

این مار خوش خط و خال

آشنایی و کار با زبان برنامه‌نویسی پایتون

قسمت دوم

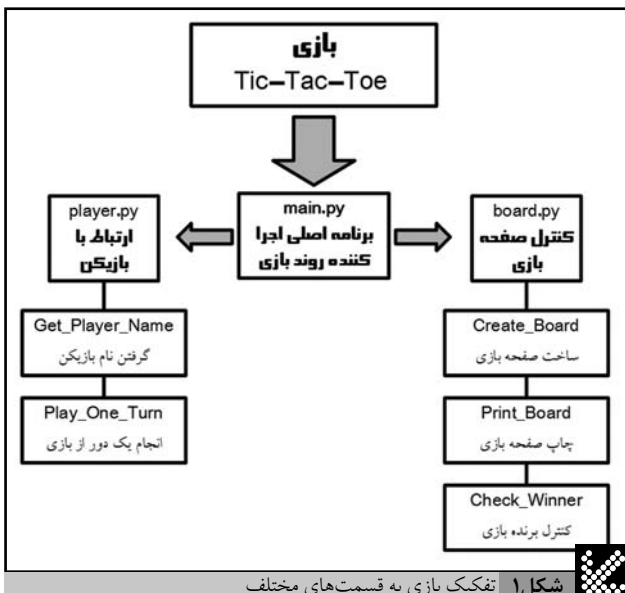
«احمد شریف پور»

در قسمت قبلی این مجموعه به طور خلاصه به معرفی زبان برنامه‌نویسی پایتون پرداختیم و چند نمونه ساده و ابتدایی از برنامه‌های پایتون را بررسی کردیم. در این قسمت با مفاهیم جدیدی آشنا خواهیم شد و برنامه به نسبت کاملی را پیاده‌سازی خواهیم کرد. این برنامه یک بازی بسیار ساده Tic-Tac-Toe را شبیه‌سازی می‌کند. اگرچه این بازی منطق پیچیده و خاصی ندارد، اما سیستم تفکیک کد، فراخوانی توابع، وارد کردن ماژول‌ها و... را به شما خواهد آموخت. این بازی از یک صفحه شطرنجی 3×3 تشکیل شده که بازیکنان به نوبت با حروف X یا O یکی از خانه‌های آن را اشغال می‌کنند. هدف هر بازیکن قرار دادن سه علامت در کنار هم به صورت سطری، ستونی یا قطری است. در بازی شبیه‌سازی شده ما بازیکن با حرف X در برابر کامپیوتر با حرف O بازی خواهد کرد.

تقسیم و غلبه (Divide & Conquer)

یکی از روش‌های بسیار پرکاربرد در حل مسئله و طراحی الگوریتم، روش تقسیم و غلبه است. در این روش هر مسئله به مسائلی ریزتر تقسیم می‌شود و این روند آن قدر ادامه می‌یابد تا به مسائلی بسیار ساده با راه حل آسان برسیم. حل هر مسئله ریز، به صورت جداگانه پیاده‌سازی شده و سپس از جمع آن‌ها راه حل نهایی به دست می‌آید. در برنامه‌نویسی پایتون این تقسیم‌بندی‌ها در مرحله نخست شامل تفکیک کل برنامه به تعدادی ماژول و سپس تفکیک ماژول‌ها به تعدادی شیء (Object) یا تابع (Function) است. ما مسئله بازی Tic-Tac-Toe را به صورتی مانند شکل ۱ تفکیک می‌کنیم.

ابتدا کل مسئله به سه بخش اساسی تقسیم می‌شود که عبارتند از: حلقه اصلی روند اجرای بازی، بخشی جهت تعامل با بازیکن (کاربر) و در نهایت بخشی برای کنترل و مدیریت صفحه بازی. هر یک از این بخش‌ها در فایل جداگانه‌ای به نام ماژول نوشته می‌شوند و وظایف مربوط به خود را از طریق تعدادی تابع به انجام می‌رسانند. فعلاً نگران این اصطلاحات نباشید. در پایتون همه چیز ساده‌تر از آن است که به نظر می‌رسد.



