw a random card.

##### Listing 9. Shuffling the deck and drawing a random card

|  |  |
| --- | --- |
| 1  2  3  4  5 | shuffle($deck);    $card = array\_shift($deck);    echo $card['face'] . ' of ' . $card['suit']; |

From here, it's a short path to drawing hands of a set number of cards or building a multideck shoe.

## Odds calculator: Card draw

Because we built the deck the way we did, keeping track of the face and suit individually for each card, we can make use of the deck programmatically to do something like calculate the odds of getting a specific card. Start by drawing out two hands of five cards each.

##### Listing 10. Drawing two hands of five cards each

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | $hands = array(1 => array(), 2=>array());    for ($i = 0; $i < 5; $i++) {  $hands[1][] = implode(" of ", array\_shift($deck));  $hands[2][] = implode(" of ", array\_shift($deck));  } |

Then we can look in the deck to see how many cards are left and what the odds are of drawing a specific card. How many cards are left is an easy one. That's just a count of how many elements there are in the$deck array. To get the odds of drawing a specific card, we need a function to walk through the whole deck and evaluate the remaining cards to see if they match.

##### Listing 11. Calculating the odds of drawing specific card

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | function calculate\_odds($draw, $deck) {  $remaining = count($deck);  $odds = 0;  foreach ($deck as $card) {  if ( ($draw['face'] == $card['face'] && $draw['suit'] ==  $card['suit'] ) ||  ($draw['face'] == '' && $draw['suit'] == $card['suit'] ) ||  ($draw['face'] == $card['face'] && $draw['suit'] == '' ) ) {  $odds++;  }  }  return $odds . ' in ' $remaining;  } |

Now we can pick the card we are trying to draw. To make this easy, pass in an array that looks just like a card. We can look for a specific card.

##### Listing 12. Looking for a specific card

|  |  |
| --- | --- |
| 1  2  3 | $draw = array('face' => 'Ace', 'suit' => 'Spades');    echo implode(" of ", $draw) . ' : ' . calculate\_odds($draw, $deck); |

Or we can look for a card of a given face or suit.

##### Listing 13. Looking for a card of a given face or suit

|  |  |
| --- | --- |
| 1  2 | $draw = array('face' => '', 'suit' => 'Spades');  $draw = array('face' => 'Ace', 'suit' => ''); |

## Simple poker dealer

Now that we've got a deck builder and something to help work out the odds of drawing specific cards, we can put together a really simple dealer to practice poker hands. For the purpose of this example, we build a dealer for five-card draw. The dealer will provide five cards from the deck. You specify which cards you want to discard by number, and the dealer will replace these cards with fresh ones from the deck. We won't bother putting in draw limits or house rules, though you may find that a rewarding personal exercise.

As shown in the previous section, generate and shuffle a deck, then create a single hand of five cards. Display these cards by their array index so that you can specify which cards to return. You might do it by using checkboxes to indicate which cards you are replacing.

##### Listing 14. Using checkboxes to indicate cards you are replacing

|  |  |
| --- | --- |
| 1  2  3  4 | foreach ($hand as $index =>$card) {  echo "<input type='checkbox' name='card[" . $index . "]'>  " . $card['face'] . ' of ' . $card['suit'] . "<br />";  } |

Then, evaluate the input array $\_POST['card'] to see which cards have been checked for replacement.

##### Listing 15. Evaluating the input

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | $i = 0;  while ($i < 5) {  if (isset($\_POST['card'][$i])) {  $hand[$i] = array\_shift($deck);  }  } |

Using this script, you can try your hand — pun intended — at figuring out the best ways to handle a specific set of cards.

## Hangman player

Hangman is essentially a word-guessing game. Given a word of a certain length, we have a limited number of letter guesses. If you guess a letter that appears in the word correctly, all occurrences of the letter are filled in. After a set number of wrong guesses (typically six), you've lost the game. To put together a crude game of hangman, we need to start with a word list. For now, let's make it a simple array.

