

从零开始学习 NPC

Hello, and welcome to my attempt to teach everyone NPC scripts better. Regardless of the countless tutorials, and countless Help threads that have been posted and solved, people still seem to have issues with NPCs. Hopefully, with this tutorial, people will learn something. Before we start, let me tell you a bit about myself. I joined RageZone in October of 2008. I started just like most of you guys, not knowing a god damn thing. I've had my share of nub questions asked, but I've also learned from the help others have given me. I read tutorial after tutorial to learn more, so I didn't have to rely on others to answer my questions. Sure, there are still quite a few things I don't know, no one knows everything. Anyone who says otherwise is a damn fool. The main point I'm trying to make is, if you truly want to be good at something, it takes work. You can't sit back, have others do the work, and expect to become a pro at whatever it is you wish to do. You have to show initiative, and want to learn. I hope my tutorial teaches you something, even if it's just one thing.

第一级：初心者

第一课：实用的脚本编辑器

诸多程序都能编写程序及脚本，有些还支持语法与高亮，如此一来就能看到花括号是从哪里开始与到哪里结束，整理脚本时给予不少帮助。以下是几款推荐的软件：

NotePad++

PsPad

第二课：Types of NPCs

NPC 主要有两种：需要 `status` 的，以及不需要 `status` 的。最简单的区分方法

就是...显然地...`status`。编写 NPC 时，时时刻刻扪心自问：我的 NPC 会需要两个以上的对话框吗？如果答案为是，那你绝大多数时候会需要用到

`status`；若答案为非，那 `status` 可能用不太到。请见以下范例：

不须 `status` 的 NPC

```
function start() {  
    cm.sendOk("我是一个没有 status 的 NPC。");  
    cm.dispose();  
}
```

另一个不须 `status` 的 NPC

```
function start() {  
    cm.sendOk("这是另一个没有 status 的 NPC。");  
}  
  
function action(mode, type, selection) {  
    cm.warp(100000000, 0);  
    cm.gainItem(4001126, 1);  
    cm.sendOk("看到了吗? 我不用 status 就能把你传送走并给你道具。");  
    cm.dispose();  
}
```

须要 `status` 的 NPC

```
var status;  
  
function start() {  
    status = -1;  
    action(1, 0, 0);  
}  
  
function action(mode, type, selection) {  
    if (mode == 1) {  
        status++;  
    } else {  
        status--;  
    }  
    if (status == 0) {  
        cm.sendNext("请按下一步, 我才能继续下一个 status。");  
    } else if (status == 1) {  
        cm.sendSimple("你看到我如何操作 status 了吗? \r\n #L0# 是 #1 \r\n #L1# 否 #1");  
    } else if (status == 2) {  
        if (selection == 0) {  
            cm.sendOk("我在这! 这是另一个 status。如你在脚本中看到的, 这个对话框是在 status 2。");  
        }  
    }  
}
```

```
        cm.dispose();
    } else if (selection == 1) {
        cm.sendOk("好吧，这个对话框也是在 status 2 :)");
        cm.dispose();
    }
}
}
```

第三课：与你和其他脚本作者沟通

编写脚本的重点之一就是能够了解你到底在做什么或是能够了解其他作者在干嘛。不作纪录还能有什么好方法帮你了解脚本？你可曾看过 `//` 出现在脚本某一行的后面？他们叫做「批注」。以下是两种批注：

```
// 这是一种批注

// 这种批注只能单行使用，这也是为什么它叫做单行批注。

/**
这里是多行批注，
任何被 * 包起来的东西都不会被脚本解释器解读。
**/
```

批注会被脚本解释器忽略，意味着任何「被批注」的东西都是给人读的。批注在你编写很长的脚本时能帮你纪录你正在做什么而十分有用，还能让你的程序代码读者在不懂程序时能了解你想做什么。

