



**eBook Gratuit**

# APPRENEZ ironpython

eBook gratuit non affilié créé à partir des  
**contributeurs de Stack Overflow.**

**#ironpython**

# Table des matières

À propos.....	1
<b>Chapitre 1: Démarrer avec ironpython.....</b>	<b>2</b>
Remarques.....	2
Exemples.....	2
Installation ou configuration.....	2
<b>Installez IronPython.....</b>	<b>2</b>
Utilisation de la ligne de commande IronPython.....	2
<b>Chapitre 2: Création de Windows Forms avec IronPython.....</b>	<b>3</b>
Exemples.....	3
Bonjour Word exemple utilisant Windows Forms.....	3
<b>Chapitre 3: Différence entre Python et IronPython.....</b>	<b>4</b>
Exemples.....	4
Utiliser des assemblys .Net à partir de code Python.....	4
IronPython est écrit en pure c #.....	4
Utiliser des génériques dans IronPython.....	5
<b>Chapitre 4: Vue d'ensemble.....</b>	<b>6</b>
Exemples.....	6
Qu'est-ce que IronPython?.....	6
<b>Crédits.....</b>	<b>7</b>

# À propos

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)

---

# Chapitre 1: Démarrer avec ironpython

## Remarques

Cette section fournit une vue d'ensemble de ce qu'est ironpython et pourquoi un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets dans ironpython, et établir un lien avec les sujets connexes. La documentation de ironpython étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

## Exemples

### Installation ou configuration

Instructions détaillées sur la configuration ou l'installation de ironpython.

---

## Installez IronPython

Il suffit de télécharger la version la plus récente depuis <http://ironpython.net> et de suivre les instructions du package msi. Ce paquet va configurer tout ce dont vous avez besoin pour commencer à travailler avec ironpython.

### Utilisation de la ligne de commande IronPython

Pour utiliser la ligne de commande IronPython, ouvrez `ipy.exe` ou `ipy64.exe`. Les deux fichiers se trouvent dans le chemin sélectionné lors de l'installation. Par défaut, ils seront situés dans

```
C:\Program Files\IronPython 2.7\.
```

Commencez ensuite à écrire vos instructions directement dans la ligne de commande IronPython.

Par exemple: `print 'Hello World'`

OU

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

Lire Démarrer avec ironpython en ligne: <https://riptutorial.com/fr/ironpython/topic/951/demarrer-avec-ironpython>

---

# Chapitre 2: Création de Windows Forms avec IronPython

## Exemples

### Bonjour Word exemple utilisant Windows Forms

Tout d'abord, des références seront ajoutées aux assemblées CLR qui seront utilisées.

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

Ensuite, les noms que nous utiliserons sont importés.

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

Une classe sera créée pour le formulaire Hello World en utilisant `Form` comme sous-classe.

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

L'attribut `Text` du formulaire définit le texte de la barre de titre.

Pour exécuter l'application, nous créons une instance de `HelloWorldForm`.

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

La classe `Application` fournit des méthodes statiques telles que le démarrage et l'arrêt d'une application. La méthode statique `Run` exécute le formulaire sur le thread en cours.

Lire [Création de Windows Forms avec IronPython en ligne](https://riptutorial.com/fr/ironpython/topic/2619/creation-de-windows-forms-avec-ironpython):

<https://riptutorial.com/fr/ironpython/topic/2619/creation-de-windows-forms-avec-ironpython>

---

# Chapitre 3: Différence entre Python et IronPython

## Exemples

### Utiliser des assemblies .Net à partir de code Python

Avec IronPython, vous pouvez accéder à tout assembly .net compilé avec une version identique ou inférieure à celle du noyau IronPython.

Exemple: importation d'un ensemble et d'une classe .net

```
from System import Math
```

Exemple: Utilisation d'une classe importée:

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

Vous pouvez également charger des assemblies supplémentaires à l'aide du module `clr` intégré.

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

Alors, utilisez-le comme n'importe quelle autre bibliothèque .net ou python.

### IronPython est écrit en pure c #

IronPython est entièrement écrit en utilisant le code .net (c #) géré. Ainsi, toutes les méthodes et bibliothèques python `builtin` (telles que `next()`, `int()`, etc.) sont écrites dans .net.

Cet exemple montre l'implémentation de `len()` pour une liste de types différents (quelques-uns seulement):

```
....

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 nous avons besoin d'un autre type pour compter la longueur, ajoutez-les `Builtin.cs` dans `Builtin.cs` et il sera disponible automatiquement.

## Utiliser des génériques dans IronPython

IronPython permet d'utiliser des classes et des méthodes génériques du framework .net. Les génériques peuvent être utilisés avec la même syntaxe que l'accès à un index. Pour transmettre plusieurs paramètres de type, ils doivent être séparés par une virgule:

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

De cette façon, nous créons un dictionnaire où les clés acceptent uniquement les `integers` et les valeurs doivent être une `string`.

Un exemple d'utilisation pourrait ressembler à ceci

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

### Sortie

Bonjour

Monde

Lors de l'ajout de nouveaux éléments, une vérification de type sera également effectuée:

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

Traceback (dernier appel en dernier):

Fichier "<stdin>", ligne 1, dans

TypeError: str attendu, obtenu int

Lire [Différence entre Python et IronPython en ligne](https://riptutorial.com/fr/python-et-ironpython):

<https://riptutorial.com/fr/ironpython/topic/1059/difference-entre-python-et-ironpython>

---

# Chapitre 4: Vue d'ensemble

## Exemples

### Qu'est-ce que IronPython?

IronPython est une implémentation open-source du langage de programmation Python qui est étroitement intégré au .NET Framework. IronPython peut utiliser les bibliothèques .NET Framework et Python, et d'autres langages .NET peuvent utiliser le code Python tout aussi facilement.

Lire Vue d'ensemble en ligne: <https://riptutorial.com/fr/ironpython/topic/4239/vue-d-ensemble>



---

# Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec ironpython	<a href="#">BendEg</a> , <a href="#">Chandu</a> , <a href="#">Community</a> , <a href="#">D. Alveno</a>
2	Création de Windows Forms avec IronPython	<a href="#">D. Alveno</a>
3	Différence entre Python et IronPython	<a href="#">BendEg</a>
4	Vue d'ensemble	<a href="#">D. Alveno</a>