شکل ۱ تفکیک بازی به قسمت‌های مختلف

ماجول مربوط به بازیکن

کد فهرست ۱ را در IDLE یا هر ویرایشگر متن دلخواهی نوشته و آن را با نام player.py ذخیره کنید. شماره‌های خطوط را وارد نکنید. ما از آن‌ها برای توضیحات بعدی استفاده کرده‌ایم.

```
1 ##### Getting Player>s Name #####
2 def Get_Player_Name():
3     name = raw_input("Enter your name please: ")
4     print "Hi %s , you will play against computer. \
5     \nyour mark is <X>" %name
6     return name
7 ##### Play One Turn #####
8 def Play_One_Turn(b,player):
9     while 1:
10        # Player Turn
11        place=input("Your Turn %s.Enter Location:\n\
12 (Enter comma seperated values like 2,3): " %player)
13        x,y = place[0] , place[1]
14        if x < 1 or x > 3:
15            print "Error . . . \nX is invalid"
16            continue
17        if y < 1 or y > 3:
18            print "Error . . . \nY is invalid"
19            continue
20        if b[x-1][y-1] == " ":
21            b[x-1][y-1] = "X"
22            break
23        else:
24            print "This point is aquired by: %s" %b[x][y]
25    return
```

فهرست ۱ | ماجول مربوط به بازیکن

خط 1، یک توضیح یا Comment است. علامت # در هر کجای خط قرار بگیرد، پایتون عبارات بعد از آن را نادیده می‌گیرد. از توضیحات به طور معمول برای تقسیم‌کردن کد به بخش‌های قابل فهم برای انسان و شرح روند اجرای برنامه استفاده می‌شود تا مطالعه کد و ویرایش‌های بعدی آن ساده‌تر شود.

خط 2 یک تابع را تعریف می‌کند. تابع یا function یکی از عناصر اساسی هر زبان برنامه‌نویسی است. حتی در زبان‌های برنامه‌نویسی شیء‌گرای کنونی نیز توابع یا روال‌ها (subroutine) وظیفه اصلی تقسیم‌بندی و تفکیک کد را انجام می‌دهند. تابع قسمتی از کد است که با گرفتن ورودی‌های مشخص، خروجی مشخصی را تولید می‌کند یا کاری را به انجام می‌رساند. تابع به خودی خود اجرا نمی‌شود. برای استفاده از یک تابع باید آن را صدا زد یا به اصطلاح فراخواند. در پایتون تعریف تابع (که دارای مقدار برگشتی است) با تعریف روال (که فاقد مقدار برگشتی است) تفاوتی ندارد. به‌طور کلی تعریف تابع در پایتون به شکل زیر انجام می‌شود:

```
def <function_name> ( [list of arguments] ):
    function body
    return [List of values to return]
```

به تورفتگی بدنه تابع، اختیاری بودن آرگومان‌های ورودی و مقادیر خروجی توجه کنید. توجه داشته باشید که نام توابع و متغیرها نباید از بین کلمات کلیدی پایتون (فهرست ۲) انتخاب شود. همچنین استفاده از علامت‌های _ و __ (زیر خط تکی و زیر خط دوتایی) در ابتدای نام تابع یا

متغیر، ویژگی‌های خاصی را به آن‌ها نسبت خواهد داد که بررسی آن در این مقاله نمی‌گنجد.

همان‌گونه که از پرانتز خالی جلوی نام تابع Get_Player_Name مشخص است، این تابع آرگومان ورودی ندارد. ما از این تابع برای دریافت نام بازیکن و نمایش پیغام خوشامد استفاده خواهیم کرد.

در خط 3 از فرمان raw_input برای گرفتن یک رشته یعنی نام کاربر استفاده شده است. در سری 2.x پایتون برای گرفتن اعداد از input() و برای گرفتن رشته‌ها از raw_input() استفاده می‌شود. اما در سری 3.x پایتون دستور raw_input حذف شده و تنها از دستور input استفاده می‌شود که در هر صورت یک رشته را برمی‌گرداند و تبدیل نوع داده ورودی به عدد یا سایر انواع باید توسط برنامه‌نویس صورت بگیرد. دستورات input و raw_input هر دو از توابع درونی پایتون هستند. این توابع یک آرگومان ورودی دارند که یک رشته است. این رشته به عنوان پیام برای کاربر نمایش داده شده و جواب کاربر مقدار برگشتی تابع است. در این جا مقدار برگشتی (یا همان پاسخ کاربر) به متغیر name نسبت داده می‌شود. پایتون تعداد زیادی تابع توکار یا درونی دارد که گستره وسیعی از عملیات پرکاربرد را به انجام می‌رسانند.