##### Listing 16. Creating a word list

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | $words = array (  "giants",  "triangle",  "particle",  "birdhouse",  "minimum",  "flood"  ); |

Using techniques covered earlier, we can move these words to an external word list text file and import them as we like.

Once we've got a list of words, we need to pick one out at random, display a blank for each letter, and start taking guesses. We need to keep track of right and wrong guesses from guess to guess. We'll do this cheaply by just serializing the guess arrays and passing them along with each guess. If we wanted to keep people from cheating by viewing the page source, we'd want to do something a little more secure.

Build up arrays to hold our letters, and our right/wrong guesses. For right guesses, we'll fill an array with the letters as keys and periods as values.

##### Listing 17. Building arrays to hold letters and guesses

|  |  |
| --- | --- |
| 1  2  3  4 | $letters = array('a','b','c','d','e','f','g','h','i','j','k','l','m','n','o',  'p','q','r','s','t','u','v','w','x','y','z');  $right = array\_fill\_keys($letters, '.');  $wrong = array(); |

Now we need a little but of code to evaluate guesses and show the word as it progresses through the guessing game.

##### Listing 18. Evaluating guesses and displaying progress

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | if (stristr($word, $guess)) {  $show = '';  $right[$guess] = $guess;  $wordletters = str\_split($word);  foreach ($wordletters as $letter) {  $show .= $right[$letter];  }    } else {  $show = '';  $wrong[$guess] = $guess;  if (count($wrong) == 6) {  $show = $word;  } else {  foreach ($wordletters as $letter) {  $show .= $right[$letter];  }  }  } |

In the [source archive](https://www.ibm.com/developerworks/library/os-php-gamescripts1/#download), we can see how we serialize the guess arrays and pass them from guess to guess.

## Crossword helper

I know it's bad form, but sometimes when you're working on a crossword puzzle, you just get stuck trying to find a five-letter word that starts with C and ends with T. Using the same word list we put together for Hangman, we can easily search for words that fit a certain pattern. First, establish a way to pass in words. To make this easy, replace missing letters with periods: $guess = "c...t";. Since regular expressions treat a period as a single character, we can easily walk the word list, looking for matches.

##### Listing 19. Walking the word list

|  |  |
| --- | --- |
| 1  2  3  4  5 | foreach ($words as $word) {  if (preg\_match("/^" . $\_POST['guess'] . "$/",$word)) {  echo $word . "<br />\n";  }  } |

Depending on the quality of our word list and the accuracy of our guess, we should be able to get a reasonable list of words to use for possible matches. You'll have to decide yourself if "chest" or "cheat" is a better match for "a five-letter word that means 'to not play by the rules.'"

## Mad Libber

Mad Libs is a word game where the player takes a short story and replace key types of words with different words of the same type to create a new, sillier version of the same story. Take the following text: "I was walking in the park when I found a lake. I jumped in and swallowed too much water. I had to go to the hospital." Start by replacing the word types with different word tokens. Start and end each token with an underscore to prevent accidental string matches.

##### Listing 20. Replacing word types with word tokens

|  |  |
| --- | --- |
| 1  2 | $text = "I was \_VERB\_ing in the \_PLACE\_ when I found a \_NOUN\_.  I \_VERB\_ed in, and \_VERB\_ed too much \_NOUN\_. I had to go to the \_PLACE\_."; |

Next, create a couple basic word lists. For this example, we won't get too fancy.

##### Listing 21. Creating a couple of basic word lists

|  |  |
| --- | --- |
| 1  2  3  4  5 | $verbs = array('pump', 'jump', 'walk', 'swallow', 'crawl', 'wail', 'roll');  $places = array('park', 'hospital', 'arctic', 'ocean', 'grocery', 'basement',  'attic', 'sewer');  $nouns = array('water', 'lake', 'spit', 'foot', 'worm',  'dirt', 'river', 'wankel rotary engine'); |

Now we can repeatedly evaluate the text to replace the tokens as needed.

