Advanced

Wazz Online Distributed EBooks'

CONTACT : WASE39@GMAIL.COM

ENHANCE YOUR C# APPLICATIONS

النسخة العربية

Using Microsoft Agent

Author :

Waseem Abu Zenah

إعداد : م. وسيم أبوزينة

الفهرس :

- حول هذا وذاك
- مقدمة لابد منها
- متطلبات النظام
- ماهو عميل مايكروسوفت Microsoft agent
- إضافة عميل مايكروسوفت الى تطبيقك .
- خصائص العميل .

حول الكاتب :

الاسم : وسيم أبوزينة

Wase39@gmail.com

مواليد: □□/□□/□□□□

الحالة الاجتماعية : عم بدور على بنت الحلال

الدراسة الحالية :

طالب في كلية الهندسة المعلوماتية جامعة دمشق - تخصص ذكاء صنعي

العمل :

خبرة كبيرة في كتابة المقالات عبر الانترنت والموسوعات العربية

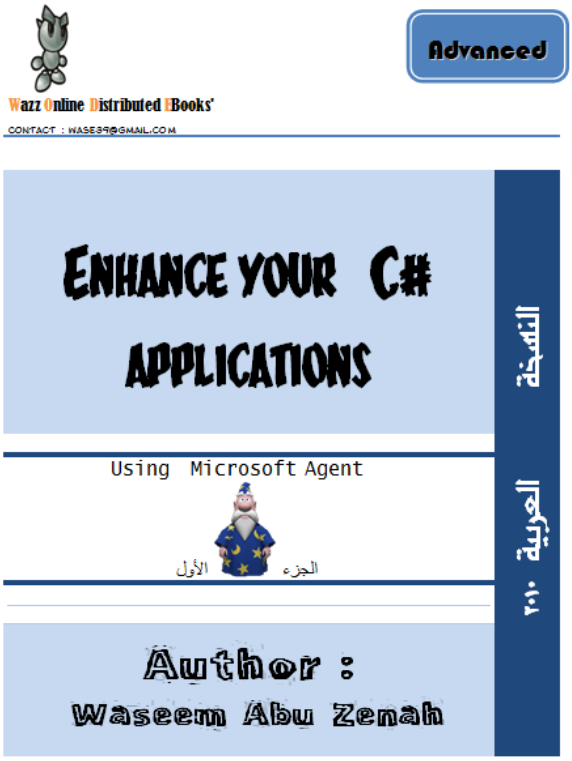
لدي مشاركات في موسوعة المعرفة العربية

ومشاركات في مواقع اجنبية code project وغيره

كما قمت بالعديد من المشاريع البرمجية

لغات البرمجة التي اتقنها :

- Pascal
- C++
- Java
- C#
- Visual basic
- Prolog
- php
- python
- ruby
- Perl
- LISP
- MIPS Assembly
- F#
- T-SQL



حول الكتاب :

استخدام عميل مايكروسوفت الجزء 1

السلسلة :

حسن تطبيقاتك البرمجية باستخدام C#

المستوى :

متقدم .

الرخصة :

GPL

تتناول السلسلة :

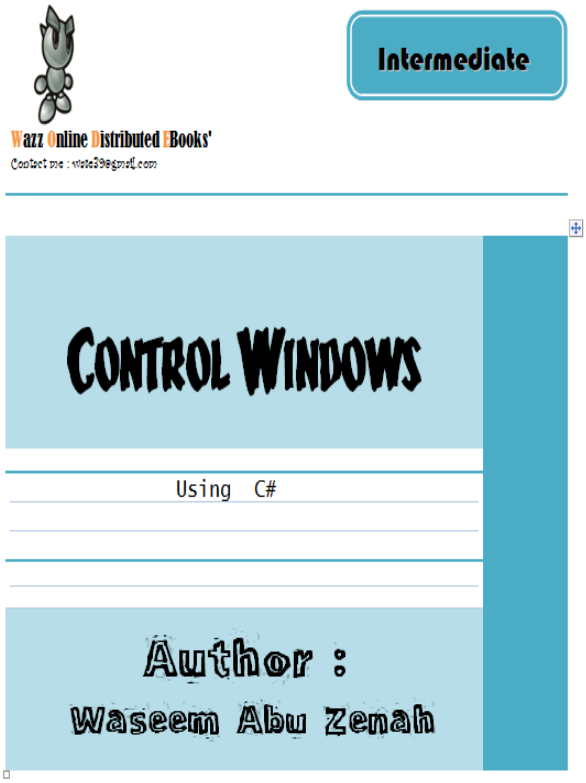
عدة دروس من المستوى متقدم ومافوق في لغة C# بحيث يمكنك من خلالها ان تقوم بإضافة جماليات او اساسيات لعمل وتسريع عمل تطبيق C# الخاص بك بحيث يتم تناول المواضيع النادرة في المنتديات العربية .

