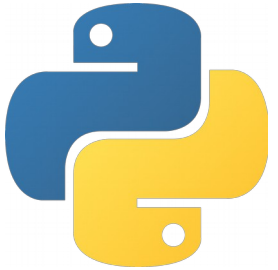
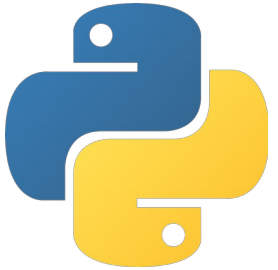
Python
Erstellen einer Distribution



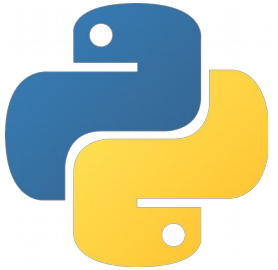
- Distribution von Python-Projekten
  - Vielleicht haben Sie sogar schon ein Programm oder Modul in Python geschrieben, das auch für andere Leute von Nutzen sein könnte
  - Dazu ist in Pythons Standardbibliothek das Modul distutils enthalten, mit dem sich fertige Distributionen Ihres Programms oder Moduls erstellen lassen

Python
Erstellen einer Distribution



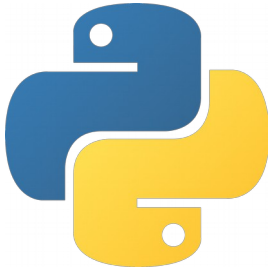
- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
    - Das Paket distutils zielt auf die Distribution von Python-Modulen ab
    - Selbstverständlich könnten Sie einfach den Quellcode des Moduls ins Internet stellen und dazu eine Installationsanweisung liefern
    - Der Installationsprozess kann durch das Paket distutils automatisiert und standardisiert werden, indem sogenannte Distributionen erstellt werden

Python
Erstellen einer Distribution



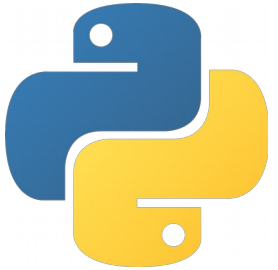
- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
    - Bei einer Distribution unterscheidet man grundsätzlich zwei Typen – 1
      - Eine sogenannte Quellcodedistribution (engl. source distribution) ist ein Archiv, das den Quellcode Ihres Moduls enthält. Zusätzlich zu dem Quellcode existiert ein Installationsscript namens setup.py, das die Installation des Moduls durchführt
      - Der Benutzer braucht diese Art einer Distribution also nur herunterzuladen, zu entpacken und das Installationsscript zu starten
      - Der Vorteil einer Quellcodedistribution ist ihre Plattformunabhängigkeit

Python
Erstellen einer Distribution



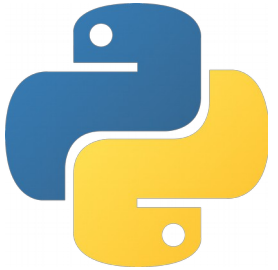
- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
    - Bei einer Distribution unterscheidet man grundsätzlich zwei Typen – 2
      - Eine sogenannte Binärdistribution (engl. binary distribution) ist eine ausführbare Datei, die die Installation Ihres Moduls automatisch durchführt
      - Der Benutzer braucht diese Art einer Distribution also nur herunterzuladen und auszuführen
      - Eine Binärdistribution ist für den Benutzer besonders komfortabel, da er nur zwei Arbeitsschritte auszuführen hat
      - Allerdings bedeutet eine Binärdistribution mehr Aufwand für den Entwickler, denn er muss das Installationsprogramm für verschiedene Plattformen erstellen

Python
Erstellen einer Distribution



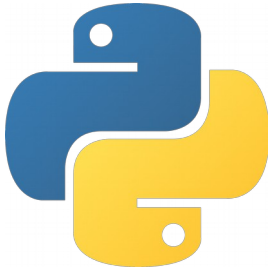
- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
    - Zum Erstellen einer Distribution sind mit dem distutils-Paket im Allgemeinen folgende Arbeitsschritte nötig
      - Schreiben Ihres Moduls oder Pakets
      - Schreiben des Installationsscripts setup.py
      - Erstellen einer Quellcodedistribution bzw. einer Binärdistribution

Python
Erstellen einer Distribution



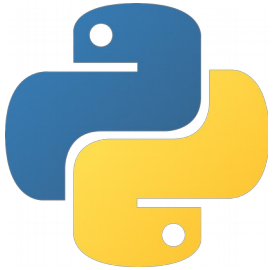
- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
    - Hinweis
      - Grundsätzlich lassen sich mit distutils nicht nur Distributionen von Modulen oder Paketen erstellen, sondern auch von Extensions
      - Solche Extensions können später wie ein Modul oder Paket eingebunden werden, sind aber im Gegensatz zu normalen Modulen oder Paketen in einer anderen Programmiersprache, üblicherweise C oder C++, geschrieben

Python
Erstellen einer Distribution



- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Schreiben des Moduls
    - Rufen Sie sich aber noch einmal ins Gedächtnis, dass es einen Unterschied zwischen einem Modul und einem Paket gibt
    - Während ein Modul aus nur einer Programmdatei besteht, ist ein Paket ein Ordner, der mehrere Untermodule oder -pakete enthalten kann
    - Ein Paket erkennt man an der Programmdatei __init__.py im Paketverzeichnis
    - Die Unterscheidung der Begriffe »Modul« und »Paket« wird beim Erstellen des Installationsscripts noch eine Rolle spielen