第四课：学习专门术语

该来的还是会来，你不可能不知道正确的专门术语就能编写 NPC。以下是几个小抄：

NPC 颜色代码、道具图片、其他

```
#b = 蓝色文本
#d = 紫色文本
#g = 绿色文本
#k = 黑色文本
#r = 红色文本
#e = 粗体文本
#n = 正常文本 (移除粗体)
```

```
#c[道具 ID]# 显示玩家背包中有多少 [道具 ID]
#h # - 显示玩家名称
#m[地图 ID]# - 显示地图名称
#o[怪物 ID]# - 显示怪物名称
#p[NPCID]# - 显示 NPC 名称
#q[技能 ID]# - 显示技能名称
#s[技能 ID]# - 显示技能图片
#t[道具 ID]# - 显示道具名称
#i[道具 ID]# - 显示道具图片
#z[道具 ID]# - 显示道具名称
#v[道具 ID]# - 显示道具图片
#x - Returns "0%" (need more information on this).
#B[%]# - 显示进度条
#f[图片地址]# - 显示 WZ 档案中的图片
#F[图片地址]# - 显示 WZ 档案中的图片
#L[数子]# 选项开始
#l - 选项结束
\r\n - 换行
\r = 确认(回车)
\n = 新行
\t = Tab (4 个空格)
\b = Backwards
```

cm.[指令]

dispose

结束与 NPC 的对话，让你可以与其他 NPC 对话。

用法：cm.dispose();

sendNext

显示一个带有「下一个」按钮的对话框。

用法：cm.sendNext("[文本]");

sendPrev

显示一个带有「上一个」按钮的对话框。

How to use: cm.sendPrev("[文本]");

sendNextPrev

显示一个带有「上一个」、「下一个」按钮的对话框。

用法：cm.sendNextPrev("[文本]");

sendOk

显示一个带有「确认」按钮的对话框。

用法：cm.sendOk("[文本]");

sendYesNo

显示一个带有「是」、「否」按钮的对话框，「否」将结束对话，除非额外改写。

用法：`cm.sendYesNo("[文本]");`

sendAcceptDecline

显示一个带有「接受」、「拒绝」按钮的对话框，「拒绝」将结束对话，除非额外改写。

用法：`cm.sendAcceptDecline("[文本]");`

sendSimple

显示一个不带有任何按钮的对话框。

用法：`cm.sendSimple("[文本]");`

sendStyle

显示一个选择造型的对话框。

用法：`cm.sendStyle("[文本]", [变数]);` // 你需要宣告该变量

warp

传送脚本到地图。

用法：`cm.warp([地图 ID], [传送点]);` // 预设则设定 [传送点] 为 0

openShop

开启商店窗口。

用法：`cm.openShop([商店 ID]);`

haveItem

检查角色是否有道具（背包或身上）。

用法：`cm.haveItem([道具 ID]);`

gainItem

给予或收回角色道具

用法：`cm.gainItem([道具 ID], [数量]);` // 设定 [数量] 为负数来收回道具

changeJob

修改角色的职业。

用法：`cm.changeJob([职业 ID]);`

getJob

取得角色的职业。

用法：`cm.getJob();`

startQuest

开始任务。

用法: `cm.startQuest([任务 ID]);`

completeQuest

完成任务。

用法: `cm.completeQuest([任务 ID]);`

forfeitQuest

放弃任务。

用法: `cm.forfeitQuest([任务 ID]);`

getMeso

取得角色枫币。

用法: `cm.getMeso();`

gainMeso

给予或收回角色枫币。

用法: `cm.gainMeso([数量]);` // 设定 [数量] 为负数来收回枫币

gainExp

给予或收回角色经验值。

用法: `cm.gainExp([数量]);` // 设定 [数量] 为负数来收回经验值

getLevel

取得角色的等级。

用法: `cm.getLevel();`

teachSkill

教角色技能。

用法: `cm.teachSkill([技能 ID], [技能等级], [技能最大等级]);`

get[状态]

取得角色的 [状态]。[状态] 可以是: HP, MP, STR, DEX, INT, LUK.

用法: `cm.get 状态;`

modifyNX

Gives/Takes the player nx

How to use: `cm.gainNX([amount]);`

Make it negative to make it take away.