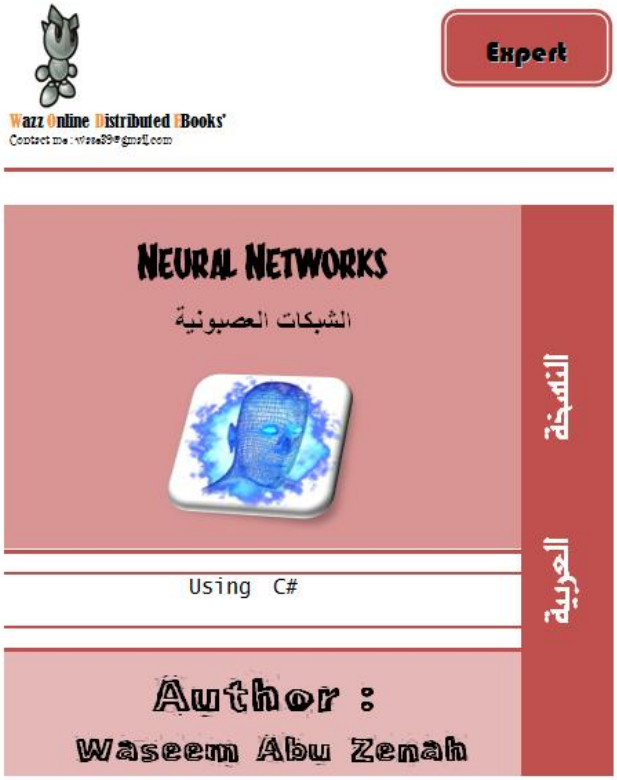
من اهم الدروس في السلسلة :

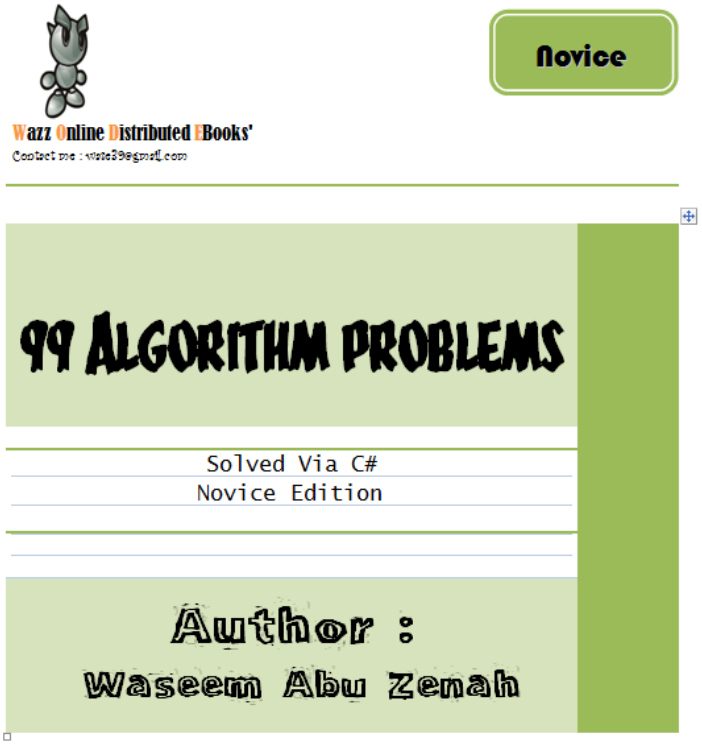
1- ربط مشروعك مع اي مشاريع برمجية بلغات اخرى
2- الواجهة الزجاجية في C#
3-كيفية ادارة الذاكرة بشكل فعال وكيفية بناء Garbage collector خاص فيك
4-مفهوم ال Reflection
5-استعمال مكتبات التعرف على صوت المستخدم
6-استعمال المكتبات الرسومية ثنائية البعد
7-استعمال المكتبات الرسومية ثلاثية البعد باستخدام opengl
8-استعمال المكتبات الرسومية ثلاثية البعد باستخدام XNA
9-C# Conventions

والعديد من الدروس الاخرى التي سيتم توفرها عبر الانترنت بشكل مجاني .
كل ذلك تحت رخصة GPL الحرة .بحيث يمكنك ان تقوم بالاقتباس ونشر الكتاب شريطة نشر اسم الكاتب .

