



Tecnológico de Monterrey

Diseño de Compiladores
Ing. Elda Quiroga
02/04/18

Manual de Usuario Lenguaje “EDJO”

Introducción

El lenguaje **edjo** fue desarrollado utilizando Python3 como base, para compilar un programa en edjo es necesario realizar instalaciones. Se requiere Python3, Git y posteriormente descargar el compilador **edjo** de GitHub. A continuación se explicara detalladamente como realizar cada instalación.

Instalacion Python3

Ingresa a la liga <https://www.python.org/downloads/> y descargar la versión más reciente para tu sistema operativo.

Para confirmar la instalación de Python3 debes abrir la línea de comando y escribir lo siguiente:

```
[Edgars-MacBook-Pro:~ edgarserna$ python --version  
Python 2.7.10
```

Instalación Git

Ingresa a la liga <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git> e ingresar los comandos en la terminal dependiendo tu sistema operativo.

Para confirmar la instalación de Git debes abrir la línea de comando y escribir lo siguiente:

```
[Edgars-MacBook-Pro:~ edgarserna$ git --version  
git version 2.15.1 (Apple Git-101)
```

Instalación edjo

Abir la línea de comando y vamos a crear un directorio donde se guardaran los archivos del compilador edjo, para escoger la dirección dentro de tu ordenador utilizar el comando “cd” para navegar en tu ordenador. Una vez en la dirección deseada escribir los siguientes comandos en la línea de comando.

Copiar la liga <https://github.com/joelcantu/CompiladorEdjo.git>

-> git clone https://github.com/joelcantu/CompiladorEdjo.git

Tu consola se debe ver así:

```
[Edgars-MacBook-Pro:Documents edgarserna$ git clone https://github.com/joelcantu/CompiladorEdjo.git  
Cloning into 'CompiladorEdjo'...  
remote: Counting objects: 300, done.  
remote: Compressing objects: 100% (202/202), done.  
remote: Total 300 (delta 171), reused 221 (delta 96), pack-reused 0  
Receiving objects: 100% (300/300), 455.76 KiB | 376.00 KiB/s, done.  
Resolving deltas: 100% (171/171), done.  
Edgars-MacBook-Pro:Documents edgarserna$
```

¡FELICIDADES! Ya puedes empezar a programar en **edjo**

Tutorial lenguaje edjo

Definición de Funciones

Existen dos tipos de funciones, tipo 'void' o tipo 'int'.

Funcion 'VOID'

"func 'nombredefunción' ('parámetros separados por comma') { }

Funcion 'INT'

"func int 'nombredefunción' ('parámetros separados por comma') { return 0 }

Declaración de variables

Se pueden declarar variables globales y locales, esto significa que pueden ser declaradas fuera del main y/o dentro de las funciones.

**No se puede asignar un valor a una variable al momento de declararla
(EJ. int a = 0;)*

TIPO '*nombredevariable*' ;

**TIPO = int, decimal, string o bool*

Asignación de variables

Una vez declarada la variable es posible asignarle un valor. También es posible hacer una asignación de una expresión.

(EJ. a = a + b + c / d;)

'nombredevariable' = 'variable o constante' ;

Llamada de función

Se puede mandar a llamar una función con o sin parámetros. OJO, es distinto como mandar a llamar una funcion 'void' a una función 'int'.

FUNCION 'VOID'

'nombredefunción' ('Parametros separados por comas');

FUNCION INT

'nombredevariable' = 'nombredefunción' ('Parámetros separados por comas');

**Para funciones int es necesario asignar el resultado a una variable.*

Ciclos

Existen dos tipos de ciclos en el lenguaje edjo, WHILE y DOWHILE.

WHILE

while('condición') { }:

DOWHILE

do { } while ('condición');

Condición IF

En el lenguaje edjo es posible hacer IFs anidados, es decir, NO se puede hacer un if else, para realizar este proceso es necesario repetir la instrucción if múltiples veces.

IF

```
if ( 'condición' ) { ..... } else { .... }
```

IFs Anidados

```
if ( 'condición' ) { ..... } if ( 'condición' ) { ..... } if ( 'condición' ) { ..... } else { .... }
```

Métodos de Turtle

Para crear un objeto 'Turtle' es necesario inicializarlo y destruirlo al final.

INICIALIZAR TURTLE

```
turtle.Turtle();
```

FINALIZAR TURTLE

```
turtle.finish();
```

El compilador edjo cuenta con funciones default de Turtle, tales como 'circle', 'square', 'rectangle' y 'triangle'. Cada una de estas funciones genera automáticamente la figura del método a elegir.

CÍRCULO

```
turtle.circle('tamañodelradio o variable int')
```

CUADRO

```
turtle.square('tamañodellado o variable int')
```

RECTÁNGULO

```
turtle.rectangle('tamañodeancho o variable int' , 'tamañodellargo o variable int')
```

**Los parámetros son: ancho y largo*

TRIÁNGULO

```
turtle.triangle('tamañodelado o variable int')
```

**La función 'triangle' solo genera triángulos equiláteros*

FORWARD

La función 'forward' mueve la tortuga la distancia especificada

```
turtle.forward('cantidadnumerica' o variable)
```

RIGHT

La función 'right' gira la tortuga en unidades de grados. La tortuga gira los grados especificados a la derecha.

```
turtle.right('cantidadnumerica' o variable)
```

LEFT

La función 'left' gira la tortuga en unidades de grados. La tortuga gira los grados especificados a la izquierda.

```
turtle.left('cantidadnumerica' o variable)
```

INICIAR FILL DE UNA FIGURA

```
turtle.begin_fill()
```

DARLE COLOR AL FILL DE UNA FIGURA

```
turtle.color('color' o variablestring)
```

TERMINAR FILL DE UNA FIGURA

```
turtle.end_fill()
```

COLOR DE LA PLUMA

```
turtle.pencolor('color' o variablestring)
```

Compilar tu programa

Para compilar tu programa es necesario guardar el archivo de tu código con extensión .txt dentro de la carpeta donde descargaste el compilador edjo. Una vez guardado el programa dentro de la carpeta CompiladorEdjo, es necesario abrir el archivo edjo.py y escribir el nombre de tu programa generado para que este sea capaz de ejecutarlo.

1. Abrir edjo.py
2. Ir hasta el final del archivo y localizar la línea

```
with open(' ','r') as f:
    input = f.read()
    pp.pprint(parser.parse(input))
    edjo.dirFuncion.printDirFuncion()
    edjo.printCuadрупlos()
    maquinaVirtual = MaquinaVirtual(edjo.memoria, edjo.dirFuncion, edjo.cuadрупlos)
    maquinaVirtual.execute("Y")
```

3. En la línea " with open(' ', 'r') as f: dentro de las comillas vacías, escribir el nombre de tu programa (nombre de archivo)
*EJ. with open ('Prueba.txt', 'r') as f:

4. Abrir terminal e ir a la carpeta del compilador edjo donde se encuentra el archivo edjo.py
5. Escribir el siguiente comando
-> python3 edjo.py

```
[Edgars-MacBook-Pro:Avance4 edgarserna$ python3 edjo.py
```