

Empezando a Programar con KPL

Por Jon Schwartz

Versión en Español: [José Antonio Méndez A.](#)

Actualizado: Noviembre 16, 2005

sitio web: www.kidsprogramminglanguage.com

Empezando a Programar con KPL

Tabla de Contenido

Introducción: ¿Qué es la Programación?	3
¿Por qué aprender a programar con KPL?.....	3
¿Cómo debo utilizar este tutorial?.....	3
Está bien, ¡Muéstrame un Programa!	5
¡¿Qué pasa con las Gráficas de la Computadora?!.....	7
Juegos Gráficos Utilizando Sprites	12
Utilizando Variables y Ciclos en KPL	14
Utilizando KPL en Tu Computadora	18
Siguientes Pasos Después de Este Tutorial	26

Introducción: ¿Qué es la Programación?

La programación de computadoras es, de manera simple: **darle instrucciones a una computadora.**

Las computadoras son especialistas en seguir instrucciones. Ellas realizar exactamente lo que nosotros les decimos que hagan. ¡Pero ellas **no tienen imaginación!** Y por lo tanto, cuando escribimos programas para darle instrucciones a las computadoras, debemos decirles de manera muy precisa qué es lo que queremos que hagan.

¿Por qué aprender a programar con KPL?

Los diferentes lenguajes de las computadoras te dan a ti, el programador, diferentes maneras de decirle a tu computadora lo que quieres que haga. El Lenguaje de Programación para Niños (por sus siglas en inglés, Kid's Programming Language), ó KPL, tiene grandes ventajas para ti que apenas estás empezando:

- KPL fue diseñado cuidadosamente para hacer **lo más fácil posible** el aprendizaje para un principiante.
- KPL fue diseñado cuidadosamente para hacer **lo más divertido posible** tu aprendizaje.
- KPL, a diferencia de otros lenguajes, también fue diseñado cuidadosamente para ser **lo más similar posible** a los lenguajes utilizados actualmente por los programadores profesionales.

La gente sigue utilizando algunos dichos antiguos al hablar, porque son ciertos. La frase sabia que debemos recordar al hablar de KPL es: **aprendamos a caminar antes de correr.** Programar con KPL, es como aprender a caminar. Después de aprender KPL, será **mucho** más fácil para ti aprender a correr – ya sea que decidas correr con Java, Python, Visual Basic ó C#.

¿Cómo debo utilizar este tutorial?

Nota Muy Importante: Este tutorial fue escrito con base en las mejoras efectuadas a KPL que se encuentran en las versiones de KPL liberadas a partir del 10 de Octubre de 2005. Si usted descargó KPL antes del 10 de Octubre, o si los ejemplos no funcionan de la manera descrita, por favor, descargue e instale la última versión de KPL desde <http://www.kidsprogramminglanguage.com/espdownload.htm>

Si eres un verdadero principiante, y nunca antes has hecho programación, la mejor manera de utilizar este tutorial es leerlo y estudiarlo siguiendo sección por sección, en orden, asegurándote de haber entendido cada lección antes de pasar a la siguiente. Probablemente, sea mejor estudiar este tutorial sin utilizar KPL en la computadora, hasta haber llegado a la sección de este tutorial llamada **“Usando KPL en tu computadora”**. Estudiar y aprender KPL de esta manera evitará que te distraigas con los detalles, de los cuales habrá que ocuparse al final de este tutorial.

La programación de computadoras requiere que aprendamos a pensar en una forma en la que la gente no piensa normalmente. ¡Las computadoras necesitan que nosotros seamos mucho más lógicos, ordenados y precisos de lo que normalmente necesitamos ser! Aprender esta forma de

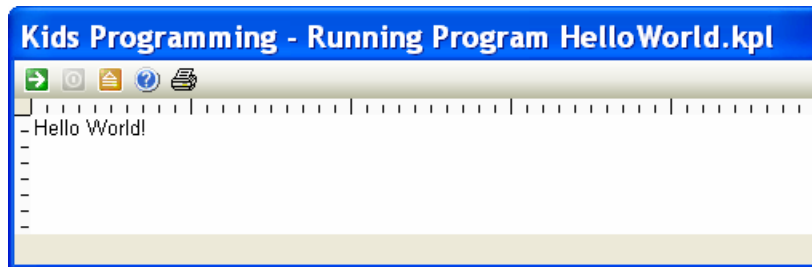
pensar puede ser difícil al principio, pero ¡tú puedes hacerlo! Y cuando realmente lo hayas entendido, notarás que se convierte en algo muy sencillo.

Una de las mejores maneras para aprender esta nueva forma de pensar es hacerle preguntas y escuchar las explicaciones de alguien que ya entienda la programación de computadoras. ¿Conoces a alguien que pueda ayudarte con preguntas y respuestas, mientras vas estudiando KPL a través de este tutorial?

Está bien, ¡Muéstrame un Programa!

```
1 Program HelloWorld
2   Method Main()
3     Print("Hello World!")
4   End Method
5 End Program
```

Bueno, ¡eso es todo! Cuando ejecutas este programa KPL, lo que verás es lo siguiente:



Las 5 líneas de código KPL que ves arriba son una imagen de cómo se ve el código de KPL, así puedes verlo exactamente de la misma forma como lo verás al estar programando con KPL.

Lo primero que debo mencionar es que los números de línea que ves al lado izquierdo de cada línea de código, no se utilizan dentro de KPL – solo están ahí como una información para ayudarte mientras lees y trabajas con los programas en KPL. Algunos lenguajes antiguos, como GWBASIC, realmente usaban los números de línea para procesar el código en el orden deseado por el programador. KPL no lo hace de esta manera. Los programas KPL normalmente se procesan instrucción por instrucción, comenzando al inicio y avanzando hacia abajo a través del programa. Existen algunas excepciones a esta regla, pero no las discutiremos en este tutorial para principiantes.

Una regla importante para programar en KPL es que cada línea de código necesitar estar escrita en una línea dentro del archivo de tu programa. Por ejemplo, este programa KPL tiene el mismo código que el que viste anteriormente, pero no está escrito en líneas separadas, por lo tanto este programa no funcionará:

```
1 Program HelloWorld Method Main() Print("Hello World!") End Method End Program
```

Todos los lenguajes de computadora tienen algunas reglas que los programadores deben respetar, para que la computadora pueda entender las instrucciones. La comunicación Persona-a-Persona también tiene muchas reglas, esas reglas tienen el mismo propósito de ayudar a entendernos – lo que pasa es que estamos tan acostumbrados a esas reglas que las respetamos, sin tener que detenernos a pensar en ellas. Por ejemplo, ¿no decimos “Adiós” cuando contestamos el teléfono! ¡Y tampoco decimos “Hola” cuando lo colgamos! Ya lo sé, es un ejemplo bastante tonto, pero es muy importante, porque KPL también necesita de una forma específica para iniciar y terminar un programa. Todos los programas KPL deben iniciar con la línea como esta: **Program HelloWorld** como puedes ver en la línea 1 de nuestro programa. Y todos los programas KPL deben terminar con **End Program** como lo ves en la línea 5:

```

1 Program HelloWorld
2   Method Main()
3     Print("Hello World!")
4   End Method
5 End Program

```

Tú puedes darle a tu programa el nombre que desees, pero es mejor si le das un nombre que describa lo que el programa hace. En este caso **HelloWorld** es el nombre que elegimos. Tú puedes fácilmente utilizar un nombre diferente, como **Program MiPrimerPrograma**.

Method Main() es el siguiente elemento importante que debes conocer de la programación KPL. Todos los programas KPL comienzan procesando la primera instrucción que se encuentre dentro del **Method Main()**. Echa un vistazo más abajo y verás que esa instrucción es **Print("Hello World!")**.

Method Main() es una manera un poco arbitraria de definir el lugar donde el programa comienza a procesar instrucciones – pero esto está basado, en realidad, en la manera como funcionan todos los lenguajes modernos de programación. Como puedes lógicamente imaginarte, **End Method** corresponde y determina el final del **Method Main()**.

No explicaremos más sobre los métodos en este tutorial, pero por ahora, espero que sea lo suficientemente claro que al mirar el código, te des cuenta de que solamente existe una instrucción “dentro de” el método principal del programa **Method Main()**, y que se trata de la línea que dice **Print("Hello World!")**.

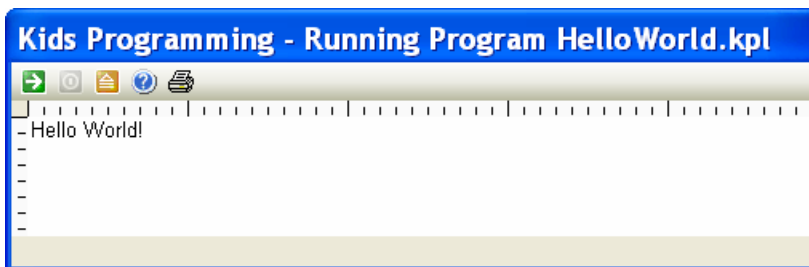
Todo esto fue una explicación detallada de por qué razón este programa solamente contiene una instrucción, se trata de la que le dice a la computadora que debe mostrar la frase “Hello World!” en la pantalla y que dice: **Print("Hello World!")**. Así que, en resumen: Las líneas 1 y 5 le dicen a la computadora que estamos empezando y terminando este programa KPL. Y las líneas 2 y 4 le dicen a la computadora que ahí inicia y termina el **Method Main()**.

```

1 Program HelloWorld
2   Method Main()
3     Print("Hello World!")
4   End Method
5 End Program

```

Ahora miremos nuevamente la ventana que aparece cuando ejecutamos este programa. Fíjate que “dentro” de esa ventana, la computadora ha hecho una sola cosa, de hecho, solamente hizo lo que le dijimos que hiciera con este programa KPL. La computadora ha mostrado las palabras **Hello World!**