// 外流端专属

modifyCSPoint

给予或收回点数。

用法: `cm.modifyCSPoint([点数], [种类]);` // [种类]: 1 = GASH 点数, 2 = 枫叶点数

检查枫币、道具、GM、性别

```
if (cm.getPlayer().isGM()) { // 检查是否为 GM

if (cm.getChar().isDonator() == true) { // checks for donator

if (cm.getJob().equals(net.sf.odinms.client.MapleJob.BOWMAN)) { //
checks for Bowman job (list of jobs in below spoiler)

if (cm.getLevel() >= 30) { // 检查等级是否足满 30

if (cm.getChar().getGender() == 0) { // 0 = 公, 1 = 母

if (cm.getPlayer().getGender() == 0) { // 0 = 公, 1 = 母

if (cm.getMeso() >= 数量) { // 检查枫币数量

if (cm.haveItem(itemid, amount)) { // 检查角色是否拥有某道具某数量

if (cm.getPlayer().getItemQuantity(itemid)); // 取得角色拥有某道具的数量
```

职业代码

```
BEGINNER - 0
WARRIOR - 100
FIGHTER - 110
CRUSADER - 111
HERO - 112
PAGE - 120
WHITEKNIGHT - 121
PALADIN - 122
SPEARMAN - 130
DRAGONKNIGHT - 131
DARKKNIGHT - 132
```

MAGICIAN - 200
FP_WIZARD - 210
FP_MAGE - 211
FP_ARCHMAGE - 212
IL_WIZARD - 220
IL_MAGE - 221
IL_ARCHMAGE - 222
CLERIC - 230
PRIEST - 231
BISHOP - 232
BOWMAN - 300
HUNTER - 310
RANGER - 311
BOWMASTER - 312
CROSSBOWMAN - 320
SNIPER - 321
CROSSBOWMASTER - 322
THIEF - 400
ASSASSIN - 410
HERMIT - 411
NIGHTLORD - 412
BANDIT - 420
CHIEFBANDIT - 421
SHADOWER - 422
PIRATE - 500
BRAWLER - 510
MARAUDER - 511
BUCCANEER - 512
GUNSLINGER - 520
OUTLAW - 521
CORSAIR - 522
MAPLELEAF_BRIGADIER - 800
GM - 500(v55) / 900(v62+)
SUPERGM 510(v55) / 910(v62+)
DAWNWARRIOR1 - 1000
DAWNWARRIOR2 - 1010
DAWNWARRIOR3 - 1011
DAWNWARRIOR4 - 1012
BLAZEWIZARD1 - 1100
BLAZEWIZARD2 - 1110
BLAZEWIZARD3 - 1111
BLAZEWIZARD4 - 1112
WINDARCHER1 - 1200
WINDARCHER2 - 1210


```
WINDARCHER3 - 1211
WINDARCHER4 - 1212
NIGHTWALKER1 - 1300
NIGHTWALKER2 - 1310
NIGHTWALKER3 - 1311
NIGHTWALKER4 - 1312
THUNDERBREAKER1 - 1400
THUNDERBREAKER2 - 1410
THUNDERBREAKER3 - 1411
THUNDERBREAKER4 - 1412
ARAN1 - 2100
ARAN2 - 2110
ARAN3 - 2111
ARAN4 - 2112
```

Keep in mind that some repacks use a different Terminology than this, so it's best to browse your repack/source folder to get familiar with yours.