در پایتون لازم نیست پیش از استفاده از متغیر آن را تعریف کرد. با نخستین دستور انتساب، متغیر تعریف شده و نوع آن براساس مقدار نسبت داده شده تعیین می‌شود. در اینجا متغیر name خود به خود از نوع رشته خواهد بود.

در خط 4 از دستور print برای چاپ پیام خوشامد استفاده شده است. به علامت \ در انتهای خط توجه کنید. همان‌طور که در شماره گذشته اشاره شد، از این علامت برای شکستن خطوط طولانی به تعدادی خط منطقی استفاده می‌شود. همچنین توجه داشته باشید که انتقال خروجی دستور چاپ به سطر بعد، در حین اجرا با عبارت \n انجام می‌شود. نکته آخر در این دستور علامت % است. برای جای دادن داده‌های خاص نظیر اعداد و رشته‌های ذخیره شده در یک متغیر، در دل یک رشته (به عنوان مثال، رشته‌ای که قرار است چاپ شود) از این گونه علامت‌ها استفاده می‌شود. از 'd%' و 'f%' نیز برای جای‌گذاری اعداد صحیح و اعشاری استفاده می‌شود. سیستم این جای‌گذاری به فرم زیر است:

```
" string %<s|d|f|...> string " % (
<variable> [,variable , variable, . . . ] )
```

یعنی هر عبارت بعد از علامت % بیرون رشته، به ترتیب جایگزین یکی از علامت‌های موجود در رشته خواهد شد. اگر مجموعه دستورات زیر را در نظر بگیریم:

```
score = 86
field = "mathematic"
print "I got %d in %s" % (score , field)
```

خروجی آن برابر «I got 86 in mathematic» خواهد بود.

در نهایت، در خط 6 توسط کلمه کلیدی return، مقدار متغیر name به صدا زنده تابع بازگردانده می‌شود.

خط 8 تعریف یک تابع جدید است و همان‌گونه که از نام آن بر می‌آید، تعامل با بازیکن برای انجام یک نوبت از بازی را به عهده دارد. پرانتز پس

```

1 ##### Creating the Board #####
2 def Create_Board():
3     b=[]
4     for i in range(3):
5         b.append([])
6         for j in range(3):
7             b[i].append(" ")
8     return b
9 ##### Printing the Board #####
10 def Print_Board(b):
11     s=""
12     for j in range(2,-1,-1):
13         s=s+"\n-----\n|"
14         for i in range(3):
15             s= s+ " " + b[i][j]+ " |"
16         s=s+"\n-----\n"
17     print s
18     return
19 ##### Checking for Winner #####
20 def Check_Winner(b):
21     for sign in ["X" , "O"]:
22         #check rows
23         for i in range(3):
24             score=0
25             for j in range(3):
26                 if b[i][j] == sign :
27                     score += 1
28             if score == 3 :
29                 return sign
30         #check columns
31         for j in range(3):
32             score=0
33             for i in range(3):
34                 if b[i][j] == sign :
35                     score += 1
36             if score == 3 :
37                 return sign
38         #check diameters
39         score1=score2 = 0
40         for i in range(3):
41             if b[i][i]== sign:
42                 score1 +=1
43             if b[i][2-i] == sign:
44                 score2 += 1
45         if score1 == 3 or score2 == 3:
46             return sign
47         #check if board is full
48         empty = 0
49         for i in range(3):
50             for j in range(3):
51                 if b[i][j] == « «:
52                     empty += 1
53         if empty == 0:
54             return "No one"
55         return None

```

and	assert	break	class
continue	def	del	elif
else	except	exec	finally
for	from	global	if
import	in	is	lambda
not	or	pass	print
raise	return	try	while

از نام تابع نشان می‌دهد که این تابع دو مقدار ورودی دارد. ما این دو مقدار را به ترتیب صفحه بازی b و نام بازیکن player در نظر گرفته‌ایم. مقدار b به اصطلاح به تابع پاس شده است تا از روی آن، نقطه مورد نظر بازیکن با نقاط خالی صفحه مقایسه شود. نام بازیکن هم برای نمایش پیغام‌ها مورد استفاده قرار خواهد گرفت.