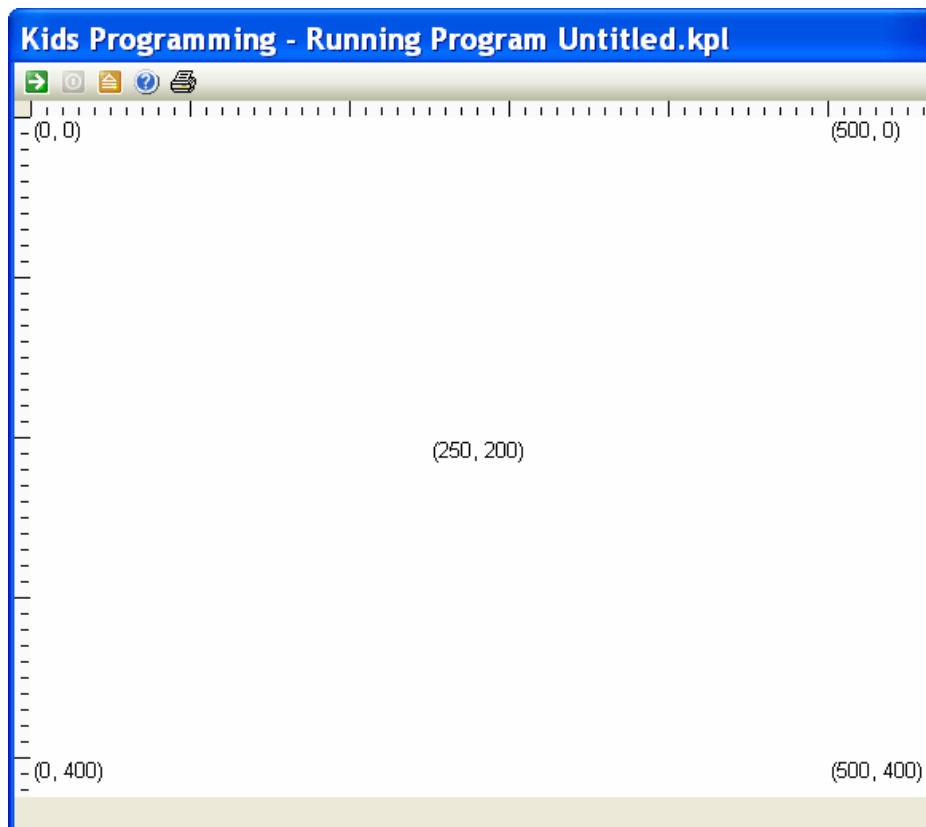
¿Qué pasa con las Gráficas de la Computadora?!

Hello World! es un primer programa clásico – ¡pero no es muy emocionante! Así que hablemos un poco sobre los gráficos.

Una explicación y ejemplo sobre los gráficos de una computadora, tiene que empezar mencionando dónde es que las computadoras colocan los gráficos que se muestran en la pantalla. Las computadoras utilizan un **sistema de coordenadas** que es diferente al sistema de coordenadas que has estudiado en el **álgebra** que todos estudiamos en la escuela. Pero en realidad, el sistema de coordenadas de la computadora puede ser más fácil de utilizar, sobre todo porque hace más fácil el trabajo con las ubicaciones dentro de la pantalla de la computadora.

Las computadoras utilizan un sistema de coordenadas (**X, Y**) para ubicar las posiciones en la pantalla, donde el **borde izquierdo de la pantalla indica que $X = 0$** , y **el borde superior de la pantalla indica que $Y = 0$** . Esto significa que el **origen**, donde **$X = 0$ y $Y = 0$** , es la **esquina superior izquierda** de la pantalla. El movimiento hacia la derecha de la pantalla incrementará el valor de la X, el movimiento hacia abajo en la pantalla incrementará el valor de la Y.

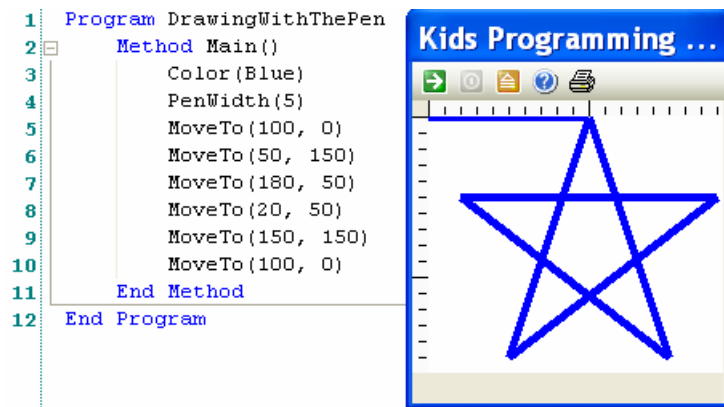
Si no estás acostumbrado a los sistemas de coordenadas, esta puede ser una explicación un poco complicada, por eso a continuación puedes ver una imagen que te indica dónde se encuentran varias posiciones (**X, Y**) desplegadas en la pantalla por un programa KPL:



Por favor analiza un momento los números que aparecen en la imagen. El **primer valor** de cada pareja es el **valor de X** para esa posición en la pantalla – y como puedes verlo, el valor de X crece

conforme te mueves de izquierda a derecha en la pantalla. El **Segundo valor** de cada pareja es el **valor de Y** para esa posición en la pantalla – y como puedes verlo, el valor de Y crece conforme te mueves de arriba hacia abajo en la pantalla.

Ahora escribamos nuestro primer programa KPL de gráficos, en el proceso verás exactamente como usar el sistema de coordenadas gráficas para realizar imágenes interesantes en la pantalla usando KPL. Empezaremos mostrando el programa KPL completo, acompañado de las gráficas que crea cuando lo ejecutas. Y después analizaremos cada paso del programa KPL en detalle, para explicarte exactamente cómo funciona:

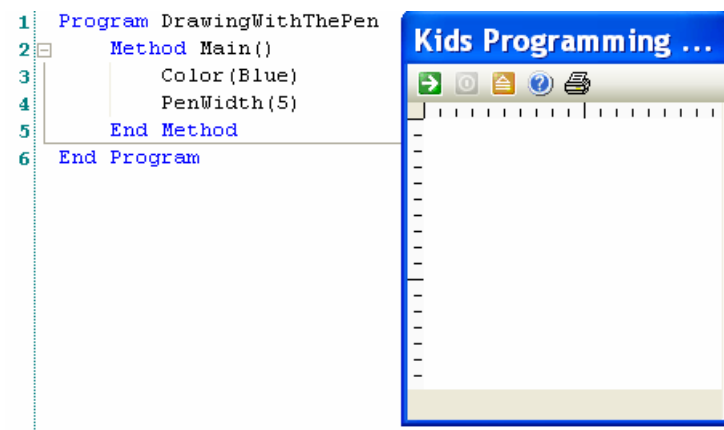


Puedes darte cuenta de que ahora tenemos 8 instrucciones KPL dentro de **Method Main()**, desde la línea 3 hasta la línea 10. Esto es mucho más que en nuestro primer ejemplo, pero seguramente estarás de acuerdo en que ¡es muy fácil poder dibujar con KPL una estrella azul en la pantalla de la computadora utilizando tan sólo 8 instrucciones!

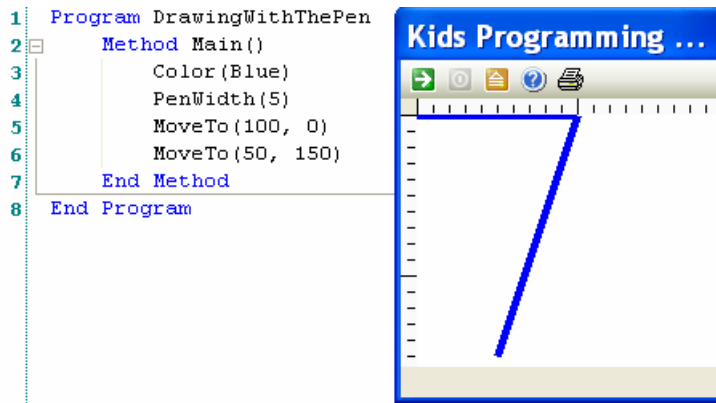
Fíjate que el programa comienza y termina en la misma forma que nuestro primer programa, excepto que ahora hemos llamado a este programa **DrawingWithThePen**:

```
1 Program DrawingWithThePen
2   Method Main()
3
4   End Method
5 End Program
```

Ahora agreguemos nuestras instrucciones al **Method Main()**, para revisar qué hace cada una:



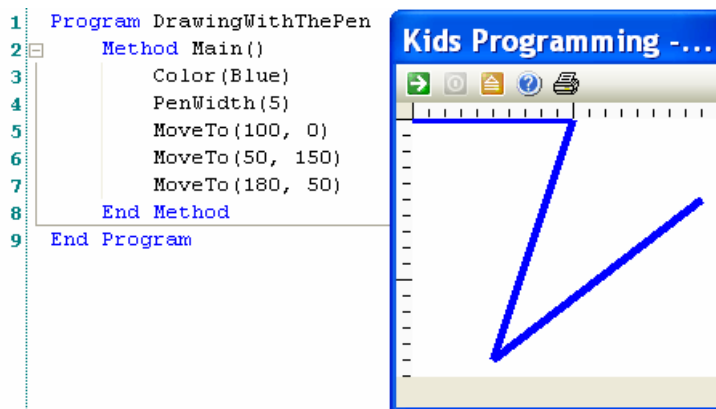
Hemos agregado las primeras dos instrucciones a nuestro programa, pero como puedes notar, el programa aún no muestra ningún gráfico. **Color(Blue)** le dice a KPL que queremos que empiece a dibujar utilizando el color Azul (Blue). **PenWidth(5)** indica que cuando dibujemos con la pluma (Pen), queremos que KPL dibuje una línea de 5 píxeles de ancho. Por supuesto, tú puedes utilizar un valor diferente para el ancho de la pluma. **PenWidth(2)** dibujaría una línea más delgada, y **PenWidth(10)** dibujaría una línea más gruesa. Hasta ahora, le hemos dicho a KPL cómo queremos que dibuje, pero aún no le decimos que dibuje nada. Ahora, dibujemos la primera línea de nuestra estrella:



La primera instrucción que agregamos es **MoveTo(100, 0)**. Esto le dice a KPL que mueva la pluma a la posición (100, 0) de la pantalla, que indicará la punta superior de nuestra estrella. Debido a que KPL siempre comienza con la pluma en la posición (0, 0) – la esquina superior izquierda – cuando movemos la pluma a la posición (100, 0) estamos dibujando la línea horizontal que puedes ver en el borde superior de la pantalla.