第二级：中坚份子

第一课：Learning to code an NPC

编写 NPC 时最好对自己在干嘛有个概念，简单的方法是打开记事本，把你想要 NPC 做的事情写下来。例如：

道具交换

对话框 1 - 显示几个选项

对话框 2 - 对话框 1 选择后显示几个选项

对话框 1 选项 - 矿石换黄金枫叶标志、矿石换枫叶

对话框 2 选项 - 如果选择黄金枫叶标志，10 个矿石交换 1 个黄金枫叶标志，20 个矿石交换 3 个黄金枫叶标志

如果选择枫叶，5 个矿石交换 1 个枫叶，10 个矿石交换 3 个枫叶

以上只是个范例，但这样子做纪录，你就能事先想好最佳的布局，甚至能提醒你当初的想法，假如你没法一次写好脚本的话。

现在，在你开始之前，你要先思考你需要什么类型的 NPC。如果你需要参考，请回到第一级第二课。在这个范例中，我会使用需要 `status` 的 NPC，请阅读批注（第一级第三课）来了解我在做什么。

```

var status;

function start() { // NPC 开始

    status = -1; // 设定 status 为 -1

    action(1, 0, 0); // 设定 mode 为 1, type 为 0, selection 为 0

} // 关闭 start 函数


function action(mode, type, selection) { // calls what you set above in
function start, almost all actions are done here

    if (mode == 1) { // mode 被设为 1, 因为函数 start, 如上

        status++; // 让 NPC 进到下一个 status, 此时 status 变为 0

    } else { // 如果 mode 不是 1

        status--; // 不让 NPC 进到下一个 status

    }

    if (status == 0) { // 如果 mode 是 1, status 会从 -1 变为 0。如果
status 为 0, 以下会发生

        cm.sendSimple("哈啰, 我来示范如何使用 #belse#k 和 #bif#k, 准备好了
吗? \r\n #L0# 是的, 船长。 #l \r\n #L1# 不, 我还没好。 #l"); // 显示一个有两
个选项的对话框

    } else if (status == 1) { // 如果作出了选择, NPC 会进到下一个 status

        if (selection == 0) { // 选项 0 是 #L0#, "是的, 船长。"

            if (cm.haveItem(4001129, 10)) { // 检查道具

                cm.sendOk("如果 (if) 你选择选项 0, 简单来说就是第一个选项, 我
就会说这句话。"); // 如果 (if) 你有道具, 会显示这个对话框

                cm.dispose();
            } else {
                cm.sendOk("抱歉, 你没有该道具。"); // 否则 (else), 你没有该
道具
            }
        }
    }
}

```

```
        cm.dispose();
    }
} else if (selection == 1) { // "不，我还没好。"

    cm.sendOk("否则如果 (else if) 你选择选项 1，我会说这句话。");
    cm.dispose();
}
}
}
```

这里是一些能帮你弄懂 NPC 的东西：

sendNext(); & sendOk();

Type = 0

如果点了停止 - mode = -1

如果点了下一个/确认 - mode = 1

sendNextPrev();

Type = 0

如果点了停止 - mode = -1

如果点了下一个 - mode = 1

如果点了上一个 - mode = 0

sendYesNo();

Type = 1

如果点了停止 - mode = -1

如果点了是 - mode = 1

如果点了否 - mode = 0

sendAcceptDecline();

Type = 12

如果点了停止 - mode = -1

如果点了接受 - mode = 1

如果点了拒绝 - mode = 0

sendGetText();

没事儿

sendGetNumber();

Type = 3

如果点了停止 - mode = 0

如果点了确认 - mode = 1

sendSimple();

Type = 4

如果点了停止 - mode = 0

如果点了选择 - mode = 1

Credits: BENG

括号会是决定 NPC 能不能运作的关键。让我们看看上面 NPC 脚本的一个小片段：

```
if (cm.haveItem(4001129, 10)) { // 检查道具

    cm.sendOk("如果 (if) 你选择选项 0，简单来说就是第一个选项，我就会说这句话。"); // 如果 (if) 你有道具，会显示这个对话框

    cm.dispose();
}
```

看看第一行...

```
if (cm.haveItem(4001129, 10)) {
```

在第一行的尾端有一个左括号，这代表当条件为真时，其后所接的事情将会发生。以该行来说，如果你有 10 个 4001129，下一行就会执行。看看此片段的最后一行，注意到右括号了吗？

```
}
```

右括号结束条件，所以任何介于左括号与右括号之间的事情都将会在指定的条件成立后发生。右括号又经常伴随着 **else**，也就是当你没有足够的道具时，**else** 之后的片段便会执行。

第二课：使用运算符

在这一课，你会学到几种运算符以及如何用他们。运算符可以帮你决定某 A 是大于、小于还是等于某 B。以下是你最有可能会用到的运算符清单。

运算符	名称	类型	描述
!	非	一元	Returns true if 右操作数 evaluates to false. Returns false If the 右操作数 is true.
&&	条件 且	二元	If the operand on the left returns false, returns false without evaluating the operand on the right.
\	\		条件或

以上运算符常用在条件句之间。什么是条件句？那是 NPC 脚本常看到的东西。

```
if (cm.haveItem(itemid, amount)) {  
if (cm.getMeso() >= amount) {  
if (cm.getPlayer().getLevel() >= amount) {
```

以上都是条件句。因此，在其中运用运算符，可以让你指定某种条件。例如：

```
if (cm.haveItem(4001129, 50) && cm.getMeso() >= 1000) { // 此处需要角色拥有 50 黄金枫叶标志 且 拥有 1000 枫币
```

```
if (cm.haveItem(4001129, 50) || cm.getMeso() >= 1000) { // 此处需要角色拥有 50 黄金枫叶标志 或 拥有 1000 枫币
```

```
if (!cm.haveItem(4001129, 50)) { // 此处会检查角色是否 非 拥有 50 黄金枫叶标志
```

