

# Non-official Python implementations

All Versions

## Examples

### IronPython

Open-source implementation for .NET and Mono written in C#, licensed under Apache License 2.0. It relies on DLR (Dynamic Language Runtime). It supports only version 2.7, version 3 is currently being developed.

Differences with CPython:

- Tight integration with .NET Framework.
- Strings are Unicode by default.
- Does not support extensions for CPython written in C.
- Does not suffer from Global Interpreter Lock.
- Performance is usually lower, though it depends on tests.

### Hello World

```
print "Hello World!"
```

You can also use .NET functions:

```
import clr
from System import Console
Console.WriteLine("Hello World!")
```

### External links

- [Official website](#)
- [GitHub repository](#)

### Jython

Open-source implementation for JVM written in Java, licensed under Python Software Foundation License. It supports only version 2.7, version 3 is currently being developed.

Differences with CPython:

- Tight integration with JVM.
- Strings are Unicode.
- Does not support extensions for CPython written in C.
- Does not suffer from Global Interpreter Lock.
- Performance is usually lower, though it depends on tests.

### Hello World

```
print "Hello World!"
```

You can also use Java functions:

```
from java.lang import System
System.out.println("Hello World!")
```

### External links

- [Official website](#)
- [Mercurial repository](#)

### Transcrypt

Transcrypt is a tool to precompile a fairly extensive subset of Python into compact, readable Javascript. It has the following characteristics:

- Allows for classical OO programming with multiple inheritance using pure Python syntax, parsed by CPython's native parser
- Seamless integration with the universe of high-quality web-oriented JavaScript libraries, rather than the desktop-oriented Python ones
- Hierarchical URL based module system allowing module distribution via PyPi
- Simple relation between Python source and generated JavaScript code for easy debugging
- Multi-level sourcemaps and optional annotation of target code with source references
- Compact downloads: kBs rather than MBs

- Compact downloads, KB's rather than MB's
- Optimized JavaScript code, using memoization (call caching) to optionally bypass the prototype lookup chain
- Operator overloading can be switched on and off locally to facilitate readable numerical math

## Code size and speed

Experience has shown that 650 kB of Python sourcecode roughly translates in the same amount of JavaScript source code. The speed matches the speed of handwritten JavaScript and can surpass it if call memoizing is switched on.

## Integration with HTML

```
<script src="__javascript__/_hello.js"></script>
<h2>Hello demo</h2>

<p>
<div id = "greet">...</div>
<button onclick="hello.solarSystem.greet ()">Click me repeatedly!</button>

<p>
<div id = "explain">...</div>
<button onclick="hello.solarSystem.explain ()">And click me repeatedly too!</button>
```

## Integration with JavaScript and DOM

```
from itertools import chain

class SolarSystem:
    planets = [list (chain (planet, (index + 1,))) for index, planet in enumerate ((
        ('Mercury', 'hot', 2240),
        ('Venus', 'sulphurous', 6052),
        ('Earth', 'fertile', 6378),
        ('Mars', 'reddish', 3397),
        ('Jupiter', 'stormy', 71492),
        ('Saturn', 'ringed', 60268),
        ('Uranus', 'cold', 25559),
        ('Neptune', 'very cold', 24766)
    ))]

    lines = (
        '{ } is a { } planet',
        'The radius of { } is { } km',
        '{ } is planet nr. { } counting from the sun'
    )

    def __init__ (self):
        self.lineIndex = 0

    def greet (self):
        self.planet = self.planets [int (Math.random () * len (self.planets))]
        document.getElementById ('greet') .innerHTML = 'Hello {}'.format (self.planet [0])
        self.explain ()

    def explain (self):
        document.getElementById ('explain') .innerHTML = (
            self.lines [self.lineIndex] .format (self.planet [0], self.planet [self.lineIndex +
            ])
        )
        self.lineIndex = (self.lineIndex + 1) % 3
        solarSystem = SolarSystem ()
```

## Integration with other JavaScript libraries

Transcript can be used in combination with any JavaScript library without special measures or syntax. In the documentation examples are given for a.o. react.js, riot.js, fabric.js and node.js.

## Relation between Python and JavaScript code

### Python

```
class A:
    def __init__ (self, x):
        self.x = x

    def show (self, label):
        print ('A.show', label, self.x)

class B:
    def __init__ (self, y):
        alert ('In B constructor')
        self.y = y

    def show (self, label):
        print ('B.show', label, self.y)

class C (A, B):
    def __init__ (self, x, y):
        alert ('In C constructor')
        A.__init__ (self, x)
```

```

        B.__init__(self, y)
        self.show('constructor')

    def show(self, label):
        B.show(self, label)
        print('C.show', label, self.x, self.y)

a = A(1001)
a.show('america')

b = B(2002)
b.show('russia')

c = C(3003, 4004)
c.show('netherlands')

show2 = c.show
show2('copy')
```

## JavaScript

```

var A = __class__('A', [object], {
    get __init__() {return __get__(this, function(self, x) {
        self.x = x;
    }}),
    get show() {return __get__(this, function(self, label) {
        print('A.show', label, self.x);
    }})
});
var B = __class__('B', [object], {
    get __init__() {return __get__(this, function(self, y) {
        alert('In B constructor');
        self.y = y;
    }}),
    get show() {return __get__(this, function(self, label) {
        print('B.show', label, self.y);
    }})
});
var C = __class__('C', [A, B], {
    get __init__() {return __get__(this, function(self, x, y) {
        alert('In C constructor');
        A.__init__(self, x);
        B.__init__(self, y);
        self.show('constructor');
    }}),
    get show() {return __get__(this, function(self, label) {
        B.show(self, label);
        print('C.show', label, self.x, self.y);
    }})
});
var a = A(1001);
a.show('america');
var b = B(2002);
b.show('russia');
var c = C(3003, 4004);
c.show('netherlands');
var show2 = c.show;
show2('copy');
```

## External links

- Official website: <http://www.transcrypt.org/>
- Repository: <https://github.com/JdeH/Transcrypt>

## Syntax

## Parameters

## Remarks