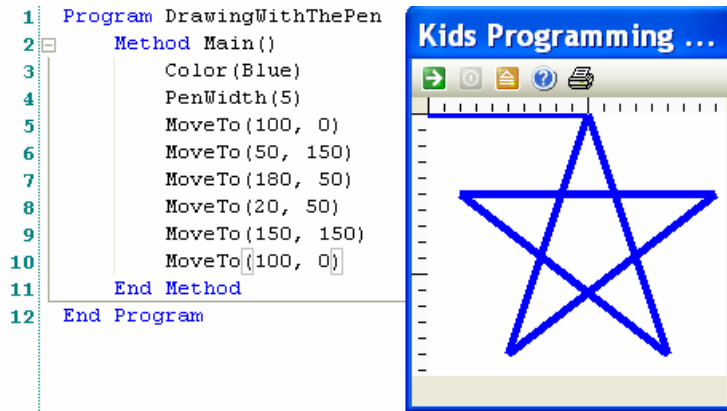
La segunda instrucción que agregamos es **MoveTo(50, 150)**. Esta instrucción dibuja la primera línea de nuestra estrella. Consideremos los valores (X, Y) de estas dos posiciones por el momento. Debido a que la instrucción le dice a KPL que mueva la pluma desde X = 100 hacia X = 50, la línea se moverá hacia la izquierda. Y debido a que también nos movemos desde Y = 0 hacia Y = 150, la línea también se mueve hacia abajo en la pantalla.

A continuación agreguemos otra instrucción a KPL, para dibujar la siguiente línea de la estrella:



La instrucción que agregamos es **MoveTo(180, 50)**. KPL sigue moviendo la pluma desde el punto anterior, que era (50, 150). ¿Has visto esos juguetes que te permiten dibujar cosas en la

pantalla solamente girando dos perillas? La pluma de KPL es muy similar a un juego computarizado que funciona de la misma manera. Agreguemos ahora las líneas que faltan para terminar de dibujar nuestra estrella:



Como puedes ver, esas tres nuevas instrucciones hacen que KPL dibuje tres líneas más en la pantalla. En total, hemos movido la pluma 6 veces, dibujando seis líneas, y el resultado es una estrella, dibujada por medio de las instrucciones en nuestro programa KPL.

El programa KPL necesario para lograr esto es pequeño – solamente 8 instrucciones KPL. Lo más difícil de este programa es acostumbrarnos a la manera como funciona el sistema de coordenadas (X, Y) dentro de KPL. Si hasta este momento, los valores (X, Y) aún no te quedan claros, probablemente deberás regresar al comienzo de esta sección y leerla una vez más. Si después de hacerlo, las coordenadas (X, Y) aún no te quedan claras, tal vez desees escribir tu propia versión de este programa KPL en tu computadora, y experimentar con el dibujando líneas desde y hacia diferentes posiciones de la pantalla. ¿Quizás pudieras practicar dibujando un cuadrado y un triángulo? Si logras hacer esto con KPL en tu computadora, podrás saltar hasta la sección **Utilizando KPL en Tu Computadora**, para obtener ayuda al utiliza KPL en tu computadora. Cuando te sientas seguro utilizando las coordenadas (X, Y), podrás regresar a este punto del tutorial.

Es importante que dediques el tiempo y esfuerzo necesario hasta que te sientas cómodo con la forma como KPL maneja las coordenadas (X, Y), ya que esta es la base para toda la programación de gráficos en computadora. Otros lenguajes manejan las gráficas de la misma manera, así que al aprender a utilizarlas con KPL, en realidad estás aprendiendo los conceptos fundamentales para poder programar gráficos de computadora dentro de cualquier otro lenguaje.

Agregaremos ahora un pequeño detalle final a nuestro ejemplo, que también te dará un poco más del control que puedes lograr sobre la pluma de KPL. ¡Esto es algo que no podrás lograr con los juguetes que mencionábamos antes! ¡Tampoco puedes controlar el color con que esos juguetes dibujan!, ahora te mostraremos que fácil es hacerlo con KPL:

```

1 Program DrawingWithThePen
2 Method Main()
3     Color (Green)
4     PenWidth(5)
5     Pen(False)
6     MoveTo(100, 0)
7     Pen(True)
8     MoveTo(50, 150)
9     MoveTo(180, 50)
10    MoveTo(20, 50)
11    MoveTo(150, 150)
12    MoveTo(100, 0)
13 End Method
14 End Program

```



Como viste, solamente cambiamos la instrucción **Color(Blue)** por **Color(Green)**. ¡Qué fácil! También, agregamos como primera instrucción **Pen(False)**, justo antes de **MoveTo(100, 0)**. **Pen(False)** le dice de una manera simple a KPL que no queremos que dibuje con la pluma cuando nos movemos de posición. Así que, ahora cuando la pluma se mueve de (0, 0) a (100, 0), ya no dibuja la línea horizontal que aparecía arriba de nuestra estrella – ¡así nuestra estrella se ve mucho mejor!

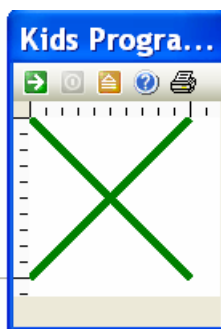
Recuerda que las computadoras necesitan instrucciones muy específicas, y cuando dijimos **Pen(False)**, KPL dejó de dibujar conforme la pluma se movía. Así que ¿qué necesitamos hacer ahora para que KPL dibuje las líneas que forman nuestra estrella? Pues, necesitamos darle a KPL la instrucción: **Pen(True)**, de esta manera KPL empezará a dibujar nuevamente conforme movamos la pluma.

En resumen, **Pen(False)** le dice a KPL “no dibujes” conforme se mueve la pluma, y **Pen(True)** le dice a KPL “dibuja” al mover la pluma. ¡Esto tampoco se puede hacer con el juguete que conoces! Encender y apagar la pluma cada que la mueves te permitirá dibujar todo tipo de objetos en la pantalla, también te permitirá dibujar varios objetos. Aquí está un ejemplo de lo que puedes dibujar cuando apagas la pluma de KPL, la mueves y la vuelves a encender después:

```

1 Program DrawingWithThePen
2 Method Main()
3     Color (Green)
4     PenWidth(5)
5     MoveTo(100, 100)
6     Pen(False)
7     MoveTo(100, 0)
8     Pen(True)
9     MoveTo(0, 100)
10 End Method
11 End Program

```




Juegos Gráficos Utilizando Sprites

El objetivo del ejemplo de dibujo anterior era explicarte cómo se utilizan las coordenadas (X, Y) para desplegar gráficos en la computadora. Pero los juegos gráficos no son dibujados de forma normal. Estos se cargan desde un archivo de imagen, como la imagen de un platillo volador, de un asteroide ó de un duende con un arco. ¡KPL utiliza los Sprites para hacerlo de la forma más fácil!

Como lo hicimos en nuestro último programa, empezaremos mostrando el ejemplo terminado y después explicaremos en detalle cómo es que el programa funciona, para que después puedas escribir otros programas tú mismo.

Este es el programa KPL complete – que si te fijas dentro del **Method Main()** tiene exactamente 8 instrucciones, igual que en nuestro ejemplo anterior. Estas 8 instrucciones, sin embargo, le dicen a KPL que muestre un platillo volador en la parte superior de la pantalla y que lo mueva lentamente hacia abajo hasta la posición donde lo puedes ver a continuación:

```
1 Program UFO
2
3 Method Main()
4
5     LoadSprite( "UFO", "UFO.GIF" )
6     MoveSpriteToPoint( "UFO", 50, 0 )
7     ShowSprite( "UFO" )
8
9     Define ufoY As Int
10    For ufoY = 1 To 150
11        Delay(10)
12        MoveSpriteToPoint( "UFO", 50, ufoY )
13    Next
14
15 End Method
16
17 End Program
```



Nota que el programa comienza y termina de la misma forma que nuestros ejemplos anteriores – y de la misma forma que todos los programas KPL excepto que a este lo hemos llamado **UFO**:

```
1 Program UFO
2
3 Method Main()
4
5 End Method
6
7 End Program
```

Déjame mostrarte algunos ejemplos de archivos de imágenes y fotografías que se incluyen con KPL. Tú puedes utilizar **CUALQUIER** archivo de imagen en KPL, hasta los que tú agregues o diseños – estos son solo una muestra de las 65 imágenes incluidas en KPL:



UFO.GIF



QUAD.GIF



PLANE.PNG

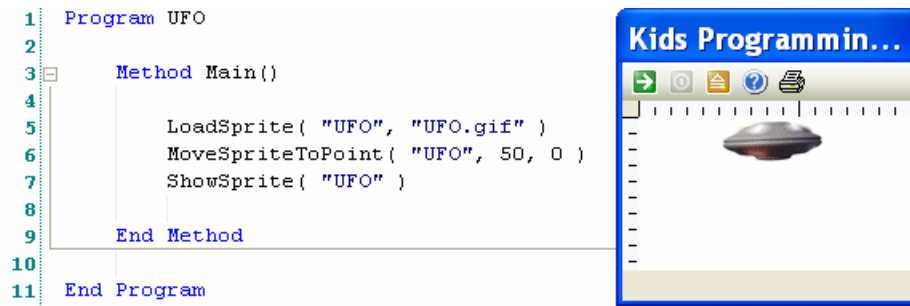


SPIDER.PNG



BLUEBALL.PNG

Ahora que has visto un ejemplo de los tipos de gráficos que puedes utilizar con KPL, empecemos a agregar las instrucciones KPL que harán que el platillo volador se muestre en la parte superior de la pantalla:



La primera instrucción que agregamos es `LoadSprite("UFO", "UFO.gif")`. Como lo mencionamos antes, KPL tiene algunas reglas sobre la manera como le damos las instrucciones. Las reglas para la instrucción `LoadSprite` no son difíciles, ¡pero si son precisas! `LoadSprite` requiere de dos “valores” para poder trabajar. El primer valor debe ser el nombre con el que haremos referencia al Sprite – en nuestro caso, nos referiremos a él como `"UFO"`. Y el segundo valor debe ser el nombre del archivo de imagen que KPL utilizará para este sprite– en nuestro caso, el nombre del archivo es `"UFO.gif"`. Estos valores deben estar encerrados entre comillas y separados por una coma, como lo puedes ver.