##### Listing 22. Evaluating the text

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | while (preg\_match("/(\_VERB\_)|(\_PLACE\_)|(\_NOUN\_)/", $text, $matches)) {  switch ($matches[0]) {  case '\_VERB\_' :  shuffle($verbs);  $text = preg\_replace($matches[0], current($verbs), $text, 1);  break;  case '\_PLACE\_' :  shuffle($places);  $text = preg\_replace($matches[0], current($places), $text, 1);  break;  case '\_NOUN\_' :  shuffle($nouns);  $text = preg\_replace($matches[0], current($nouns), $text, 1);  break;  }  }    echo $text; |

Obviously, this is a simple and crude example. The more precise our word lists, and the more time we put into our base text, the better our results will be. We've already used text files to create lists of names and basic word lists. Using the same principle, we can create lists of words broken up by type and use them to create more diverse Mad Libs.

## Lotto picker

Picking the right six numbers in a lotto is, to say the least, statistically improbable. Nevertheless, many people still pay to play, and if you like numbers, it can be entertaining to see the trends. Let's throw together a script that lets us keep track of winning numbers and provides the six least-picked numbers in our list.

(Disclaimer: This will not help you win the lotto, so please don't spend your money on tickets. This is just for fun.)

We save winning lotto picks in a text file. We separate individual numbers by comma and put each set of numbers on its own line. When we get the file contents, split on newlines, and split each line on commas, we get something that looks like Listing 23.

##### Listing 23. Saving winning lotto picks in a text file

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | $picks = array(  array('6', '10', '18', '21', '34', '40'),  array('2', '8', '13', '22', '30', '39'),  array('3', '9', '14', '25', '31', '35'),  array('11', '12', '16', '24', '36', '37'),  array('4', '7', '17', '26', '32', '33')  ); |

Obviously, that's not much of a base file for drawing stats. But it's a start, and enough to illustrate the principles.

Set a base array to hold the pick range. For example, if we pick numbers from 1 to 40 (e.g., $numbers = array\_fill(1,40,0);, then walk through our picks, incrementing appropriate matching values.

##### Listing 24. Walking through our picks

|  |  |
| --- | --- |
| 1  2  3  4  5 | foreach ($picks as $pick) {  foreach ($pick as $number) {  $numbers[$number]++;  }  } |

Finally, sort the numbers based on value. This should put the least-picked numbers at the front of the array.

##### Listing 25. Sorting the numbers based on value

|  |  |
| --- | --- |
| 1  2  3  4  5 | asort($numbers);    $pick = array\_slice($numbers,0,6,true);    echo implode(',', array\_keys($pick)); |

By regularly adding actual lotto picks to the text file containing our list of picks, we can trend number-picking over the long term. It's interesting to see how often some numbers appear.

## Summary

This article is a resource for you to get a jump into using PHP to help enrich your gaming experience. In [Part 2](http://www.ibm.com/developerworks/opensource/library/os-php-gamescripts2/) of this "30 game scripts you can write in PHP" series, we build on the code found here and go into more involved scripts that can give you even more benefit.

content 3

# [Lưu trữ dữ liệu trong game: php javascript mysql!](http://vietgamedev.net/blog/104/l%C6%B0u-tr%E1%BB%AF-d%E1%BB%AF-li%E1%BB%87u-trong-game-php-javascript-mysql/)

bởi [Nguyễn Tài Hải](http://vietgamedev.net/NguyenTaiHai/)

Trong quá trình làm game Cuộc Chiến Không Gian để trở thành game app của VietGamDev mình đã đối diện với vấn đề : truyền dữ liệu qua lại giữa Javascript, Php and mysql đã tốn của mình hơn 3 ngày làm việc. Với nhiều khó khăn, cheat chọt ( Solution) . Cuối cùng thì cũng đã có kết quả chia sẻ lại cho mọi người để tiết kiệm chi phí về sau vì mình tin chắc rằng cũng sẽ có một số bạn giống mình chọn con đường này để lưu trữ dữ liệu.