سلاسل اخرى بلغة C# قيد العمل :







مقدمة :

لقد تطورت منصات العمل والمكاتب البرمجية بشكل كبير بحيث باتت تلبي معظم الاحتياجات .فامست المشاكل التي يتعرض لها المطور developer تتطلب نمطين من الحلول .

النمط الاول : هو ان يكون لدى المشكلة حل وبنى معطيات data structure اساسية متوفرة في تلك المكتبات كمثال على ذلك بنى مشكلة معينة تعتمد في حلها على بنية معطيات stack,hash, Queue

والنمط الثاني : هو ان يكون لدى متطلبات حل المشكلة بنى معطيات غير ذلك وبالتالي نحن امام إيجاد بنية معطيات جديدة لتناسب الحل .

كمثال على ذلك بنية المعطيات في منطق لغة برمجة prolog التي تعتمد على مبدأ المحاكمات والمنطق . والتي وفقاً لذلك تم تطوير بنية معطيات خاصة لتلبية متطلبات المحاكمات المنطقية والعقلية .

بناء على الكلام السابق فإنه يمكن ان نقسم المبرمجين Developers الى نمطين

1- الزبون client

مهمة المبرمج الزبون هي التعرف على الخدمات المقدمة من قبل منصات عمل ومكاتب برمجية عديدة كمنصة dot net frame work وكيفية الاستخدام الامثل لتلك الخدمات المقدمة لحل القضايا .

الصلاحيات المتاحة : هي استعمال الصفوف classes والتوابع methods ذات الخاصية public

2- المبرمج implementers

مهمة المبرمج هو القيام بإنشاء بنى معطيات وصفوف classes من الصفر . وهي المهمة الاصعب ولكنها تتيح لك كامل صلاحيات التحكم .

الصلاحيات المتاحة : له كامل الصلاحيات كونه منشأ المكتبة .

ويجب التنويه الى انه يجب الاهتمام بالجانبين كونه لا يجب اعادة اختراع العجلة من البداية في معظم الاحيان ولا يمكن ان يكون المطور عبارة عن فهرس للمكتبات البرمجية فحسب .

تهدف هذه السلسلة من الدروس بالتحديد الى التعريف ببعض تقنيات مكتبة dot net frame work الاصدار 3 من وجهة نظر client programmer

متطلبات النظام :

Microsoft Agent core components دعم مكونات عميل مايكروسوفت

نظام تشغيل Windows يدعم بشكل افتراضي Microsoft Agent core components بجميع النسخ ، ولكن في النسخة Windows 7 لا توفر حالياً دعم لتلك المكونات مع وعد مايكروسوفت بتوفيرها بالتحديثات اللاحقة . للتحقق من دعم نظام تشغيل ويندوز 7 لتلك المكونات :

اذهب الى قرص النظام (بشكل افتراضي C) ثم المجلد windows

اذا وجدت مجلد باسم MSAGENT فاعلم ان الميزة مدعومة .

في حال عدم توفر ذلك المجلد في مجلد النظام قم بداية بتحميل المكونات الاساسية

الكلمات المفتاحية : windows 7+ hot fix for Microsoft agent

في حال لم تكن تملك المكونات السابقة قم بالبحث عن Microsoft Agent 2.0 SDK

- نماذج وموديلات Models من لاحقة acs :

يمكنك تحميل شخصيات بشكل مجاني من الانترنت من لاحقة asc كما يوفر الرابط التالي بعض النماذج الجاهزة. http://www.cantoche.com/english/gallery/msagent.htm

في حال اردت تصميم شخصيتك الخاصة قم بتحميل برنامج Microsoft Agent Character Editor المجاني

- محرك تعرف على الصوت speech recognition (اختياري لتفعيل بعض الميزات)
- محرك Text to speech لتمكين العميل من الكلام .
- المحركين السابقين متوفرين بشكل افتراضي ضمن بيئة تشغيل ويندوز فيستا ومايليها من اصدارات .