Aquí te mostramos algunos ejemplos sobre formas **incorrectas** de indicarle a KPL que cargue este Sprite. Lo repito: **cada una de estas cuatro instrucciones a continuación, no funcionarán** porque no siguen las reglas de KPL definidas para referirse a `LoadSprite`. ¿Puedes identificar qué debes cambiar en cada una de ellas para que funcionen bien?

```
LoadSprite( 'UFO', 'UFO.gif' )
LoadSprite( UFO, UFO.gif )
LoadSprite( "UFO" . "UFO.gif" )
LoadSprite( "UFO" )
```

Resumen: `LoadSprite("UFO", "UFO.gif")` dice a KPL que haga un nuevo sprite usando la foto del archivo `"UFO.gif"` y que lo nombre `"UFO"`.

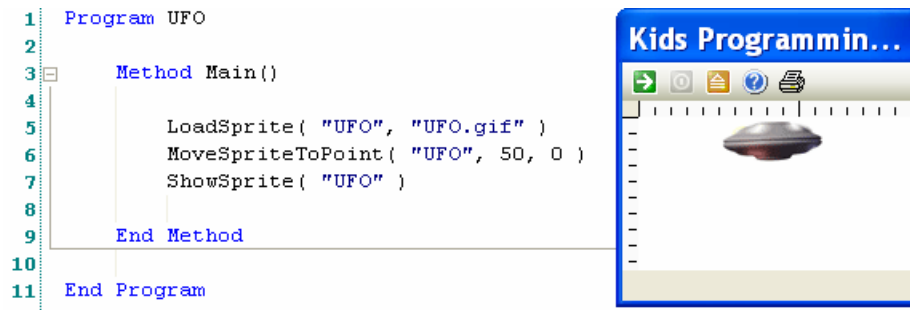
Después KPL necesita que le indiquemos dónde colocar el sprite dentro de la pantalla, así que le damos la instrucción `MoveSpriteToPoint("UFO", 50, 0)`. `MoveSpriteToPoint` requiere tres valores. El primero es el nombre con que hacemos referencia al sprite, que en este caso es `"UFO"` porque ese es el nombre que le dimos en `LoadSprite`. El segundo es la **posición en el eje X** para el sprite, que nosotros queremos que sea `50`. El tercero es la **posición en el eje Y** para el sprite, en este caso queremos que sea `0`.

Un detalle importante es que **no debemos colocar comillas alrededor de los valores numéricos**. En general, KPL **necesita** comillas alrededor de valores que sean **palabras**, KPL **no necesita** comillas alrededor de valores que sean **números**. Consulta la sección Tipos de Datos, en la **Guía del Usuario KPL** para obtener información más detallada sobre este tema.

Resumen: `MoveSpriteToPoint("UFO", 50, 0)` dice a KPL que mueva el sprite llamado `"UFO"` a la posición `(50, 0)` de la pantalla.

Lo que nos falta ahora es decirle a KPL que muestre el sprite. Esta es una instrucción más sencilla: **ShowSprite("UFO")**. ¡Eso es todo! ¡Nuestro platillo volador aparecerá en la pantalla en la posición en que dijimos a KPL que lo colocara!

Esa fue una explicación muy detallada, ahora veamos qué sencillo es en KPL. Este es el programa KPL y lo que verás en la pantalla cuando lo ejecutes:



Utilizando Variables y Ciclos en KPL

Ahora queremos que nuestro platillo volador "atterrice" moviéndolo hacia abajo en la pantalla hasta llegar a la parte baja de la ventana. Para lograr esto vamos a definir y utilizar nuestra primera **"variable"**, de esta forma veras cómo funcionan las variables en un programa KPL.

El término variable es muy descriptivo – significa que su valor puede cambiar en el tiempo. La posibilidad de cambiar su valor en diferentes momentos es el verdadero poder de utilizar una variable dentro de un programa, lo verás en este ejemplo.

Las variables deben ser definidas con un nombre que nos ayude a identificarlas y que nos diga qué es lo que vamos a hacer con ellas. En este caso, usaremos una variable para cambiar la posición de nuestro platillo volador en el eje X, así lograremos moverlo hacia abajo en la pantalla y lo haremos "atterrizarse", por eso vamos a conocer a nuestra variable como **ufoY**.

Esta es la línea de código que define nuestra variable:

```
Define ufoY As Int
```

Define es una palabra clave dentro de KPL que le dice que vamos a definir una nueva variable para poderla usar dentro de nuestro programa KPL. **ufoY** es el nombre que utilizaremos para referirnos a la variable. **As Int** le dice a KPL que la variable **ufoY** que estamos definiendo será una variable de tipo Entero. **Int** es la abreviación de Integer en inglés, que significa **Entero**. Las variables de tipo Entero pueden tomar valores numéricos, como **-1** ó **0** ó **43**. Las variables de tipo Int solo pueden tener números Enteros como valor.

Sabemos que nuestro platillo volador empieza en la posición **(50,0)** de la pantalla, porque ese es el punto donde le dijimos a KPL que lo moviera. ¿Cómo hacemos que se mueva hacia abajo a través de la pantalla? Eso lo logramos al ir incrementando **el valor de su posición en el eje Y** – a **(50, 1)** después a **(50, 2)** luego a **(50, 3)** después a **(50, 4)**, etc... Depende de nosotros que tan lejos vaya el platillo volador, así que decidamos que nuestro platillo volador de detenga al llegar a **(50, 150)**.

¿Te das cuenta del truco? El valor de posición del sprite en el eje **X** siempre es **50**. El valor de posición en el eje **Y** va creciendo en **1** píxel cada vez, desde **0** hasta **150**. Es importante que entiendas muy bien este ciclo antes de que te mostremos cómo poder utilizarlo en el código de KPL – así que si aún no te queda claro este ciclo de incremento, por favor lee nuevamente el párrafo anterior.

A continuación verás el código KPL que nos sirve para incrementar el valor de la variable **ufoY** desde 1 hasta 150, 1 número a la vez:

```
For ufoY = 1 To 150
    MoveSpriteToPoint( "UFO", 50, ufoY )
Next
```

Este es tu primer “ciclo”. Los ciclos son otro de los nuevos conceptos que necesitas aprender si es que nunca antes has programado – pero una vez que hayas entendido el concepto, te darás cuenta de que utilizar los ciclos es bastante sencillo.

Este ciclo empieza con el valor de **ufoY = 1**. Debido a que también especificamos **To 150**, podemos saber que el ciclo se detiene al momento de llegar al valor de **ufoY = 150**. Pon atención en la instrucción **Next**, ya que es la instrucción que define el final del ciclo. Cuando KPL ejecuta la instrucción **Next**, KPL sabe que es el momento de **incrementar el valor de ufoY** al siguiente valor que corresponda – de 1 a 2, ó de 2 a 3, ó de 3 a 4, etc..., hasta llegar a incrementarlo de 149 a 150.

Básicamente, le estamos diciendo a KPL que cuente de 1 a 150, y queremos que para llevar esta KPL utilice la variable ufoY para ir guardando el valor conforme hace esta cuenta.

¿Pero qué es lo que queremos que haga KPL con esta cuenta? Queremos que mueva al platillo volador hacia abajo en la pantalla, ¿cierto? La instrucción **MoveSpriteToPoint("UFO", 50, ufoY)** le dice a KPL lo que queremos que haga.

Como puedes ver, esta instrucción se encuentra “adentro” de la instrucción **For** de nuestro ciclo, por esta razón KPL ejecutará esta instrucción **cada vez que cuente** desde 1 hasta 150. Esta es la verdadera utilidad de los ciclos, por supuesto. Si solamente contáramos de 1 a 150 no estaríamos haciendo nada muy interesante, pero sí podemos **hacer algo útil e interesante cada vez que contamos**, esto nos permite lograr **toda clase de cosas**, ¡incluso mover nuestro platillo volador a través de la pantalla! Ahora examinemos qué es lo que pasa:

```
For ufoY = 1 To 150
    MoveSpriteToPoint( "UFO", 50, ufoY )
Next
```

Ya hemos platicado sobre cómo funciona la instrucción **MoveSpriteToPoint**: KPL mueve el sprite llamado **"UFO"** a la posición (**X, Y**) que le indicamos. ¿Qué es diferente en esta ocasión? Anteriormente, movimos el platillo volador a la posición (**50, 0**). ¿Qué pasará cuando utilicemos esta instrucción dentro del ciclo, y en su lugar utilicemos nuestra variable **ufoY**? Recuerda que la primera vez que entramos en el ciclo, el valor de **ufoY = 1**, y la segunda vez que entremos al ciclo **ufoY = 2**, después **ufoY = 3**, en la siguiente **ufoY = 4**, etc... así hasta que **ufoY = 150**. De esta manera, al entrar la primera vez a nuestro ciclo, KPL utiliza para mover el sprite un valor de **ufoY=1**, así que en realidad la instrucción que KPL está ejecutando en nuestro programa es:

```
MoveSpriteToPoint( "UFO", 50, 1 )
```


La siguiente ocasión que entramos al ciclo, `ufoY = 2`, así que KPL ejecuta:

```
MoveSpriteToPoint( "UFO", 50, 2 )
```