**Yêu cầu:**

Lưu dữ liệu từ game như: điểm số, tên, hình ảnh người chơi từ Javascript vào cơ sở dữ liệu mysql

Lấy dữ liệu từ mysql để thể hiện trên game.

**Đặt vấn đề**

+ Javascript là mã hoạt động phía client nên về nguyên tắc nó không thể kết nối trực tiếp đến cơ sở dữ liệu

+ PHP hoạt động ở server side nên phụ trách chuyện này rất tốt.

=> Giao tiếp dữ liệu Javascript (game) và mysql (database) thông qua PHP

**Truyền biến từ PHP qua Javascript**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | <?php  $data\_php= 10;  <?>  <!DOCTYPE html>  <html>  <head>  <title>Test</title>  <script language="javascript" type="text/javascript">  var data\_js= "<?php echo $data\_php?>";    </head>    <body>    </body>  </html> |

**Dùng PHP lấy dữ liệu từ mySQL**

Bạn muốn kết nối đến dữ liệu để lấy data có thể sử dụng ví dụ sau. Đây là lần đầu làm việc này nên các bạn nào chuyên nghiệp về PHP hãy gợi ý cho mình cách tốt hơn nhé.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | <?php  //Lấy thông tin trong cơ sở dữ liệu  $con = mysql\_connect("localhost","yourUserName","yourPass");  if (!$con)  {  die('Could not connect: ' . mysql\_error());  }  mysql\_select\_db("yourDataBase", $con);    $sql="SELECT \* FROM yourTable ORDER BY Score DESC LIMIT 5";    //Biến $result sẵn sàng dùng trong javascript  $result = mysql\_query($sql);  ?> |

**Truyền biến từ Javascript qua PHP:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | function ajax\_post(data\_js)  {  // Create our XMLHttpRequest object  var hr = new XMLHttpRequest();  // Create some variables we need to send to our PHP file  var url = "mysqlConnecter.php";    var vars = "data\_php=" + data\_js;    hr.open("POST", url, true);  // Set content type header information for sending url encoded variables in the request  hr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");  // Access the onreadystatechange event for the XMLHttpRequest object  hr.onreadystatechange = function() {  if(hr.readyState == 4 && hr.status == 200) {  var return\_data = hr.responseText;  }  }  // Send the data to PHP now... and wait for response to update the status div  hr.send(vars); // Actually execute the request  } |

Ở đây chúng ta thấy chúng ta thông qua javascript để gọi đến PHP file: "mysqlConnecter.php" với dữ liệu tuyền vào là data\_js thông qua phương thứ POST.

Đây là code của "mysqlConnecter.php"

|  |  |
| --- | --- |
| 1  2  3 | <?php  $data\_js = $\_POST['data\_php'];  ?> |

Lúc này mình có thể lấy data tha hồ mà insert vào mysql.

Vi dụ:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | <?php  $data\_js = $\_POST['data\_php'];    $con = mysql\_connect("localhost","yourUserName","yourPass");  if (!$con)  {  die('Could not connect: ' . mysql\_error());  }    mysql\_select\_db("yourDataBase", $con);    $sql="INSERT INTO yourTable (TestCollum)  VALUES  ('$data\_js')";    if (!mysql\_query($sql,$con))  {  die('Error: ' . mysql\_error());  }    mysql\_close($con);  ?> |

Mình đã làm và chạy trên Cuộc Chiến Không Gian và chia sẻ trong thảo luận: [http://www.vietgamedev.net/forum/thread/16/truy%E1...](http://www.vietgamedev.net/forum/thread/16/truy%E1%BB%81n-d%E1%BB%AF-li%E1%BB%87u-qua-m%E1%BA%A1ng-php-javascript-mysql/) .

Nếu bạn nào làm cách khác hay hơn chia sẻ cùng mọi người nhé.

nhận xét

trong h1” 30 kịch bản lệnh trò chơi bạn có thể viết trong PHP, Phần 1: Tạo 10 kịch bản lệnh cơ sở”

có trò chơi php trùng với từ khóa đã search