در خط 9 حلقه تعامل با کاربر آغاز می‌شود. این حلقه تا زمانی که کاربر یک نقطه مناسب (آزاد و در محدوده صفحه بازی) را انتخاب نکرده باشد، ادامه خواهد یافت. همان‌گونه که در قسمت قبل دیده‌اید یکی از راه‌های خروج از حلقه دستور break است که کنترل عملیات را به خط بعد از حلقه منتقل می‌کند.

در خط 11 محل مورد نظر کاربر پرسیده می‌شود. بازیکن باید محل را به صورت یک جفت عدد که با کاما (,) از یکدیگر جدا شده‌اند وارد کند. متغیر حاصل یعنی place از دید پایتون یک تویپل (tuple) است. تویپل در زبان پایتون بسیار شبیه لیست است، اما تفاوت آن با لیست غیر قابل تغییر بودن آن است؛ یعنی عناصر یک لیست را می‌توان ویرایش کرد، اما عناصر موجود در تویپل را نمی‌توان تغییر داد. بعدها درباره تویپل بیشتر صحبت خواهیم کرد. تنها نکته مهم این است که برای ارجاع به عناصر لیست یا تویپل از علامت [] در جلوی نام لیست یا تویپل استفاده می‌شود. توجه کنید که مانند غالب زبان‌های برنامه‌نویسی اندیس تویپل‌ها و لیست‌ها از عدد صفر آغاز می‌شود.

بنابراین، در خط 13، place[0] عضو نخست تویپل را به عنوان x و place[1] عضو دوم آن را به عنوان y جدا می‌کند.

در خطوط 14 تا 19، مقادیر x و y کنترل می‌شوند تا در محدوده صفحه باشند. اگر اعداد در محدوده صفحه بازی (3x3) نباشد یک پیغام خطا چاپ شده و به کمک دستور continue حلقه گردش بعدی را انجام می‌دهد. در هر حلقه، چه از نوع for و چه از نوع while، دستور continue باعث می‌شود که روند فعلی حلقه قطع شده و کنترل دوباره به ابتدای بدنه حلقه بازگردد. در خط 20 کنترل می‌شود که آیا محل انتخاب شده (که اکنون مطمئن هستیم در محدوده صفحه بازی است) قبلاً اشغال شده یا خیر؟ اگر اشغال نشده باشد، علامت مختص بازیکن در آن ثبت شده و با دستور break از حلقه خارج می‌شویم. چون اندیس لیست‌ها و تویپل‌ها از صفر شروع می‌شود برای کنترل صفحه بازی از x و y یک واحد کم کرده‌ایم.

خط 23 استفاده از دستور else را نشان می‌دهد. در صورتی که شرط if آخر درست نباشد، کنترل به دستور else منتقل می‌شود. در این حالت پیامی مبنی بر اشغال بودن نقطه مورد نظر چاپ شده و حلقه دوباره از ابتدا اجرا می‌شود.

همان‌طور که در خط 25 مشاهده می‌کنید، این تابع مقدار برگشتی ندارد. نکته مهمی که همیشه باید به خاطر داشته باشید این است که در پایتون لیست‌های پاس شده به تابع نظیر b، به صورت «ارجاع به محل» یا By Reference مورد استفاده قرار خواهند گرفت و بنابراین هر تغییری در آن به طور مستقیم به متغیر اصلی (اینجا یعنی صفحه اصلی بازی board) اعمال خواهد شد و در نتیجه به مقدار بازگشتی نیازی نخواهد بود.