ما هو عميل مايكروسوفت Microsoft Agent ؟

هو تقنية جديدة طبقت من قبل شركة مايكروسوفت لتنمذج شخصيات كرتونية متحركة تفاعلية تعتمد محرك Text To Speech والتعرف على الصوت speech recognition بهدف تحسين عملية التفاعل مع المستخدمين .

يتم تعريف عميل مايكروسوفت بانه عنصر activeX **وبالتالي يمكنه العمل على صفحات الويب بشكل مباشر بدون اي برامج اضافية .**

الاساس النظري وراء تقنية عميل مايكروسوفت تأتي من دراسة ومعالجة اللغات الطبيعية .

أول ظهور لعميل مايكروسوفت كان في حزمة برامج Microsoft office 97

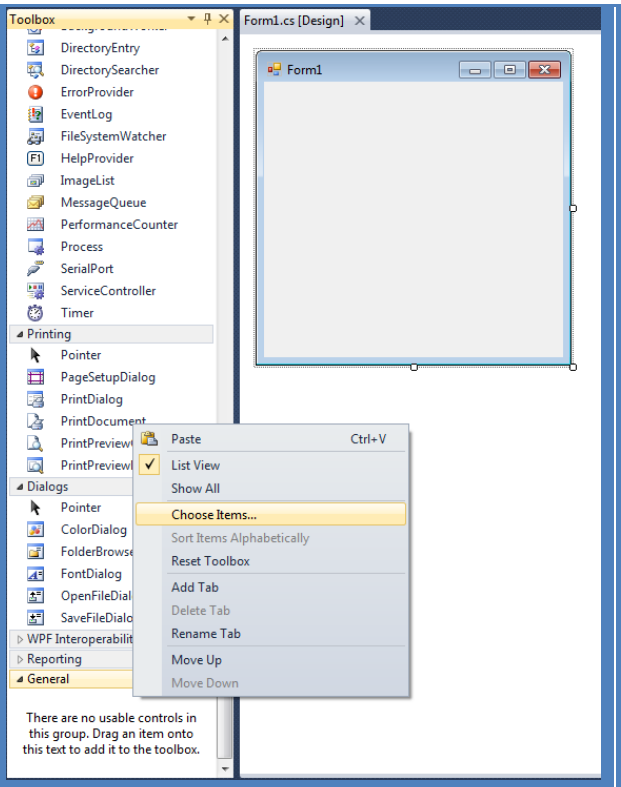
كيفية إضافة عميل مايكروسوفت ؟

المرحلة الاولى : إنشاء برنامج windows form application

1- قم بفتح برنامج visual studio واختر جديد new project

2- قم باختيار WindowsFormsApplication C#

المرحلة الثانية : قم بإضافة Agent control الى مشروعك



1- من نافذة الادوات toolbar قم بالضغط على زر الفأرة الايمن ثم اختيار choose items ثم قم بإضافة Microsoft Agent Control 2.0 ستجدها ضمن COM Components tab .

في هذه المرحلة سيقوم visual studio بالبحث عن جميع المكتبات ضمن مجلد النظام system ومسار النظام المحدد ضمن خيارات الكمبيوتر variables path .

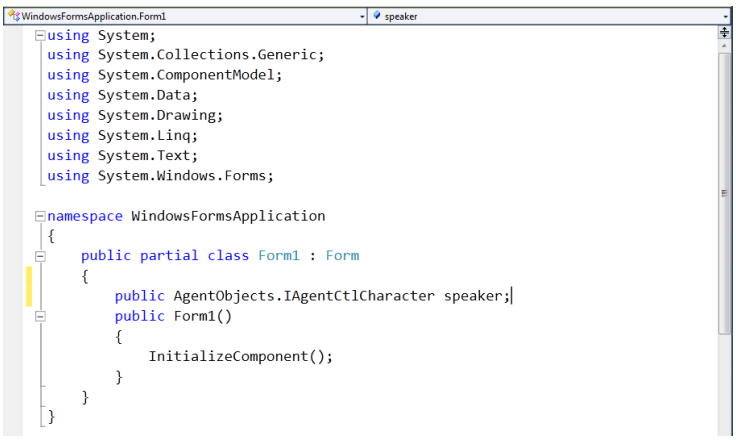
2-عند الإضافة ستظهر لك هذه الايقونة في شريط الادوات . قم بعمل drag and drop لهذه الوحدة التحكمية في مشروعك على الواجهة form كما تقوم بإضافة اي زر او اي اداة اخرى وقم بتسمية الغرض الناشىء ب MainAgent حالياً بعد تحديده وعمل الخطوة السابقة ضمن مربع properties .