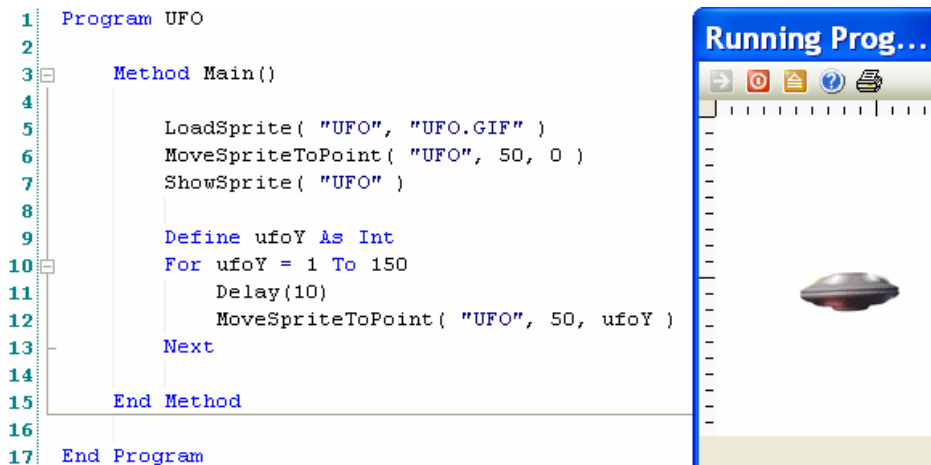
Y así de forma similar, hasta que KPL ha utilizado el ciclo para `ufoY` hasta 150:

```
MoveSpriteToPoint( "UFO", 50, 3 )
MoveSpriteToPoint( "UFO", 50, 4 )
MoveSpriteToPoint( "UFO", 50, 5 )
...
MoveSpriteToPoint( "UFO", 50, 150 )
```

En cada vez el sprite se mueve 1 píxel hacia abajo en nuestra pantalla- ¡justo como queríamos que pasara!

Esta es tu primera variable, tu primer ciclo, y tu primera instrucción dentro de un ciclo - ¡estos son conceptos **muy importantes** para la programación! Si aún no te quedan muy claros, por favor regresa al inicio de esta sección y revísala nuevamente.

Agreguemos un último detalle para terminar con nuestro programa:



La única instrucción que agregamos, como puedes verlo es `Delay (10)`, dentro de nuestro ciclo `For`. ¿Para qué hicimos esto? ¡Pues porque las computadoras cuentan muy muy muy rápido! Contar desde 1 hasta 150 le toma menos tiempo a una computadora que lo que a nosotros nos toma abrir y cerrar nuestros ojos. ¡En serio! ¡Las computadoras son **tan** rápidas! Y si de verdad queremos ver que nuestro platillo volador se mueve hacia abajo a través de la pantalla, tenemos que decirle a la computadora que no cuente tan rápidamente como ella puede hacerlo de forma normal. `Delay (10)` le dice a KPL de una manera muy simple que puede tomar un pequeño descanso antes de mover cada vez al platillo volador.

Es lo que hacemos por ejemplo cuando queremos contar los segundos de un reloj, acostumbramos decir "un mil uno, un mil dos, un mil tres" para hacer un poco más lenta nuestra cuenta. Esto es exactamente lo que hace KPL cuando le damos la instrucción `Delay (10)` ya que de esta forma estamos haciendo un poco más lenta la cuenta. Cuando trabajes en este programa dentro de KPL, intenta cambiar el `10` a otros valores y después ejecuta el programa.

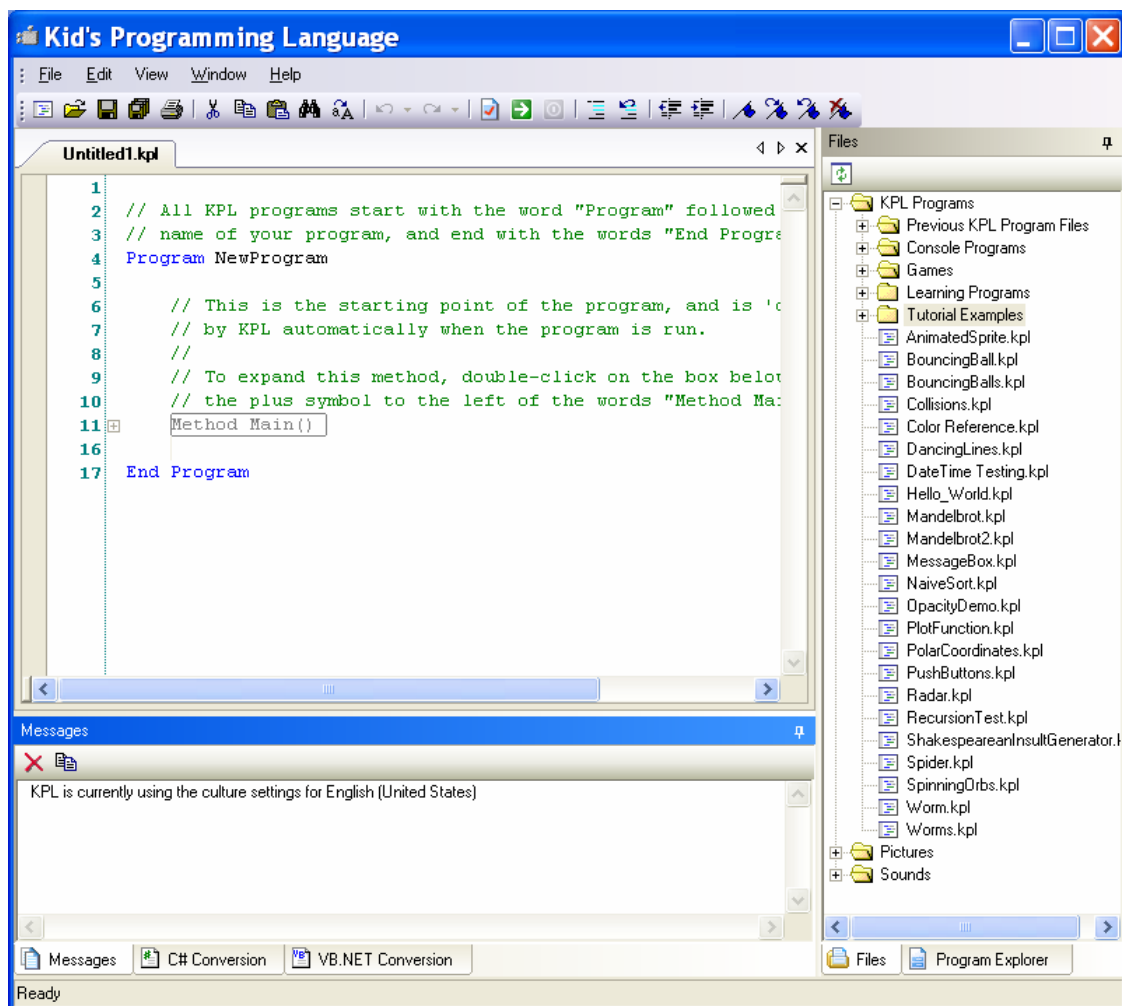
Puedes intentar con **Delay (2)** y después intenta con **Delay (100)**. ¿Ves la gran diferencia que logras en el movimiento de nuestro platillo volador?

Hemos dedicado varias páginas a explicar detalladamente este programa, lo que tal vez haga que te parezca más complicado de lo que es en realidad. Pero una vez que te acostumbras a programar con KPL, teclear estas 8 instrucciones no será un gran esfuerzo para ti – y esperamos que estés de acuerdo en que ¡necesitas muy pocas instrucciones para lograr increíbles programas de gráficas con KPL!

Utilizando KPL en Tu Computadora

Nota Muy Importante: Este tutorial fue escrito con base en las mejoras efectuadas a KPL que se encuentran en las versiones de KPL liberadas a partir del 10 de Octubre de 2005. Si usted descargó KPL antes del 10 de Octubre, o si los ejemplos no funcionan de la manera descrita, por favor, descargue e instale la última versión de KPL desde <http://www.kidsprogramminglanguage.com/espdownload.htm>

Esta sección supone que KPL ya está actualmente instalado, y se enfoca en mostrarte cómo empezar a utilizar KPL. Localiza y arranca KPL desde el grupo de programas llamado “Kids Programming Language” dentro del menú Inicio de Windows en tu computadora. Al arrancar KPL, verás una pantalla similar a esta:

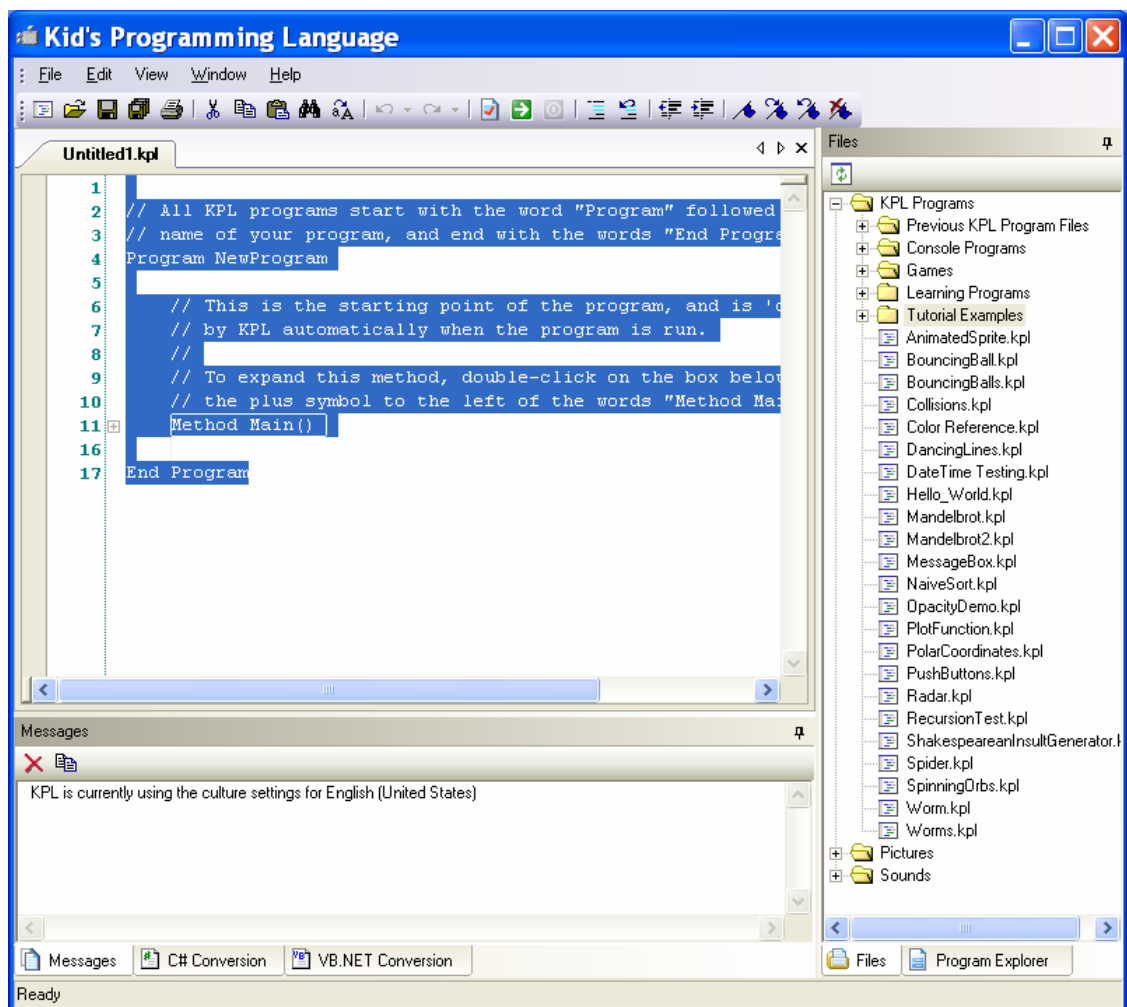


KPL muestra inmediatamente un Nuevo programa KPL llamado Untitled1.kpl en la ventana del Editor de Código (Code Editor). Las líneas coloreadas en verde que puedes ver en el editor de código son “comentarios” que les proporcionan información a las personas que leen el programa, pero que en realidad no son instrucciones de KPL. Los comentarios los indicamos empezando la línea con dos líneas diagonales: //

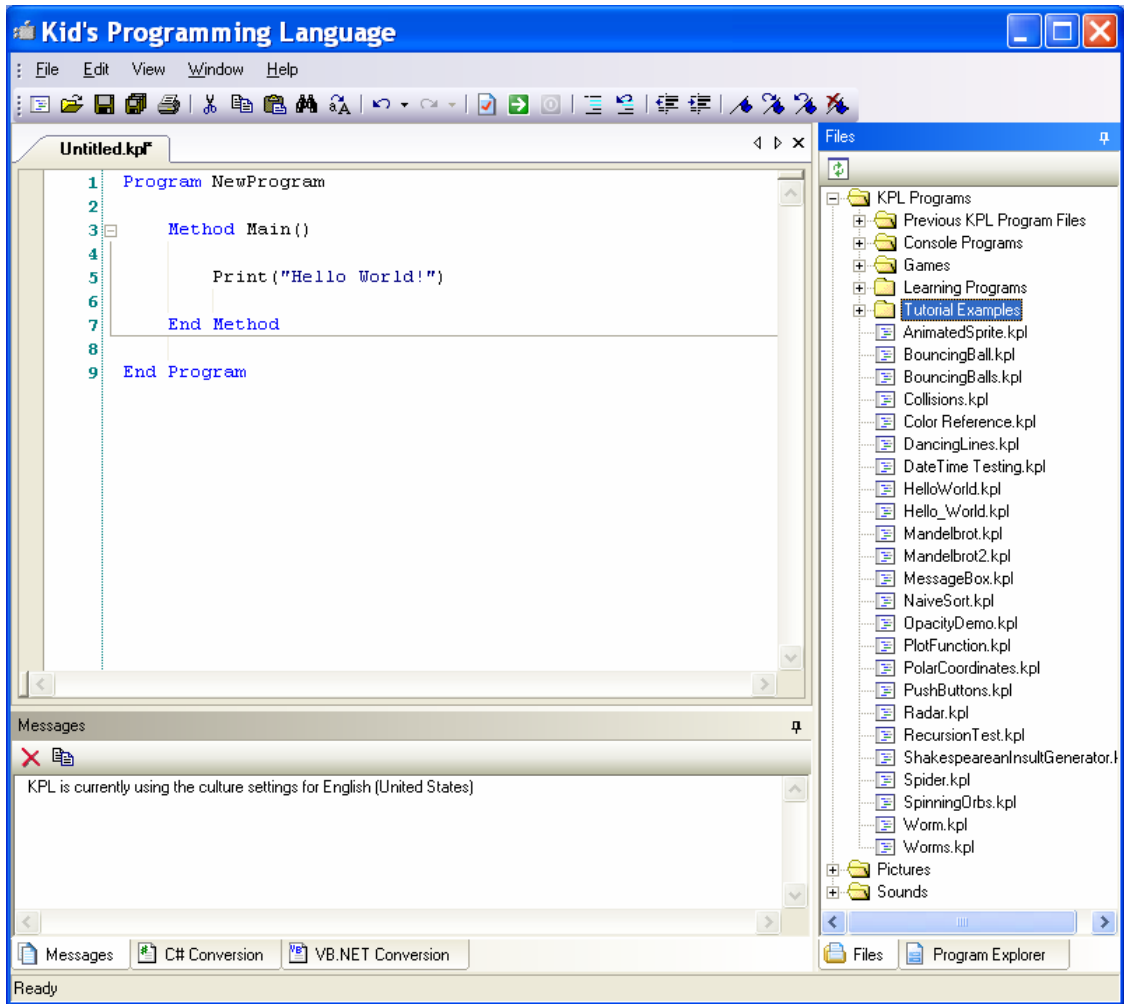
Encontrarás una gran cantidad de comentarios en los programas de ejemplo de KPL, y poco a poco irás aprendiendo a utilizar los comentarios, conforme vayas escribiendo tus propios programas KPL. Los comentarios ayudan a los demás a entender lo que hace el programa – ¡hasta pueden ayudarte a recordar lo que hacía un programa que vuelves a ver, pero que habías escrito hace mucho tiempo!

El editor de código de KPL tiene muchas funciones similares a las que existen en tu programa de edición de textos ó de correo electrónico. Tómate un momento para examinar las funciones del menú y de la barra de herramientas. Si colocas el ratón de tu computadora sobre los íconos de la barra de herramientas, podrás ver los mensajes de ayuda que aparecen para identificarlos.


Queremos que te enfoques en reproducir los tres programas ejemplo que creamos anteriormente en este mismo tutorial, así que empecemos borrando todo el código que se muestra en el programa llamado Untitled1.kpl. Para lograr esto, da clic en el menú Edición (**Edit**), y después selecciona la opción Seleccionar todo (**Select All**) que aparece en el menú. Verás que todas las líneas de código dentro del editor de código son seleccionadas como se muestra a continuación. Cuando todo el código esté seleccionado de esta manera, puedes presionar la tecla “Eliminar” (**Delete**) en el teclado de tu computadora, o seleccionar el menú Edición (**Edit**) y después el comando Cortar (**Cut**), y de esta manera todo el código que aparecía en la pantalla se eliminará del editor de código de KPL.

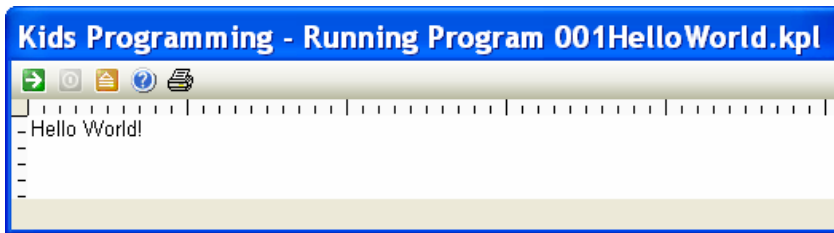


Cuando hayas eliminado el código, podrás empezar a escribir tu primer programa de ejemplo en KPL, como lo verás a continuación:



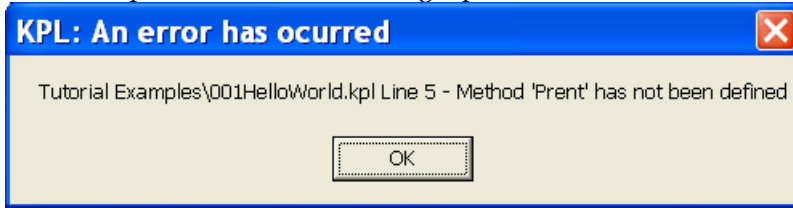
Recuerda que las computadoras son muy precisas – así que por favor teclea el programa KPL exactamente como se muestra en la imagen de arriba. Puede ser que trabaje correctamente o no si es que lo escribieras diferente. Puedes utilizar la tecla **TAB** para lograr el acomodo del código KPL como se muestra, a estos diferentes espacios se les llama **indentación**. Puedes presionar la tecla **ENTER** (Return) en una línea para dejarla vacía como se muestra en la imagen. No es necesario utilizar la indentación ni las líneas en blanco dentro de tu programa, pero al hacerlo te ayudas a ti y a los demás personas para que poder leer y entender más fácilmente tu programa de KPL.

Después que hayas terminado de escribir tu programa, da clic en el botón que tiene el icono de una flecha verde  o presiona la tecla **F5** para empezar a ejecutar tu programa KPL. Si escribiste correctamente tu código KPL, a continuación deberá aparecer una ventana muy similar a la que verás a continuación, excepto porque la ventana que aparecerá en tu pantalla será un poco más grande:



¡Eso es todo! ¡Ya eres un programador de computadoras! ¡Qué tal! ¡Yupiiii! Está bien, está bien, es un programa muy pequeño y sencillo, pero de todas maneras – ¡tú lo escribiste y lograste hacerlo funcionar!