ماجول مربوط به صفحه بازی

در فهرست ۲، ماجول دوم که سازنده و کنترل کننده صفحه بازی است، نمایش داده شده است. این ماجول را تایپ کرده و با نام board.py ذخیره کنید. با یک نگاه کلی و با توجه به توضیحاتی که تاکنون داده‌ایم، مشخص می‌شود که این ماجول سه تابع برای ساخت صفحه نمایش، چاپ آن و کنترل برنده بازی تعریف می‌کند. در خط 3 متغیری به نام b از نوع لیست تعریف می‌شود. این متغیر در این مرحله هیچ عضوی ندارد.

در خط 4 با یک دستور for حلقه‌ای با سه بار تکرار تعریف می‌شود که در خط 5 به کمک متد append یک لیست تهی دیگر را به لیست b اضافه می‌کند. در دو خط بعدی با یک حلقه for دیگر، به این لیست داخلی، سه رشته با مقدار " " افزوده می‌شود. در نهایت متغیر b به فرم زیر در خواهد آمد:

```
b= [ [ " ", " ", " " ], [ " ", " ", " " ], [ " ", " ", " " ] ]
```

همان‌طور که پیش‌تر توضیح دادیم برای دسترسی به عناصر داخل لیست از اندیس‌ها و علامت [] استفاده می‌کنیم. بنابراین، b[0] نخستین عضو b است که خود یک لیست دیگر است و b[1][2] به سومین عضو در دومین عضو b اشاره می‌کند. ما از این تجرید برای شبیه‌سازی مختصات یا همان x و y صفحه بازی استفاده کرده‌ایم. به عبارت دیگر، عنصر نخست b نماینده ستون نخست یا x=1 و به عنوان مثال، عنصر دوم نماینده ستون دوم خواهد بود. به همین شکل مقدار b[2][1] نماینده محتویات نقطه‌ای با مختصات (۲،۲) خواهد بود.

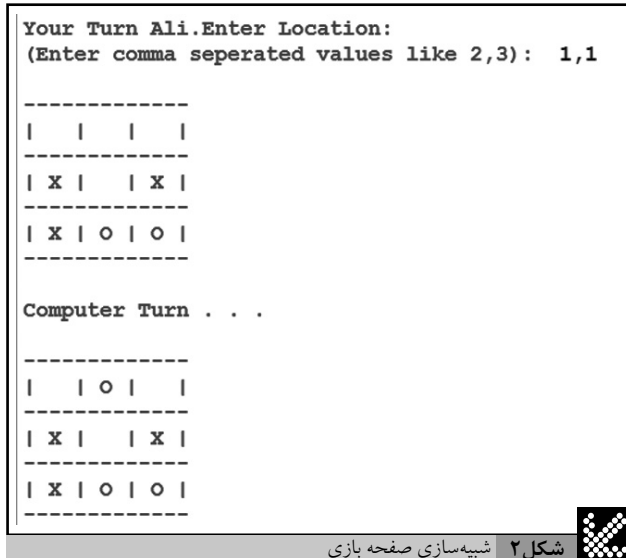
لیست‌ها که از قسمت قبل با تعریف آن‌ها آشنا شده‌اید، از پرکاربردترین انواع بنیادی پایتون هستند. لیست در واقع چیزی شبیه آرایه در سایر زبان‌ها است، با این تفاوت که لازم نیست از ابتدا اندازه آن را مشخص کنید یا نوع داده آن را تعریف کنید و همچنین هنگام اجرای برنامه می‌توانید هم خود لیست و هم عناصر آن را تغییر دهید. همان‌گونه که دیدید با متد append می‌توان هر چیزی را به لیست اضافه کرد.

در خط 10 تابع دیگری برای چاپ صفحه نمایش تعریف می‌شود. در خط 11 متغیر s از نوع رشته و با مقدار تهی تعریف شده است.