Using this is a good way to limit players from saving up loads of an item to cash in at once. 接着，我们还有关系运算符，他们很常被用在条件当中。列举其中一部分：

运算符	名称	描述
==	等于	Returns true if the expression on the left evaluates to the same value as the expression on the right.
<	小于	Returns true if the expression on the left evaluates to a value that is less than the value of the expression on the right.
<=	小于或等于	Returns true if the expression on the left evaluates to a value that is less than or equal to the expression on the right.
>	大于	Returns true if the expression on the left evaluates to a value that is greater than the value of the expression on the right.
>=	大于或等于	Returns true if the expression on the left evaluates to a value that is greater than or equal to the expression on the right.

我已经给过范例了，如果你没注意到的话，我再列几个给你：

```
if (cm.getMeso() >= 1000) { // 如果角色的枫币「大于或等于」1000
```

```
if (cm.getPlayer().getLevel() == 120) { // 如果角色的等级「等于」120
```

```
if (cm.getJobById() <= 112) { // 如果角色的职业「小于或等于」112 (Hero, Crusader, Fighter, Warrior, Beginner)
```

第三课：学习使用变量

在你学习编写时，你一定有发现一些以 `var` 起头的东西出现在脚本开头，那就是变量。简单来说，变量是用来「取代」东西的。你可以用你命名的变量来「取代」数字或文字等。一般来说，你会想要用变量来缩短脚本，即使只是几个字而已。范例：

```
var gl = 4000313;

function start() {
    cm.sendOk("哈啰，你带来 #v" + gl + "#了吗? "); // Calls the variable
    gl, and the information it is replacing

    cm.dispose();
}
```

As you can see from the example, the item id 4000313 is being replaced with the variable "gl". Even though it's only a minor difference, I shortend the script by 1 character. Now imagine having hundreds of places on a script where 4000313 was replaced by "gl". It adds up on how much space you save by using variables. That was an example on how to use a variable for a number. Here is one on how to use a variable for a string, or text sentence.

```
var yes = "太好了，你带齐了黄金枫叶！";
var no = "抱歉，你身上的黄金枫叶不够。";
var status;

function start() {
    status = -1;
    action(1, 0, 0);
}

function action(mode, type, selection) {
    if (mode == 1) {
        status++;
    }else{
        status--;
    }
    if (status == 0) {
        cm.sendSimple("#L0# 我有 10 个黄金枫叶 #1 \r\n #L1# 我有 20 个黄金
        枫叶 #1");
    }
}
```

```

    }else if (status == 1) {
        if (selection == 0) {
            if (cm.haveItem(4000313, 10)) {
                cm.sendOk(yes); // 使用变量「yes」，并显示你设定给他的内容

                cm.dispose();
            } else {
                cm.sendOk(no); // 使用变量「no」，并显示你设定给他的内容

                cm.dispose();
            }
        } else if (selection == 1) {
            if (cm.haveItem(4000313, 20)) {
                cm.sendOk(yes); // 使用变量「yes」，并显示你设定给他的内容

                cm.dispose();
            } else {
                cm.sendOk(no); // 使用变量「no」，并显示你设定给他的内容

                cm.dispose();
            }
        }
    }
}
}
}

```

如你清楚地看见，使用这些变量省下了很大的空间。我只用了 **111** 个字就打完了脚本，还包含开头的变量，而不用打长长的 **202** 个字。未来如果要编辑脚本，有使用变量的话就更加简单了。

第三级：高手过招

第一课：学习数组

好玩的来了！「高手」都会强调数组，如果能用就用。使用数组不仅让脚本更简洁，还能提高脚本整体的表现。数组十分容易修改，一旦你学会如何使用，你就能了解脚本作者在做什么。以下是几个数组的范例：

```

var item = [4000313, 4001129, 4001126];

function start() {

```



```

        cm.sendSimple("你想要什么? \r\n #L0# 黄金枫叶 #1 \r\n #L1# 黄金枫叶标志 #1 \r\n #L2# 枫叶 #1");
    }

    function action(mode, type, selection) {
        if (mode == 1) {
            cm gainItem(item[selection], 1);
        }
        cm.dispose();
    }