تهانينا لقد اصبح لديك عميل مايكروسوفت ضمن تطبيقك ولكن ليس هو الرئيسي .

3-قم بتعريف متحول ضمن الصف form1 class.

public AgentObjects.IAgentCtlCharacter speaker ;

سيكون هذا المتحول speaker هو المتحول الذي سنتعامل معه من الآن فصاعداً .



4-قم بعمل تهيئة للمتحول speaker قم بإنشاء تابع وليكن Load_MS_Agents() وقم باستدعاءه في باني constructer صف form class .

```
private void Load_MS_Agents()
    {
      try
      {
        MainAgent.Characters.Load("Check ", @"C:\Agents\ Check.acs");
        *speaker = MainAgent.Characters["Check"];
        speaker.Show(0);
      }
      catch (FileNotFoundException)
      {
        MessageBox.Show("No such character ");
      }
    }
```

التعليمة MainAgent.Characters.Load

تاخذ محددين المحدد الاول يكون اسم الشخصيةID لكي نتعامل معها والمتحول الثاني هو المكان الذي يوجد فيه الموديل او النموذج الكرتوني .

نقوم بعد ذلك باسناد الشخصية الكرتونية الى الغرض speaker وفق التعليمة *

ثم نقوم بعرضه عن طريق التابع show(0) method

وفي حال لم نجد الملف سيظهر رسالة خطأ لنا .

خصائص العميل :

1- تحميل الشخصية وعرضها :

```
public  AgentObjects.IAgentCtlCharacter speaker ;

MainAgent.Characters.Load("CharacterID", path);

Speaker = MainAgent.Characters("CharacterID");
```

يتم تعريف MainAgent عن طريق عملية drop item التي قمنا بها قبل قليل .
بينما يتم تعريف Speaker كغرض instance من نمط IAgentCtlCharacter.
بعد ذلك يتم تحميل الشخصية الناطقة من قبل MainAgent بحيث يمكن تصور كلا من الغرضين بكرة مرسومة في عالم ثلاثي البعد بحيث نخزن المواصفات الرياضية للكرة بصف و نخزن مواصفات الكرة الرسومية في صف آخر .

تعيين اللغة الافتراضية الى اللغة الإنجليزية :

```
Speaker.LanguageID = &H409;
```

لإظهار العميل :

```
Speaker.Show();
```

لإخفاء العميل :

```
Speaker.Hide();
```

لإلغاء تحميل شخصية :

```
MainAgent.Characters.Unload("CharacterID")
```

2- الحصول على معلومات العميل :

من الهام جداً الحصول على معلومات العميل كي نستطيع تحديد الحركات التي يمكن ان يقوم بها على سبيل المثال لا الحصر ومعلومات اخرى :

```
String sName , sDescription , sExtraData;


sName = Speaker.Name                    //Character's Name property
sDescription = Speaker.Description      // Character's Description
sExtraData = Speaker.ExtraData          // Character's ExtraData
```

الحصول على اسماء الحركات التي يمكن للعميل ان يقوم بها :

سنقوم بوضع جميع اسماء الحركات ضمن ComboBox بغرض instance يدعى cboCombo

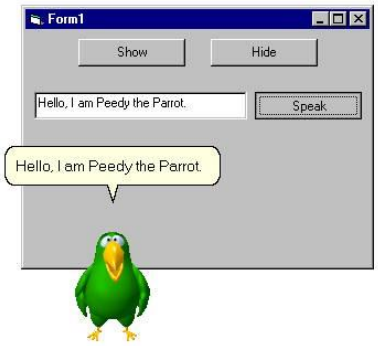
```
string animationName ;
cboCombo.Items.Clear() ; // Clear the ComboBox
For Each (animationName In Character.AnimationNames)
        cboCombo.Items.Add(animationName)
```

سنرى اهمية هذه الخطوة بعد قليل .