خط 12 نمونه دیگری از کاربرد range() را نشان می‌دهد. همان‌طور که به احتمال حدس زده‌اید، range() هم یکی از توابع داخلی پایتون است. در این خط، تابع range() سه آرگومان گرفته است که به ترتیب عدد شروع، عدد پایان و میزان تغییر در هر مرحله است. فرم کلی دستور range() مانند زیر است:

```
range ( [start , ] <stop> [,step] )
```

اکنون می‌دانید که عبارت فوق یعنی آرگومان stop یا پایان لیست تولیدی الزامی ولی آرگومان شروع و مقدار افزایش در هر مرحله اختیاری است. نتیجه خروجی range(2,-1,-1) در اینجا لیستی با مقدار [2, 1, 0] است. دلیل استفاده از این لیست معکوس این است که هنگام چاپ، لیست از بالا به پایین یعنی از خانه‌هایی به مختصات (1,3)، (2,3) و (3,3) چاپ می‌شود در حالی که مقادیر موجود در متغیر b به ترتیب از (1,1)، (1,2) و الی آخر شروع می‌شوند. مجموعه خطوط 13 تا 16 با ترکیب عناصر b و کاراکترهای " " و " | " جدول‌هایی شبیه شکل ۲ را برای بازی به وجود آورده و چاپ می‌کنند. به دلیل طولانی بودن تابع سوم ما از توضیح خط به خط آن صرف نظر می‌کنیم. این تابع به رغم ظاهر طولانی، منطقی بسیار ساده دارد. به طور خلاصه این که در خط 21 حلقه‌ای تعریف می‌شود که تنها دو بار یک



شکل ۲ شبیه‌سازی صفحه بازی

بار با مقدار X (نماینده بازیکن) و بار دیگر با مقدار O (نماینده کامپیوتر) اجرا خواهد شد. در خطوط 22 تا 29، با یک سری حلقه for امتیاز X یا O در همه سطرها محاسبه می‌شود که اگر تنها در یک سطر برابر ۳ بود، علامت X یا O با دستور return به عنوان برنده به صدا زنده تابع بازگردانده شده و عملکرد تابع تمام می‌شود. در خطوط 31 تا 37 همین کار برای ستون‌ها انجام می‌شود. در نهایت در خطوط 39 تا 46، قطرهای صفحه بازی کنترل می‌شود. رسیدن کنترل اجرا به خط 48، به این معنی خواهد بود که بازی تاکنون هیچ برنده‌ای نداشته است. در این جا کل صفحه برای وجود خانه‌های خالی بررسی می‌شود که در صورت عدم وجود بازی مساوی اعلام خواهد شد. در صورتی که صفحه بازی هنوز جای خالی داشته باشد، مقدار None یا هیچ، که از مقادیر از پیش تعریف شده پایتون است، برگشت داده می‌شود.

ماجول اصلی

فهرست ۴ ماجول اصلی یا کنترل کننده برنامه را نشان می‌دهد. این مجموعه کد را تایپ کرده و با نام main.py در کنار سایر فایل‌ها ذخیره کنید. به خط 1 که با علامت # شروع شده است، به اصطلاح شی‌بنگ یا shebang گفته می‌شود. این خط در محیط‌های لینوکسی به سیستم عامل اعلام می‌کند که برای اجرای این اسکریپت (در صورتی که به آن مجوز اجرا داده شده باشد) از چه مفسر یا مترجمی استفاده کند. در سیستم ویندوز یا سایر موارد این خط همانند یک توضیح یا comment دیده می‌شود.


خطوط 4 و 5 و 6 ماجول‌ها و توابع مورد نظر را بارگذاری می‌کنند. دستور import برای استفاده از ماجول‌ها به کار می‌رود. این دستور برای یافتن ماجول مورد نظر نواحی خاصی از سیستم را جست‌وجو می‌کند. یکی از این نواحی پوشه یا فولدری است که فایل import کننده در آن قرار دارد. ناحیه دیگر کتابخانه استاندارد یا Standard Library است. دستور import به دو صورت قابل اجرا است. روش اول به شکل زیر است:

```
import <module> [,module ,module , . . . ]
```

روش دوم نیز مطابق نمونه زیر است:

```
from <module name> import <function|variable|class> [,function|variable|class , . . . ]
```