```

As you can see, the item ids are placed in the order they are shown in the sendSimple. The [selection] in the cm.gainItem method, calls the Array to determine which item to give the player. If they chose the second selection, it would call the second number in the Array. By placing the items in an Array, it shortens the script by a lot. Here is what the script would look like without an Array.

```

function start() {
    cm.sendSimple("你想要什么? \r\n #L0# 黄金枫叶 #1 \r\n #L1# 黄金枫叶标志 #1 \r\n #L2# 枫叶 #1");
}

function action(mode, type, selection) {
    if (mode == 1) {
        if (selection == 0)
            cm.gainItem(4000313, 1);
        else if (selection == 1)
            cm.gainItem(4001129, 1);
        else if (selection == 2)
            cm.gainItem(4001126, 1);
    }
    cm.dispose();
}

```

There is a clear difference between the two in which is longer. One of the most common ways to use Arrays, is with the infamous for loop, which I will explain in a later lesson.

第二课：学习多维数组与随机数

本节我会教你多维数组和随机数。Now, I'm not an expert at this, so I'll only be able to show you what I know. 什么是多维数组？多维数组就是由数组组成的数组。Basically, you can have more than one array, and simplify them further by making them into a multi-dimensional array. Unfortunately, I only know how to use them with randoms, so that is what I'll be teaching you. First, lets' set up a simple multi-dimentional array NPC.

```
var status;
var item = [[4001129, 4001129], [1082025, 1102023], [4000313, 1002085]];
var rand = Math.floor(Math.random() * 100);

function start() {
    status -1;
    action(1, 0, 0);
}

function action(mode, type, selection) {
    if (mode == -1) {
        cm.dispose();
    } else {
        if (status >= 2 && mode == 0) {
            cm.sendOk("再见! ");
            cm.dispose();
            return;
        }

        if (mode == 1) {
            status++;
        } else {
            status--;
        }

        if (status == 0) {
            cm.sendYesNo("你想要试试手气吗? ");
        } else if (status == 1) {
            var rand2;
            if ((rand >= 1) && (rand <= 50)) {
                rand2 = Math.floor(Math.random() * item[0].length);
            } else if ((rand >= 51) && (rand <= 90)) {
                rand2 = Math.floor(Math.random() * item[1].length);
            } else {
                rand2 = Math.floor(Math.random() * item[2].length);
            }
        }
    }
}
```

```

    }
    cm.gainItem([rand >= 1 && rand <= 50 ? item[0][rand2] : rand >=
51 && rand <= 90 ? item[1][rand2] : item[2][rand2]]);
    cm.sendOk("Congrats on your item.");
    cm.dispose();
  }
}

```

Alright, time to explain. This is the multi-dimensional array.

```

var item = [[4001129, 4001129], [1082025, 1102023], [4000313,
1002085]];

```

Each separate array is colored, so you can see the 3 different arrays within the multi-dimensional array. Next is the random part, or this little snippet.

```

var rand2;
  if ((rand >= 1) && (rand <= 50)) {
    rand2 = Math.floor(Math.random() * item[0].length);
  } else if ((rand >= 51) && (rand <= 90)) {
    rand2 = Math.floor(Math.random() * item[1].length);
  } else {
    rand2 = Math.floor(Math.random() * item[2].length);
  }

```

Ok, at the top of the script, you see this line.

```

var rand = Math.floor(Math.random()*100);

```

Think of this as if it were a dice. The number symbolizes the sides of a dice. So on this dice, there are 100 sides. Continue reading the random part as you read this. If the dice lands on side 1 - 50, give an item in the first array. If the dice lands on side 51 - 90, give an item in the second array. If it lands on any other side, give an item in the third array. This line here...

```

cm.gainItem([rand >= 1 && rand <= 50 ? item[0][rand2] : rand >= 51 &&
rand <= 90 ? item[1][rand2] : item[2][rand2]]);

```

Basically follows through with the action. That is all I know about these types of arrays.

