



**EBook Gratuito**

# APPENDIMENTO

---

# ironpython

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#ironpython**

# Sommario

Di.....	1
<b>Capitolo 1: Iniziare con ironpython.....</b>	<b>2</b>
Osservazioni.....	2
Examples.....	2
Installazione o configurazione.....	2
<b>Installa IronPython.....</b>	<b>2</b>
Utilizzando la riga di comando IronPython.....	2
<b>Capitolo 2: Creazione di Windows Form con IronPython.....</b>	<b>3</b>
Examples.....	3
Ciao esempio di Word usando Windows Forms.....	3
<b>Capitolo 3: Differenza tra Python e IronPython.....</b>	<b>4</b>
Examples.....	4
Usando gli assembly .Net dal codice Python.....	4
IronPython è scritto in puro C #.....	4
Usare i generici in IronPython.....	5
<b>Capitolo 4: Panoramica.....</b>	<b>6</b>
Examples.....	6
Cos'è IronPython?.....	6
<b>Titoli di coda.....</b>	<b>7</b>

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](mailto:info@zzzprojects.com)

---

# Capitolo 1: Iniziare con ironpython

## Osservazioni

Questa sezione fornisce una panoramica di ciò che ironpython è, e perché uno sviluppatore potrebbe voler usarlo.

Dovrebbe anche menzionare qualsiasi argomento di grandi dimensioni all'interno di ironpython e collegarsi agli argomenti correlati. Poiché la documentazione di ironpython è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

## Examples

### Installazione o configurazione

Istruzioni dettagliate su come installare o installare ironpython.

---

## Installa IronPython

Basta scaricare la versione più recente da <http://ironpython.net> e seguire le istruzioni dal pacchetto msi. Questo pacchetto configurerà tutto il necessario per iniziare a lavorare con ironpython.

### Utilizzando la riga di comando IronPython

Per utilizzare la riga di comando di IronPython, apri `ipy.exe` o `ipy64.exe`. Entrambi i file si trovano nel percorso che è stato selezionato durante l'installazione. Per impostazione predefinita si troveranno in `C:\Program Files\IronPython 2.7\`.

Quindi inizia a scrivere le tue istruzioni direttamente nella riga di comando di IronPython.

Ad esempio: stampa "Hello World"

O

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

Leggi Iniziare con ironpython online: <https://riptutorial.com/it/ironpython/topic/951/iniziare-con-ironpython>

---

# Capitolo 2: Creazione di Windows Form con IronPython

## Examples

### Ciao esempio di Word usando Windows Forms

Innanzitutto, i riferimenti verranno aggiunti agli assembly CLR che verranno utilizzati.

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

Successivamente verranno importati i nomi che useremo.

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

Verrà creata una classe per il modulo Hello World utilizzando `Form` come sottoclasse.

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

L'attributo di testo del modulo imposta il testo della barra del titolo.

Per eseguire l'applicazione, creiamo un'istanza di `HelloWorldForm`.

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

La classe `Application` fornisce metodi statici e come l'avvio e l'arresto di un'applicazione. Il metodo statico `Run` esegue il modulo sul thread corrente.

Leggi Creazione di Windows Form con IronPython online:

<https://riptutorial.com/it/ironpython/topic/2619/creazione-di-windows-form-con-ironpython>

---

# Capitolo 3: Differenza tra Python e IronPython

## Examples

### Usando gli assembly .Net dal codice Python

Con IronPython puoi accedere a qualsiasi assembly .net che viene compilato utilizzando la stessa versione o una versione precedente rispetto al core IronPython.

Esempio: importazione di un assembly .net e classe

```
from System import Math
```

Esempio: utilizzo di una classe importata:

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

È anche possibile caricare ulteriori assiemi utilizzando il modulo `clr` incorporato.

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

Quindi basta usarlo come qualsiasi altra libreria .net o python.

### IronPython è scritto in puro C #

IronPython è scritto completamente usando il codice gestito .net (c #). Così tutti i `builtin` metodi Python e le librerie (come `next()`, `int()`, ecc) sono writtin in .NET.

Questo esempio mostra l'implementazione di `len()` per un elenco di tipi diversi (solo alcuni):

```
....
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__();
}
....
```

Se avessimo bisogno di qualche altro tipo per contare la durata, aggiungili in `Builtin.cs` e sarà disponibile automaticamente.

## Usare i generici in IronPython

IronPython consente di utilizzare classi e metodi generici dal framework .net. Generics può essere utilizzato con la stessa sintassi dell'accesso a un indice. Per passare più di un parametro di tipo, devono essere separati da una virgola:

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

In questo modo creiamo un dizionario in cui le chiavi accettano solo `integers` e i valori devono essere una `string`.

Un esempio di utilizzo potrebbe essere simile a questo

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

## Produzione

Ciao

Mondo

Quando si aggiungono nuovi articoli, verrà eseguito anche il controllo del tipo:

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

Traceback (ultima chiamata più recente):

File "<stdin>", riga 1, in

TypeError: atteso str, ottenuto int

[Leggi Differenza tra Python e IronPython online:](https://riptutorial.com/it/python-e-ironpython-online)

<https://riptutorial.com/it/ironpython/topic/1059/differenza-tra-python-e-ironpython>

---

# Capitolo 4: Panoramica

## Examples

### Cos'è IronPython?

IronPython è un'implementazione open-source del linguaggio di programmazione Python che è strettamente integrato con .NET Framework. IronPython può utilizzare le librerie .NET Framework e Python e altri linguaggi .NET possono usare il codice Python altrettanto facilmente.

Leggi Panoramica online: <https://riptutorial.com/it/ironpython/topic/4239/panoramica>



---

## Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con ironpython	<a href="#">BendEg</a> , <a href="#">Chandu</a> , <a href="#">Community</a> , <a href="#">D. Alveno</a>
2	Creazione di Windows Form con IronPython	<a href="#">D. Alveno</a>
3	Differenza tra Python e IronPython	<a href="#">BendEg</a>
4	Panoramica	<a href="#">D. Alveno</a>