



EBook Gratis

APRENDIZAJE ironpython

Free unaffiliated eBook created from
Stack Overflow contributors.

#ironpython

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con ironpython.....	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Instalar IronPython.....	2
Usando la línea de comando de IronPython.....	2
Capítulo 2: Creando formularios de Windows con IronPython.....	3
Examples.....	3
Ejemplo de Hello Word usando Windows Forms.....	3
Capítulo 3: Diferencia entre Python y IronPython.....	4
Examples.....	4
Usando ensamblados .Net desde el código de Python.....	4
IronPython está escrito en c puro.....	4
Usando genéricos dentro de IronPython.....	5
Capítulo 4: Visión general.....	6
Examples.....	6
¿Qué es IronPython?.....	6
Creditos.....	7

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [ironpython](#)

It is an unofficial and free ironpython ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official ironpython.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con ironpython

Observaciones

Esta sección proporciona una descripción general de qué es el ironpython y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de ironpython y vincular a los temas relacionados. Dado que la Documentación para ironpython es nueva, es posible que deba crear versiones iniciales de los temas relacionados.

Examples

Instalación o configuración

Instrucciones detalladas sobre cómo configurar o instalar Ironpython.

Instalar IronPython

Simplemente descargue la versión más reciente de <http://ironpython.net> y siga las instrucciones del paquete msi. Este paquete configurará todo lo que necesitas para comenzar a trabajar con ironpython.

Usando la línea de comando de IronPython

Para usar la línea de comandos de IronPython, abra `ipy.exe` o `ipy64.exe`. Ambos archivos se encuentran en la ruta que se seleccionó durante la instalación. Por defecto, se ubicarán en `C:\Program Files\IronPython 2.7\.`

Luego, comience a escribir sus declaraciones directamente en la línea de comandos de IronPython.

Por ejemplo: imprimir 'Hola Mundo'

O

Para ironpython 3: `print ('Hello World')`

Lea [Empezando con ironpython en línea](#):

<https://riptutorial.com/es/ironpython/topic/951/empezando-con-ironpython>

Capítulo 2: Creando formularios de Windows con IronPython

Examples

Ejemplo de Hello Word usando Windows Forms

Primero, las referencias se agregarán a los ensamblajes de CLR que se utilizarán.

```
import clr
clr.AddReference('System.Windows.Forms')
```

A continuación se importan los nombres que utilizaremos.

```
from System.Windows.Forms import Application, Form
```

Se creará una clase para el formulario Hello World utilizando `Form` como su subclase.

```
class HelloWorldForm(System.Windows.Forms.Form):
    def __init__(self):
        self.Text = 'Hello World'
        self.Name = 'Hello World'
```

El atributo de texto del formulario establece el texto de la barra de título.

Para ejecutar la aplicación, creamos una instancia de `HelloWorldForm`.

```
form = HelloWorldForm()
Application.Run(form)
```

La clase de `Application` proporciona métodos estáticos, como iniciar y detener una aplicación. El método `Run` static ejecuta el formulario en el hilo actual.

Lea [Creando formularios de Windows con IronPython en línea](https://riptutorial.com/es/ironpython/topic/2619/creando-formularios-de-windows-con-ironpython):

<https://riptutorial.com/es/ironpython/topic/2619/creando-formularios-de-windows-con-ironpython>

Capítulo 3: Diferencia entre Python y IronPython

Examples

Usando ensamblados .Net desde el código de Python

Con IronPython puede acceder a cualquier ensamblaje .net que se compile utilizando la misma versión o una versión inferior a la del núcleo de IronPython.

Ejemplo: Importar un ensamblaje y una clase aa .net

```
from System import Math
```

Ejemplo: Usando una clase importada:

```
from System import Math
print Math.Abs(-123)
```

También puede cargar ensamblajes adicionales utilizando el módulo `clr` incorporado.

```
import clr
clr.AddReference('Sample') # Sample.dll inside of the working directory.
```

Simplemente utilícelo como cualquier otra biblioteca .net o python.

IronPython está escrito en c puro

IronPython está completamente escrito usando el código .net (c #) administrado. Por lo tanto, todos los métodos y bibliotecas de Python builtin (como `next()` , `int()` , etc.) están escritos en .net.

Este ejemplo muestra la implementación de `len()` para una lista de diferentes tipos (solo unos pocos):

```
....
public static int len([NotNull]List/*!*/ list) {
    return list.__len__();
}

public static int len([NotNull]PythonTuple/*!*/ tuple) {
    return tuple.__len__();
}

public static int len([NotNull]PythonDictionary/*!*/ dict) {
    return dict.__len__();
}
```

....

Si necesitaríamos algún otro tipo para contar la longitud, solo agréguelos en `BuiltIn.cs` y estará disponible automáticamente.

Usando genéricos dentro de IronPython

IronPython permite utilizar clases y métodos genéricos desde el marco `.net`. Los genéricos se pueden utilizar con la misma sintaxis que para acceder a un índice. Para pasar más de un parámetro de tipo, se deben separar con una coma:

```
l = Dictionary[int, str]()
```

De esa manera creamos un diccionario donde las claves solo aceptan `integers` y los valores deben ser una `string`.

Un ejemplo de uso podría verse así

```
from System.Collections.Generic import List
lst = List[str]()
lst.Add('Hello')
lst.Add('World')
for l in lst:
    print
```

Salida

Hola

Mundo

Al agregar nuevos elementos, la verificación de tipos también se realizará:

```
lst = List[str]()
lst.Add(123)
```

Rastreo (llamadas recientes más última):

Archivo "<stdin>", línea 1, en

`TypeError: esperado str, got int`

Lea Diferencia entre Python y IronPython en línea:

<https://riptutorial.com/es/ironpython/topic/1059/diferencia-entre-python-y-ironpython>

Capítulo 4: Visión general

Examples

¿Qué es IronPython?

IronPython es una implementación de código abierto del lenguaje de programación Python que está estrechamente integrado con .NET Framework. IronPython puede usar las bibliotecas .NET Framework y Python, y otros lenguajes .NET pueden usar el código Python con la misma facilidad.

Lea Visión general en línea: <https://riptutorial.com/es/ironpython/topic/4239/vision-general>

Creditos

S. No	Capítulos	Contributors
1	Empezando con ironpython	BendEg , Chandu , Community , D. Alveno
2	Creando formularios de Windows con IronPython	D. Alveno
3	Diferencia entre Python y IronPython	BendEg
4	Visión general	D. Alveno