

Python

Python ist eine klar strukturierte Programmiersprache, die verschiedene Programmiermodelle (imperativ, funktional, objektorientiert) unterstützt und über eine umfangreiche Standard-Bibliothek verfügt. Python-Code ist sehr gut lesbar und wartbar, da die Sprache eine einfache, elegante Syntax mit Einrückung verwendet.



Grundeigenschaften von Python

- Stark typisiert mit dynamischer Typisierung
- Case sensitive (d.h. var \neq VAR)
- Objektorientiert (d.h. alles ist ein Objekt)

Syntax

- Blöcke werden durch Einrückung definiert und Ausdrücke, die Einrückung erwarten, enden mit „:“
- Kommentarzeilen beginnen mit „#“
- Wertzuweisung erfolgt mit Gleichheitszeichen („=“)
- Test auf Gleichheit mit zwei Gleichheitszeichen („==“)

```
>>> mystring = "Hello "  
>>> mystring += "World"  
>>> print mystring  
Hello World
```

Datentypen

- Verfügbare Datenstrukturen sind Listen, Tupel und Dictionaries
- Zugriff auf Array-Bereiche mit Doppelpunkt („:“)

```
>>> mylist = ["List item 1", 2, 3.14]  
>>> mylist[0] = "New item 1"  
>>> mydict = {"key1": "value1", 2: 3, "pi": 3.14}  
>>> mydict["pi"] = 3.15  
>>> mytuple = (1, 2, 3)  
>>> print mylist[0:2]  
['List item 1', 2]  
>>> print mylist[1:]  
[2, 3.14]
```

Strings

- Haben einfache oder doppelte Anführungszeichen
- Unicode wird unterstützt (Syntax: u"Unicode String")
- Wertersetzung in Strings durch den Operator „%“

```
>>> print "Number: %s String: %s" % (17, 3 * "-")  
Number: 17 String: ---
```

Funktionen

- Werden definiert durch das Schlüsselwort „def“
- Optionale und Schlüsselwort-Argumente sind erlaubt

- Der Rückgabewerte kann ein Tupel sein

```
def example(a_list, an_int=2,  
            a_string="A Default String"):  
    a_list.append("A new item")  
    an_int = 4  
    return a_list, an_int, a_string
```

Klassen

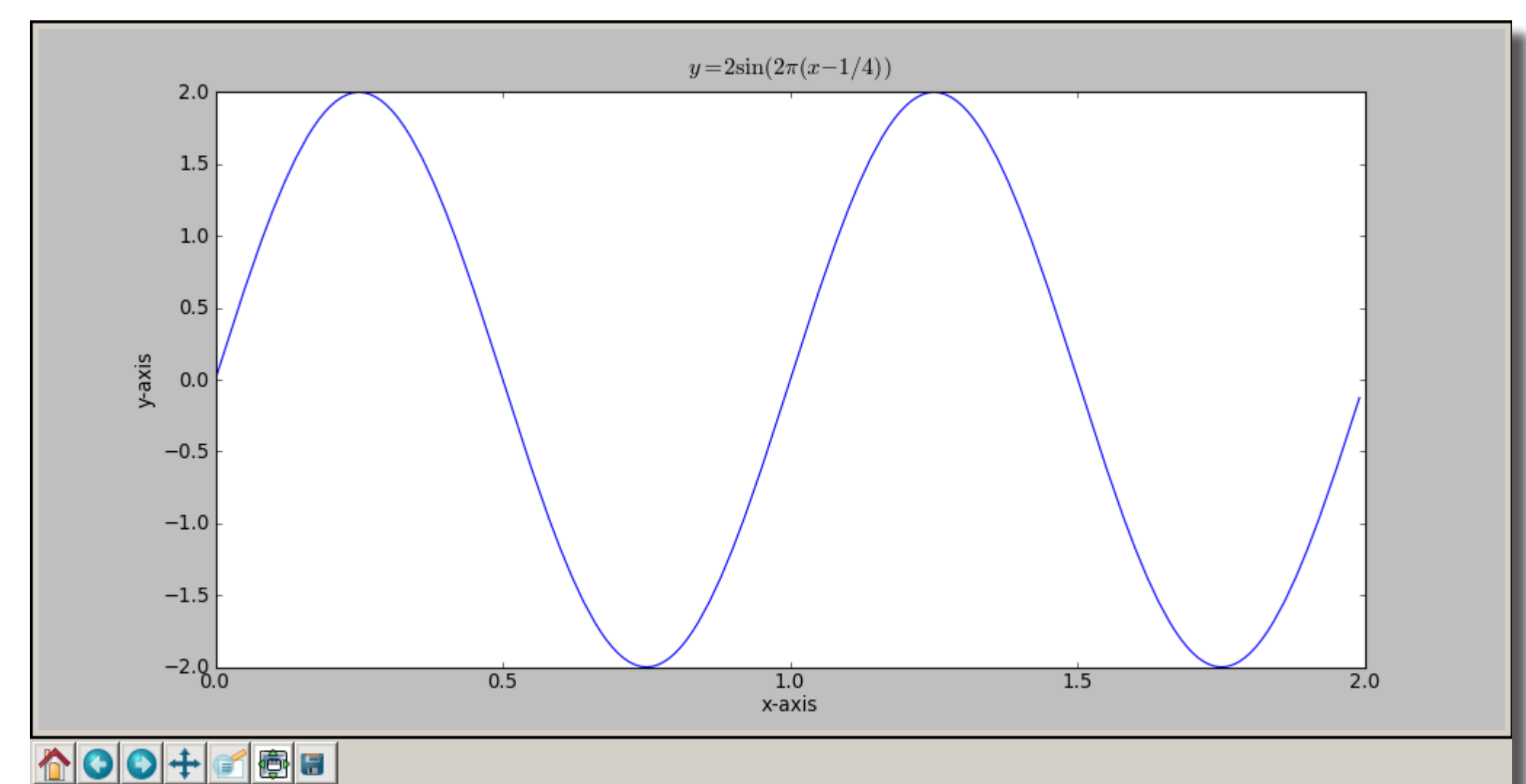
- Mehrfachvererbung wird unterstützt
- Private Variablen und Methoden beginnen mit zwei Unterstrichen (z.B. der Konstruktor „__init__“)

```
class MyClass:  
    common = 10  
    def __init__(self):  
        self.myvariable = 3  
    def myfunction(self, arg1, arg2):  
        return self.myvariable
```

Bibliotheken und Module

- Es gibt viele leistungsfähige Module, z.B. *Matplotlib*

```
from pylab import * # matplotlib  
  
x = arange(0, 2, 0.01)  
y = 2 * sin(2 * pi * (x - 1 / 4))  
  
plot(x, y)  
xlabel('x-axis')  
ylabel('y-axis')  
title(r'$y=2\sin(2\pi(x-1/4))$')  
show()
```



Alle Informationen zu Python gibt es unter <http://www.python.org>