第三课: Learning the Infamous For Loop

你可曾在脚本看见这样的东西?

```
for (var i = 0; i < options.length; i++)
    text += "\r\n#L"+i+"#" + options[i] + "#1";
```

This is the infamous for loop. It simplifies almost anything it is used with. It was created to specifically deal with Arrays. Here is the entire part of the code, so I can explain what it means.

```
var text = "#e#k What region have you trained in?#b";
var options = new Array("Aqua Road Region = 1 #v4001010#",
    "Ariant Region = 1 #v4001011#", "El Nath Region = 1 #v4001013#", "Ludas
    Lake Region = 1 #v4001012#", "Minar Forest Region = 1 #v4001009#",
    "Victoria Island Region = 5 #v4001126#", "World Tour = 5 #v4001129#",
    "Ores/Crystals = 1 #v4001014#");
for (var i = 0; i < options.length; i++)
    text += "\r\n#L"+i+"#" + options[i] + "#1";
cm.sendSimple(text);
```

Ok so, $i = 0$. While 0 is less than the options array, do the code. After executing the code, do $i++$. $i++$ basically tells the program running the script to increase i until it reaches the length of the array. So since the options array has a length of 8, it will keep increasing until it displays all 8 options.

Contributions from other members

Spoiler:

Quote Originally Posted by .:LastBreath:. [View Post](#)

Insert In NpcConversatonManager:

```
public void changeKeyBinding (int key, int type, int action) {
    getPlayer().changeKeybinding(key, new MapleKeyBinding(type,
    action));
    getPlayer().sendKeymap();
}
```

```
}
```

How to use

```
var status = 0;

function start() {
    status = -1;
    action(1, 0, 0);
}

function action(mode, type, selection) {
    if (mode == -1) {
        cm.dispose();
    } else {
        if (mode == 0 && status == 1) {
            cm.dispose();
            return;
        }
        if (mode == 1)
            status++;
        else
            status--;
        if (status == 0) {
            cm.sendYesNo("Hello there! Do you want to learn 二段跳?");
        } else if (status == 1) {
            cm.sendSimple("Which key do you want  
#fSkill/411.img/skill/4111006/icon# on?  
#b\r\n#L59#F1#L60#F2#L61#F3#L62#F4#L63#F5#L64#F6#L65#F7#L66#F8#L67#F9  
\r\n #L68#F10#L87#F11#L88#F12  
\r\n#L2#1#L3#2#L4#3#L5#4#L6#5#L7#6#L8#7#L9#8#L10#9#L11#0#L12#-#L13#=  
\r\n#L16#Q#L17#W#L18#E#L19#R#L20#T#L21#Y#L22#U#L23#I#L24#O#L25#P#L26#[#  
L27#]  
\r\n#L30#A#L31#S#L32#D#L33#F#L34#G#L35#H#L36#J#L37#K#L38#L#L39#;#L40#'  
\r\n#L42#Shift#L44#Z#L45#X#L46#C#L47#V#L48#B#L49#N#L50#M#L51#,#L52#.#L4  
2#Shift \r\n#L29#Ctrl#L56#Alt#L57#SPACE#L56#Alt#L29#Ctrl  
\r\n#L82#Ins#L71#Hm#L73#Pup#L83#Del#L79#End#L81#Pdn");
        } else if (status == 2) {
            cm.sendOk("There you go!");
            cm.changeKeyBinding(selection, 1, 4111006);
            cm.dispose();
        } else {
            cm.sendOk("See you next time then.");
        }
    }
}
```

```
        cm.dispose();
    }
}
}
```

CM command:

Code:

changeKeyBinding

allows you to place a skill anywhere in your key config.

How to use: cm.changeKeyBinding(selection, 1, SKILLID);

(THINKS THATS HOW TO USE?)

ALL CREDITS GOES TO "made4forum" FROM HIS RELEASE

LINK:Skill teaching NPC

....

Further Explanation of the For Loop

Quote Originally Posted by Rice

Final Words

This concludes the tutorial. There is still quite a bit I don't know, even with NPCs. If anyone has something they wish to add/contribute, feel free to post it and I'll include it with credits. I hope you enjoyed my wall of text and learned something valuable. Please leave credits if this is posted anywhere else. Lastly, if you need help with an NPC script or have trouble understanding something, feel free to ask [HERE](#). Do not quote the entire tutorial. If you do, I will ask a Mod to remove your post, because it's an inconvenience for others.

Credits:

Shawn aka bboy242 aka DevonsDaddy

Special Thanks:

Moogra and Osiris

.:LastBreath:.

made4forum

Alcohol

Rice