Kostenloses eBook

LERNEN

ironpython

Free unaffiliated eBook created from
Stack Overflow contributors.

#ironpython

Inhaltsverzeichnis

Über

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.

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

Kapitel 1: Erste Schritte mit Ironpython

Bemerkungen

In diesem Abschnitt erhalten Sie einen Überblick darüber, was Ironpython ist und warum ein Entwickler es verwenden möchte.

Es sollte auch alle großen Themen in Ironpython erwähnen und auf die verwandten Themen verweisen. Da die Dokumentation für Ironpython neu ist, müssen Sie möglicherweise erste Versionen dieser verwandten Themen erstellen.

Examples

Installation oder Setup

Detaillierte Anweisungen zum Einrichten oder Installieren von Ironpython.

Installieren Sie IronPython

Laden Sie einfach die neueste Version von http://ironpython.net herunter und folgen Sie den Anweisungen des msi-Pakets. Dieses Paket richtet alles ein, was Sie benötigen, um mit Ironpython zu arbeiten.

Verwenden der IronPython-Befehlszeile

Um die IronPython-Befehlszeile zu verwenden, öffnen Sie `ipy.exe` oder `ipy64.exe` . Beide Dateien befinden sich in dem Pfad, der während der Installation ausgewählt wurde. Standardmäßig befinden sie sich unter `C:\Program Files\IronPython 2.7\` .

Dann schreiben Sie Ihre Anweisungen direkt in die IronPython-Befehlszeile.

Zum Beispiel: 'Hello World' drucken

ODER

Für Ironpython 3: print ('Hello World')

Erste Schritte mit Ironpython online lesen: https://riptutorial.com/de/ironpython/topic/951/erste-schritte-mit-ironpython

Kapitel 2: Erstellen von Windows Forms mit IronPython

Examples

Hallo Word-Beispiel mit Windows Forms

Zuerst werden Referenzen zu den verwendeten CLR-Baugruppen hinzugefügt.

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

Als nächstes werden die Namen importiert, die wir verwenden werden.

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

Eine Klasse wird für das Hello World-Formular erstellt, wobei `Form` als Unterklasse verwendet wird.

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

Das Textattribut des Formulars legt den Text der Titelleiste fest.

Um die Anwendung auszuführen, erstellen wir eine Instanz von `HelloWorldForm` .

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

Die `Application` Klasse stellt statische Methoden bereit, z. B. das Starten und Stoppen einer Anwendung. Das `Run` statische Methode läuft das Formular auf dem aktuellen Thread.

Erstellen von Windows Forms mit IronPython online lesen:
https://riptutorial.com/de/ironpython/topic/2619/erstellen-von-windows-forms-mit-ironpython

Kapitel 3: Überblick

Examples

Was ist IronPython?

IronPython ist eine Open-Source-Implementierung der Programmiersprache Python, die eng in .NET Framework integriert ist. IronPython kann die .NET Framework- und Python-Bibliotheken verwenden, und andere .NET-Sprachen können Python-Code genauso einfach verwenden.

Überblick online lesen: https://riptutorial.com/de/ironpython/topic/4239/uberblick

Kapitel 4: Unterschied zwischen Python und IronPython

Examples

.Net-Assemblys aus Python-Code verwenden

Mit IronPython können Sie auf jede .net-Assembly zugreifen, die mit derselben oder einer niedrigeren Version als der IronPython-Kern kompiliert wird.

Beispiel: Importieren einer .net-Assembly und einer Klasse

```
from System import Math
```

Beispiel: Verwenden einer importierten Klasse:

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

Sie können auch zusätzliche Baugruppen laden, indem Sie das integrierte `clr` Modul verwenden.

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

Dann verwenden Sie es einfach wie jede andere .net- oder Python-Bibliothek.

IronPython ist in reinem c # geschrieben

IronPython wird vollständig mit verwaltetem .net (c #) Code geschrieben. Daher werden alle `builtin` Python-Methoden und -Bibliotheken (wie `next()` , `int()` usw.) in .net geschrieben.

Dieses Beispiel zeigt die Implementierung von `len()` für eine Liste verschiedener Typen (nur einige):

```
....

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__();
}
```

....

Wenn wir einen anderen Typ zum Abzählen der Länge benötigen, fügen Sie sie einfach in `Builtin.cs` und sie wird automatisch verfügbar sein.

Verwendung von Generika in IronPython

IronPython ermöglicht die Verwendung generischer Klassen und Methoden aus dem .NET-Framework. Generics können mit derselben Syntax wie der Zugriff auf einen Index verwendet werden. Um mehrere Typparameter übergeben zu können, müssen sie mit einem Komma getrennt werden:

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

Auf diese Weise erstellen wir ein Wörterbuch, in dem Schlüssel nur `integers` akzeptieren und die Werte eine `string` .

Eine Beispielnutzung könnte so aussehen

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

Ausgabe

> Hallo
>
> Welt

Beim Hinzufügen neuer Elemente wird auch die Typüberprüfung durchgeführt:

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

> Traceback (letzter Anruf zuletzt):
>
> Datei "<stdin>", Zeile 1, in
>
> TypeError: erwartete str, wurde int

Unterschied zwischen Python und IronPython online lesen:
https://riptutorial.com/de/ironpython/topic/1059/unterschied-zwischen-python-und-ironpython

Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit Ironpython	BendEg, Chandu, Community, D. Alveno
2	Erstellen von Windows Forms mit IronPython	D. Alveno
3	Überblick	D. Alveno
4	Unterschied zwischen Python und IronPython	BendEg