



Бесплатная электронная книга

УЧУСЬ

ironpython

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

#ironpython

.....	1
<b>1:</b> .....	<b>2</b>
.....	2
Examples.....	2
.....	2
<b>IronPython.....</b>	<b>2</b>
IronPython.....	2
<b>2:</b> .....	<b>3</b>
Examples.....	3
IronPython?.....	3
<b>3: Python IronPython.....</b>	<b>4</b>
Examples.....	4
.Net- Python.....	4
IronPython c#.....	4
IronPython.....	5
<b>4: Windows Forms IronPython.....</b>	<b>6</b>
Examples.....	6
Hello Word Windows Forms.....	6
.....	7

---

# Около

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)

---

# глава 1: Начало работы с железом

## замечания

В этом разделе представлен обзор того, что такое ironpython, и почему разработчик может захотеть его использовать.

Он также должен упомянуть о любых крупных предметах в ironpython и ссылаться на связанные темы. Поскольку Documentation for ironpython является новым, вам может потребоваться создать начальные версии этих связанных тем.

## Examples

### Установка или настройка

Подробные инструкции по установке или установке ironpython.

---

## Установить IronPython

Просто загрузите самую новую версию с <http://ironpython.net> и следуйте инструкциям из пакета msi. Этот пакет настроит все, что вам нужно, чтобы начать работу с ironpython.

### Использование командной строки IronPython

Чтобы использовать командную строку IronPython, откройте `ipy.exe` или `ipy64.exe`. Оба файла находятся в пути, который был выбран во время установки. По умолчанию они будут расположены в `C:\Program Files\IronPython 2.7\`.

Затем начните писать свои сообщения непосредственно в командной строке IronPython.

Например: `print 'Hello World'`

ИЛИ ЖЕ

Для ironpython 3: `print («Hello World»)`

Прочитайте Начало работы с железом онлайн: <https://riptutorial.com/ru/ironpython/topic/951/начало-работы-с-железом>

---

## глава 2: обзор

### Examples

#### Что такое IronPython?

IronPython - это реализация языка программирования Python с открытым исходным кодом, которая тесно интегрирована с .NET Framework. IronPython может использовать библиотеки .NET Framework и Python, а другие языки .NET могут также легко использовать код Python.

Прочитайте обзор онлайн: <https://riptutorial.com/ru/ironpython/topic/4239/обзор>

---

# глава 3: Разница между Python и IronPython

## Examples

### Использование .Net-сборок из кода Python

С IronPython вы можете получить доступ к любой сборке .net, которая скомпилирована с использованием той же или более низкой версии, что и ядро IronPython.

Пример: импорт сборки и класса aa .net

```
from System import Math
```

Пример. Использование импортированного класса:

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

Вы также можете загрузить дополнительные сборки, используя встроенный модуль `clr`.

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

Вместо этого используйте его как любую другую .net или библиотеку python.

### IronPython написан в чистом c #

IronPython полностью написан с использованием управляемого .net (с #) кода. Таким образом, все `builtin` методы и библиотеки python (такие как `next()`, `int()` и т. Д.) Записываются в .net.

В этом примере показана реализация `len()` для списка разных типов (всего несколько):

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

....

Если нам понадобится какой-то другой тип для подсчета длины, просто добавьте их в `Builtin.cs` и он будет доступен автоматически.

## Использование дженериков в IronPython

IronPython позволяет использовать общие классы и методы из .NET Framework. Дженерики могут использоваться с тем же синтаксисом, что и доступ к индексу. Для передачи нескольких параметров типа они должны быть разделены запятой:

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

Таким образом, мы создаем словарь, в котором ключи принимают только `integers` а значения должны быть `string`.

Пример использования может выглядеть так:

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

### Выход

Привет

Мир

При добавлении новых элементов также будет проверяться тип:

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

Traceback (последний последний вызов):

Файл «<stdin>», строка 1, в

TypeError: expected str, got int

Прочитайте [Разница между Python и IronPython онлайн](https://riptutorial.com/ru/python/online-differences-between-python-and-ironpython):

<https://riptutorial.com/ru/ironpython/topic/1059/разница-между-python-и-ironpython>

---

# глава 4: Создание Windows Forms с помощью IronPython

## Examples

### Пример Hello Word с использованием Windows Forms

Во-первых, ссылки будут добавлены в сборки CLR, которые будут использоваться.

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

Затем будут импортированы имена, которые мы будем использовать.

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

Класс будет создан для формы Hello World, используя `Form` качестве своего подкласса.

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

Текстовый атрибут формы устанавливает текст заголовка.

Чтобы запустить приложение, мы создаем экземпляр `HelloWorldForm`.

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

Класс `Application` предоставляет статические методы, такие как запуск и остановка приложения. Статический метод `Run` запускает форму в текущем потоке.

Прочитайте [Создание Windows Forms с помощью IronPython онлайн](https://riptutorial.com/ru/ironpython/topic/2619/создание-windows-forms-с-помощью-ironpython):

<https://riptutorial.com/ru/ironpython/topic/2619/создание-windows-forms-с-помощью-ironpython>

---

## кредиты

S. No	Главы	Contributors
1	Начало работы с железом	<a href="#">BendEg</a> , <a href="#">Chandu</a> , <a href="#">Community</a> , <a href="#">D. Alveno</a>
2	обзор	<a href="#">D. Alveno</a>
3	Разница между Python и IronPython	<a href="#">BendEg</a>
4	Создание Windows Forms с помощью IronPython	<a href="#">D. Alveno</a>