الجدول التالي يوضح اهم الحركات للشخصية Peedy

Table 2 - Peedy Animation Sequences

Animation	Return Animation	Supports Speaking	Description
Announce	exit branches	yes	Paper airplane flies in and unfolds
Congratulate	exit branches	yes	Displays blue ribbons
DoMagic1	none	yes	Raises magic wand
GetAttention	GetAttentionReturn	yes	Jumps up with wings outstretched
Greet	exit branches	yes	Bows
Pleased	exit branches	yes	Smiles
Process	none	no	Uses calculator
Read	ReadReturn	yes	Opens magazine, reads, and looks up
RestPose	none	yes	Neutral position
Search	none	no	Rotates with telescope
Suggest	exit branches	yes	Displays light bulb
Think	exit branches	yes	Looks up with wing on face
Wave	exit branches	yes	Waves
Write	WriteReturn	yes	Takes out pencil and pad, writes, and looks up



3- إضفاء الحركة والكلام للعميل :

جعل العميل يتكلم :

تحتاج الى محرك text to speech (موجودة افتراضياً ضمن ويندوز فيستا ومابعد)

```
Speaker.Speak("Hello");
```

تحريك العميل في المكان animation:

لتحريك العميل لابد من معرفة اسم الحركة التي سيقوم بالعمل وفقها ومن هنا يتبين اهمية خطوة تحصيل معلومات العميل .

لنفترض العميل merlin الذي يملك الحركة ذات الاسم Sad لذا يمكنني ان اقوم بالتالي :

```
Speaker.Play("Sad");
```

من المهم وضع معالجة للاخطاء هنا ففي حال كانت الشخصية لا تملك هذه الحركة سيعطي خطأ اثناء التنفيذ .

4- نقل العميل وفق محاور احداثيات والإيماءات :

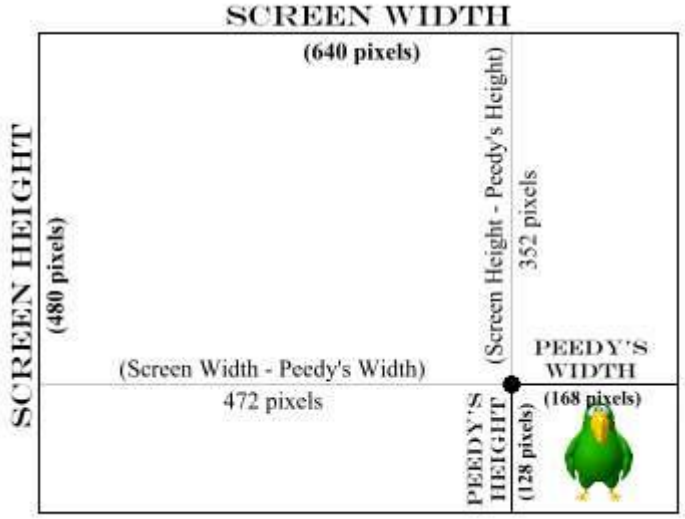
تحريك العميل في الشاشة transformation:
لنقل العميل الى الاحداثيات X coordinate 20, Y coordinate 45

```
Speaker.MoveTo(20, 45);
```

إيماء العميل في الشاشة gesture :

الإيماء يقصد به تحريك العميل بسرعة الضوء والفرق بينه وبين الخطوة السابقة هي امكانية ظهور الحركة من النقطة ا الى النقطة ب في الحالة الاولى وعدمها في الحالة الثانية

```
Speaker.GestureAt(20, 45);
```



5- التعرف على الصوت Speech recognition:

إضافة تعليمات الى العميل :
التعليمات هي تلك الجمل التي يقوم المستخدم بادخالها عن طريق الصوت المدخل للمايكروفون . ولكن العميل لن يميز التعليمات المعطاة من قبل المستخدم مالم تقوم ببرمجة تصرفات العميل وتعليمه لتلك العمليات وكيفية التصرف تجاه كل تعليمة . لكي تضيف تعليمة للعميل قم بما يلي :

```
Speaker.Commands.Add("LetsGoForLunch.", , _"Lets go for lunch.", True, False)
```

تقوم الجملة السابقة بإضافة التعليمة "LetsGoForLunch" الى العميل كي يفهمها لاحقاً .
ويجب التفريق بين اول LetsGoForLunch مكتوبة بالتعليمة وبين الثانية بحيث ان الاولى تدل على اسم التعليمة والثانية هي الاسم المفترض ان يتم سماعه من قبل المستخدم .
فيمكن ان يوجد لدينا تعليمة مسماة A1 وعند سماع الجملة التالية Come on سيقوم البرنامج عند سماعه come on بالبحث عن تابع ليقوم بمعالجة التعليمة المدعوة A1 .

سنتخصص في هذا الموضوع بشكل تفصيلي اكثر في دروس اخرى
ارجو ان ينال الدرس اعجابكم