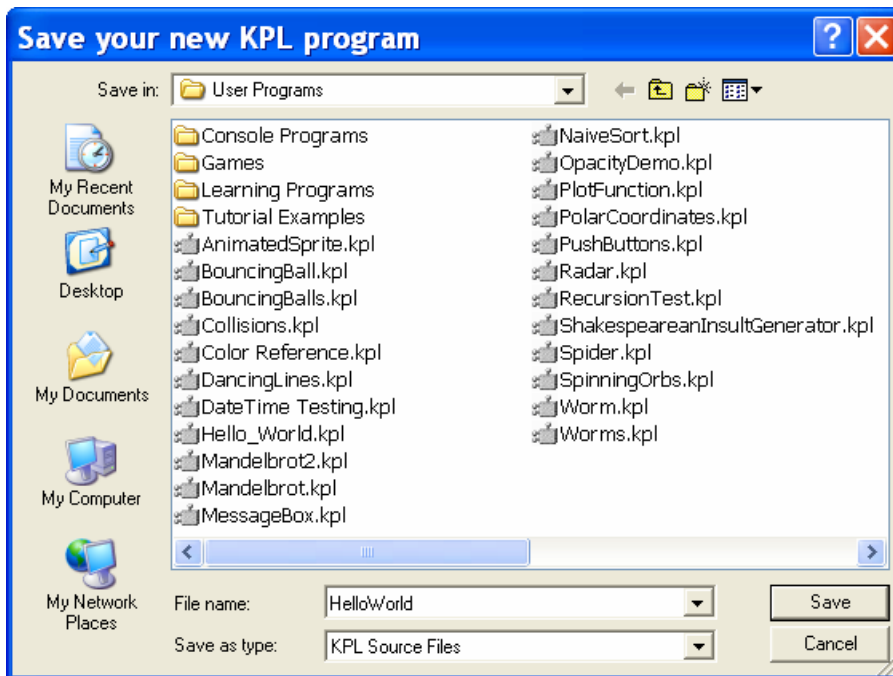
Si KPL no pudiera entender el código que escribiste, verás un mensaje de Error como este:



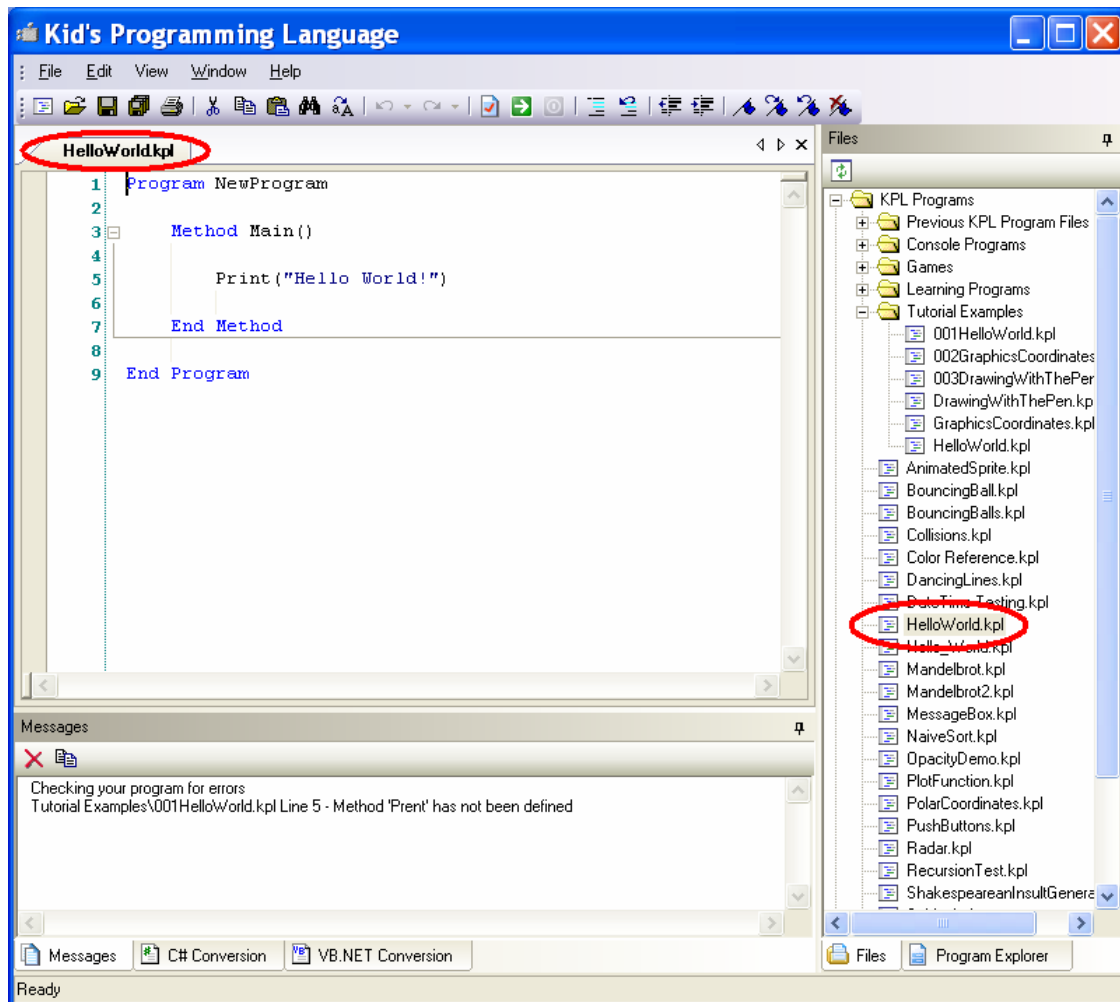
Si esto llegara a pasar, por favor dale clic al botón de OK, después revisa cuidadosamente el código que escribiste y compáralo con el que mostramos en la imagen anterior del editor de código. Cuando localices la diferencia entre el código, cámbiala para que quede igual al ejemplo y en ese momento podrás ejecutar correctamente el programa.

Todos los programadores de computadoras tienen a veces errores en su código, así que no te desanimes ni te sientas mal cuando te ocurra a ti. ¡Nadie es perfecto! Cuando esto pase con tu código, mantén la calma y la paciencia, sé muy cuidadoso para encontrar lo que está mal y corrígelo.

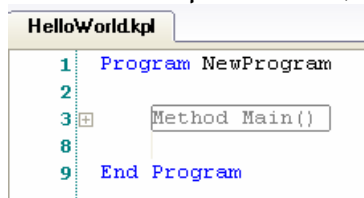
Cuando tengas el ejemplo HelloWorld funcionando correctamente, da clic en el menú Archivo (**File**), selecciona la opción Guardar (**Save**) y dale a tu programa el nombre de archivo HelloWorld como se muestra a continuación:



Después de guardar tu programa KPL, existen dos sitios en los que puedes comprobar que se guardó correctamente:

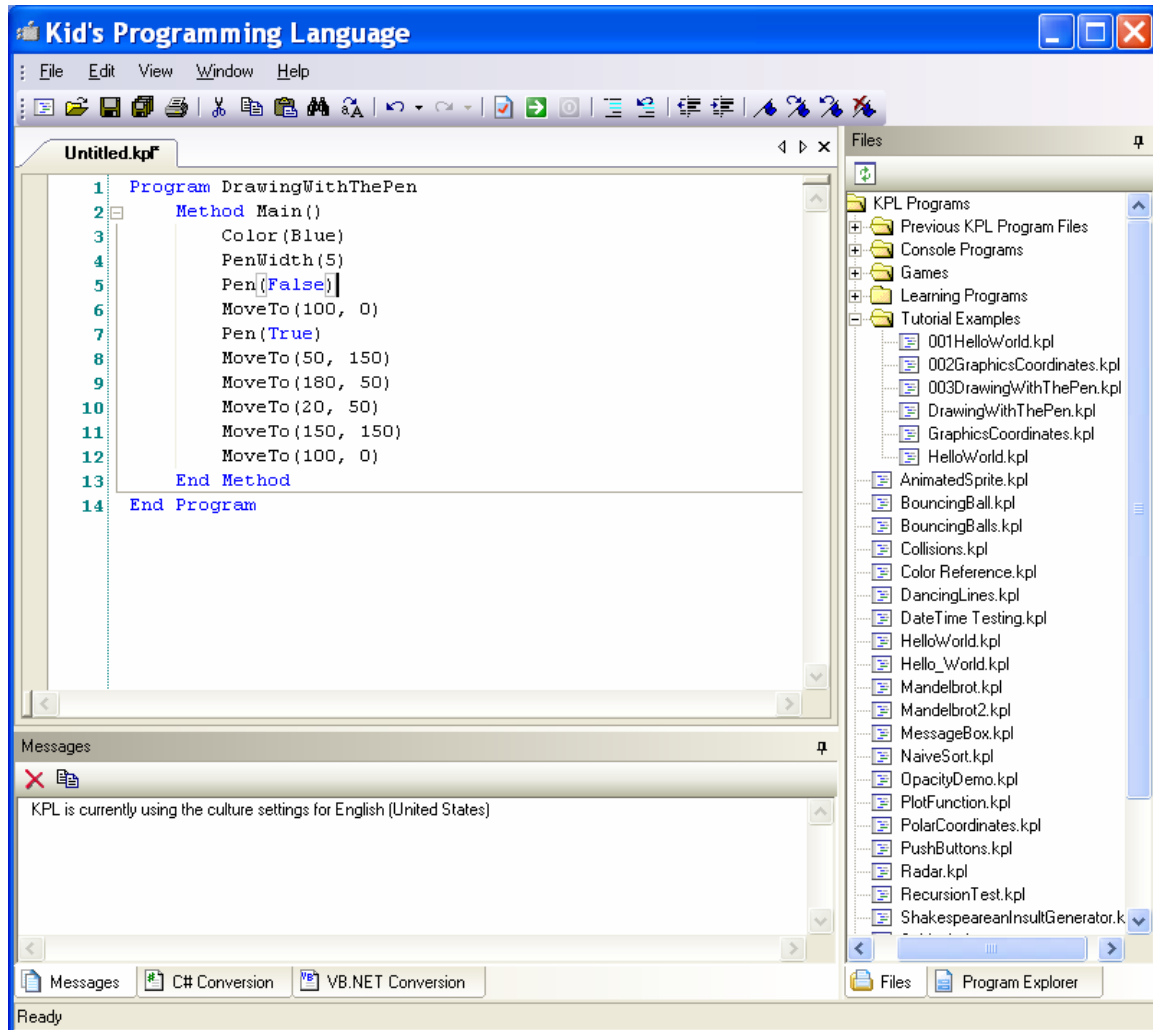


Ahora que tu programa está guardado en el disco, puedes volverlo a abrir en cualquier momento dando doble clic sobre el archivo en el explorador de Archivos (**Files**) ubicado al lado derecho de la pantalla de KPL. Cuando vuelvas a abrir tu programa KPL, lo verás aparecer como se ve a continuación – ¡no te asustes, todo tu código sigue ahí!