تفاوت این دو حالت در این است که روش نخست خود ماجول را به صورت یک موجودیت وارد اسکریپت فعلی می‌کند و برای استفاده از اجزای آن باید از سیستم نقطه‌گذاری استفاده کرد. نظیر آنچه در خط 21 و دستور random.randint() دیده می‌شود. اما در حالت دوم بخش مورد نظر به طور مستقیم وارد اسکریپت فعلی شده و بدون نام بردن ماجول مادر آن قابل استفاده است. همانند خط 11 که در آن Print_Board() که از ماجول board وارد شده بود، به طور مستقیم مورد استفاده قرار گرفته است. در خط 4، random که یکی از کتابخانه‌های استاندارد پایتون است، بارگذاری می‌شود. همان‌طور که قبلاً گفته شد، کتابخانه استاندارد پایتون طیف وسیعی از کاربردها را به سادگی برای کاربر فراهم می‌کند. کتابخانه random توابعی در زمینه کار با اعداد تصادفی فراهم کرده است. سه خط نخست بخش اصلی، نام بازیکن را پرسیده و پس از خوشامدگویی، صفحه بازی را ایجاد و چاپ می‌کند. پس از آن حلقه بازی شروع می‌شود. این حلقه در خطوط 15 تا 18 و خطوط 28 تا 31 در صورت برنده شدن یک بازیکن یا پرشدن صفحه شکسته می‌شود. در این حلقه به ترتیب در خط 13 بازیکن از طریق تابع Play_One_Turn() یک نوبت بازی می‌کند. سپس در خط بعدی صفحه بازی چاپ شده و وجود یا عدم وجود برنده بررسی می‌شود. سپس نوبت به کامپیوتر می‌رسد که با یک حلقه while و استفاده از اعداد تصادفی سعی می‌کند، یک خانه صفحه را انتخاب کند. تابع randint() که در کتابخانه random تعریف شده است، با گرفتن دو عدد صحیح، یک عدد صحیح تصادفی بین آن دو (یا مساوی یکی از آن‌ها) تولید خواهد کرد. پس از خروج از این حلقه، صفحه بازی دوباره چاپ شده، برنده کنترل می‌شود و حلقه اصلی از نو شروع به کار خواهد کرد.

اگرچه این برنامه چندان کاربردی نبود و در بسیاری از قسمت‌ها می‌توان آن را به گونه‌ای بهتر پیاده‌سازی کرد، اما حاوی نکته‌های بسیار مهمی درباره کار با زبان پایتون است. اگرچه کل کدهای این مقاله از سایت مجله قابل دریافت است، اما توصیه می‌کنیم دست‌کم یک بار آن‌ها را تایپ کنید تا با خطاهای احتمالی حین تایپ و رفع اشکال برنامه نیز آشنا شوید. در قسمت بعدی با بخش‌هایی پرکاربرد از کتابخانه استاندارد پایتون آشنا خواهید شد و برنامه قسمت بعد، استفاده مفیدتر و جالب‌تری از چرخه‌های پردازنده شما خواهد کرد. 

پی‌نوشت:

۱- در نوشتن فرم کلی دستورات علامت <> به معنای اجباری، [] به معنای اختیاری و علامت | به معنای یا است.

```

1  #!/usr/bin/env python
2
3  ##### importing required modules #####
4  import random
5  from board import Create_Board , Print_Board , Check_Winner
6  from player import Get_Player_Name , Play_One_Turn
7
8  ##### Main Section #####
9  player=Get_Player_Name()
10 board = Create_Board()
11 Print_Board(board)
12 while 1:
13     Play_One_Turn(board,player)
14     Print_Board(board)
15     winner = Check_Winner(board)
16     if winner <> None:
17         print "Game Over \n%s won the game . . ." %winner
18         break
19     while 1:
20         # Computer Turn
21         x = random.randint(1,3)
22         y = random.randint(1,3)
23         if board[x-1][y-1] == " ":
24             board[x-1][y-1] = "O"
25             print "Computer Turn . . ."
26             break
27     Print_Board(board)
28     winner = Check_Winner(board)
29     if winner <> None:
30         print "Game Over \n%s won the game . . ." %winner
31         break

```

فهرست ۴ ماجول اصلی



برای خرید آنلاین کتاب‌های
انتشارات عصر شبکه به آدرس
www.shabakeh-mag.com

مراجعه کنید.

www.shabakeh-mag.com



شکستن مرزهای
فروش

کتابی که باید یک میلیون تومان فروخته می‌شد!

نویسنده: زان بتوسیان