Da clic en el icono "+" que aparece en la línea 3 y todo el código dentro del Method Main se "expandirá" para que lo puedas ver. Si das clic en el icono "-", el código se "colapsará" nuevamente. Esta no es una función muy útil en un programa tan pequeño como este, pero conforme estés escribiendo programas más grandes y completos, verás que la posibilidad de ocultar y mostrar las líneas del código en esta manera será algo muy conveniente para ti como programador.

Después de guardar el programa **HelloWorld**, da clic en el menú Archivo (**File**), después selecciona la opción Nueva Ventana (**New Document Window**), y a continuación verás un nuevo programa llamado **Untitled.kpl**, como el que utilizaste al empezar. Elimina todo el código de la misma manera como lo hiciste anteriormente y escribe en el editor el código para nuestro segundo ejemplo:



Si escribes todo dentro de KPL exactamente en la misma forma como se muestra aquí y lo ejecutas, verás dibujada la estrella azul y en ese momento ¡serás un programador de gráficos en computadora! ¡Qué tal!

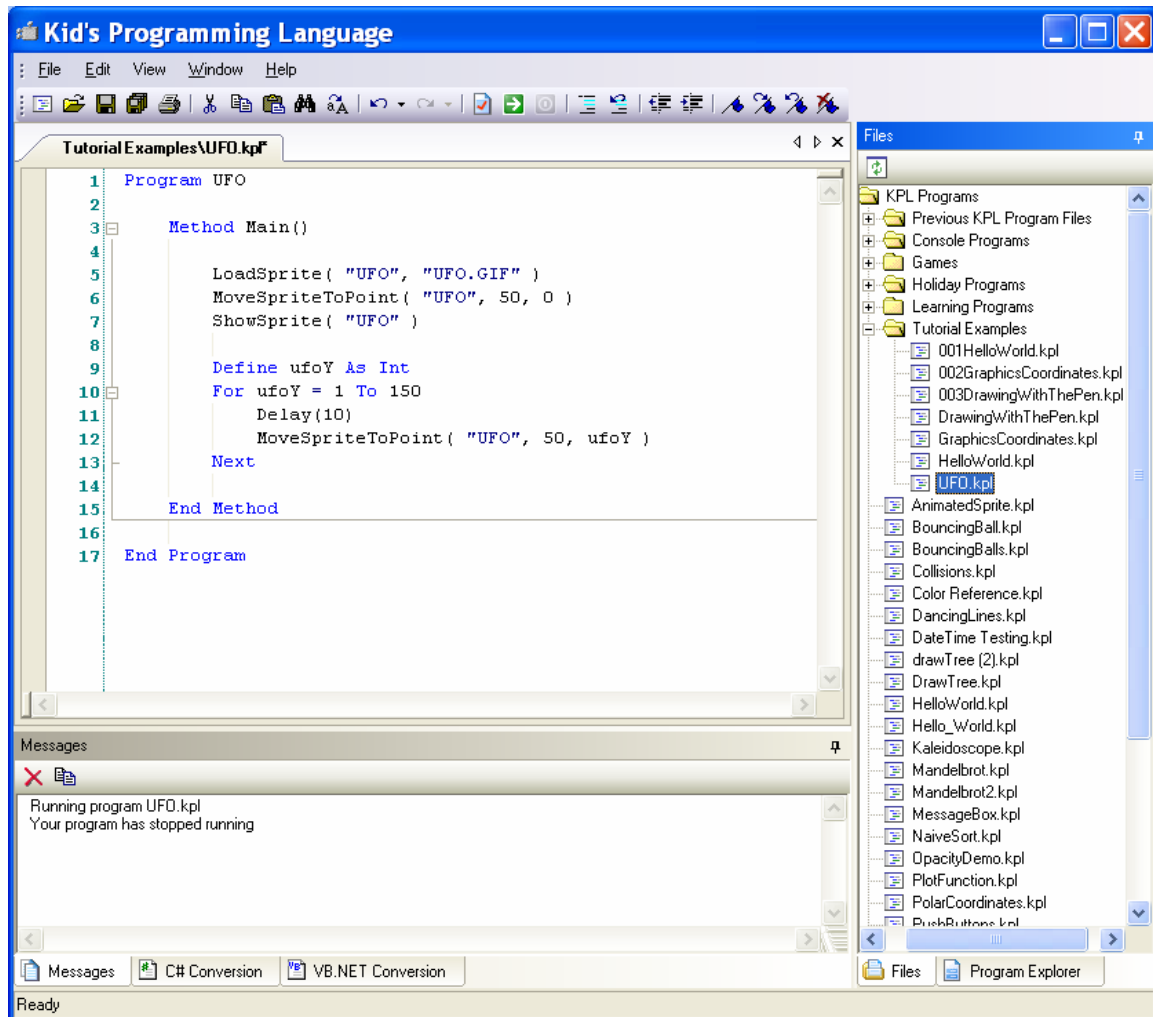
Recuerda que si encuentras un error, solo debes mantener la calma y la paciencia, siendo muy cuidadoso para localizar las diferencias entre tu programa y este ejemplo. Cuando hayas corregido el código para que se vea exactamente igual al que se muestra en la imagen, podrás ejecutarlo correctamente.

¡No olvides guardar este programa cuando funcione correctamente!

Este es también un programa divertido para que puedas experimentar un poco. ¿Qué colores se ven bien en tu estrella? ¿Cómo se ve la estrella si le dices a la pluma que dibuje líneas más

gruesas o más delgadas? ¿Cómo harías para dibujar un cuadrado o un triángulo en vez de una estrella? Esta es más difícil: ¿cómo harías para que la estrella fuera más grande o más pequeña?

Después de guardar el programa **DrawingWithThePen**, da clic en el menú Archivo (**File**), selecciona la opción Nueva Ventana (**New Document Window**), y podrás ver un nuevo programa llamado **Untitled.kpl**. Borra todo el código como lo hiciste antes y después escribe el código para nuestro ejemplo del platillo volador:



Si escribes en KPL todo el código exactamente como aparece en la imagen y lo ejecutas, verás volando al platillo volador. ¡Qué tal!

Recuerda que si encuentras un error, solo debes mantener la calma y la paciencia, siendo muy cuidadoso para localizar las diferencias entre tu programa y este ejemplo. Cuando hayas corregido el código para que se vea exactamente igual al que se muestra en la imagen, podrás ejecutarlo correctamente.

¡No olvides guardar este programa cuando funcione correctamente!

Intenta con estos divertidos ejercicios si quieres tener un poco más de práctica de programación:

- ¿Puedes lograr que el platillo volador se mueva más lejos en la pantalla?

- ¿Cómo harías para que el platillo volador viajara de derecha a izquierda en vez de hacerlo de arriba hacia abajo?
- ¿Puedes lograr que el platillo volador vuele hacia abajo y hacia la derecha **al mismo tiempo**?

Siguientes Pasos Después de Este Tutorial

Lo primero que puedes hacer en este punto del tutorial es explorar la carpeta Programas de Aprendizaje (**Learning Programs**) dentro de KPL. Ahí encontrarás 6 programas KPL de los que podrás aprender más – es mejor si los estudias en orden. De hecho, ¿los primeros 2 programas de los 6 son muy similares a los que ya has hecho en este tutorial! Es posible que hayan surgido nuevos Programas de Aprendizaje para el momento en que leas este tutorial – puede ser que ya formen parte de tu instalación de KPL, o podrás encontrarlos en la página de Descargas de nuestro sitio web en la dirección <http://www.kidsprogramminglanguage.com/espindex.htm>.

También es posible que desees descargar la **Guía del Usuario de KPL para Maestros**, que se encuentra disponible desde la misma página de descargas. No está pensado como un tutorial para principiantes, pero puede ser una referencia útil para ti mientras avanzas por este tutorial y trabajas en tus propios programas KPL.

Cuando te sientas cómodo con todos los Programas de Aprendizaje, podrás encontrar otros programas de ejemplo, que se incluyen en KPL, podrás abrirlos y examinarlos para poder aprender de ellos. **Kplong.kpl** y **NumberGuess.kpl** son un par de programas que podrás conocer a continuación. Los encontrarás dentro de la carpeta de Juegos (**Games**). Son programas mucho más largos – ¡porque son juegos completos! – pero cuando hayas entendido el tutorial y todos los Programas de Aprendizaje, probablemente no tendrás ningún problema para aprender esos programas también.

Más adelante encontrarás algunos vínculos a las comunidades de KPL en línea – estas mismas se encuentran disponibles desde la página de Comunidades de nuestro sitio web en la dirección <http://www.kidsprogramminglanguage.com/espindex.htm>:

[Programas KPL Divertidos](#)

[Discusión sobre KPL - Niños](#)

[Discusión sobre KPL - Padres](#)

[Discusión sobre KPL - Maestros](#)

[Preguntas y Respuestas sobre KPL](#)

[Versiones Internacionales del Lenguaje KPL](#)

Si tuvieras preguntas o necesitaras mayor asistencia con KPL, el Foro de Preguntas y Respuestas de KPL es el primer sitio donde debes buscar. Esta comunidad en línea tiene mucha actividad, ¡y diariamente se vuelve más activa!