Python
Erstellen einer Distribution



- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Schreiben des Moduls

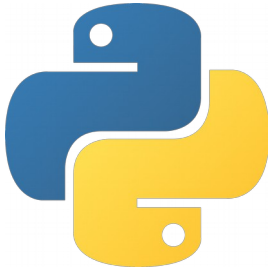
    - An dieser Stelle soll das Beispielmodul entwickelt werden, auf das wir uns die gesamte Zeit beziehen werden

    - Dabei handelt es sich um ein sehr einfaches Modul, das die grundlegende Funktionalität von distutils demonstriert

    - Sinn und Zweck des Beispielmoduls ist es, einen beliebigen Text so zu verändern, dass er sich ähnlich wie dieser liest

      Nach eienr Stidue der Cmabridge Uinverstiaet ist es eagl, in wlehcer Reiehnfogle die Bchustebaen in Woeretrn vokrmomen
      Es ist nur withcig, dsas der ertse und lettze Bchusatbe an der richtgien Stlele snid. Der Rset knan total falcsh sein, und man knan es onhe Porbelme leesn
      Das ist so, wiel das mneschilche Geihrn nihct jeedn Bchustbaen liset, sodnern das Wrot als gaznes

Python
Erstellen einer Distribution

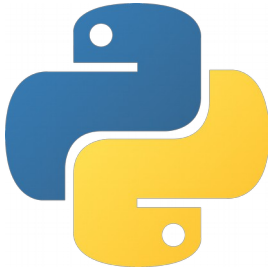


- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Schreiben des Moduls

    - Das Modul stellt dabei eine Funktion verwirble_text bereit, die einen String übergeben bekommt und diesen dann so »verwirbelt« zurückgibt, dass nur der erste und letzte Buchstabe sicher auf ihrem Platz bleiben

```
import random

def verwirble_text(text):
    liste = []
    for wort in text.split():
        w = list(wort[1:-1])
        random.shuffle(w)
        liste.append(wort[0] + "".join(w) + wort[-1])
    return " ".join(liste)
```

Python
Erstellen einer Distribution



- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Schreiben des Moduls
  - Das Installationsscript

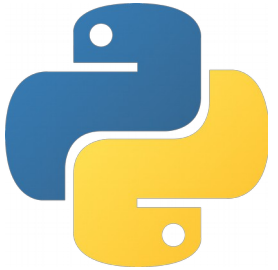
    - Der erste Schritt zur Distribution des eigenen Moduls ist das Erstellen eines Installationsscripts

    - Dies ist eine Python-Programmdatei namens setup.py, über die später das Erstellen der Distribution abläuft

    - Auch die Installation einer Quellcodedistribution aufseiten des Benutzers geschieht durch Aufruf dieser Programmdatei

    - In unserem Beispiel muss im Installationsscript nur die Funktion setup des Moduls distutils.core aufgerufen werden

Python
Erstellen einer Distribution

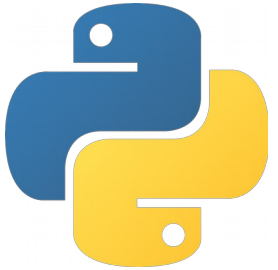


- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Schreiben des Moduls
  - Das Installationsscript

    - Beispiel

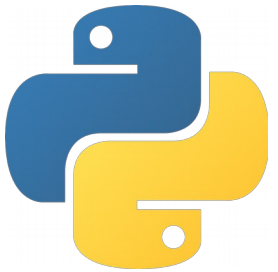
```
from distutils.core import setup

setup(
        name = "verwirbeln",
        version = "1.0",
        author = "Micky Maus",
        author_email = "micky@maus.de",
        py_modules = ["verwirbeln"]
        )
```

Python
Erstellen einer Distribution



- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Schreiben des Moduls
  - Das Installationsscript
    - distutils.core.setup(arguments)
      - Die Funktion setup des Moduls distutils.core muss in der Programmdatei setup.py aufgerufen werden und stößt den jeweils gewünschten Installationsprozess an
      - Dazu müssen Sie der Funktion verschiedene Keyword Arguments übergeben, die Informationen über das Modul bzw. Paket bereitstellen
      - Auf den folgenden Folien, werden Sie eine Liste der wichtigsten möglichen Argumente finden

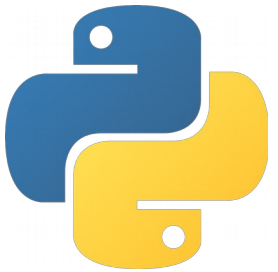
Python
Erstellen einer Distribution



- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Schreiben des Moduls
  - Das Installationsscript
    - distutils.core.setup(arguments)

Parametername	Beschreibung
name	der Name der Distribution
version	die Versionsnummer der Distribution
description	eine kurze Beschreibung der Distribution
long_description	eine ausführliche Beschreibung der Distribution
author	der Name des Autors
author_email	die E-Mail-Adresse des Autors
maintainer	der Name des Paketverwalters (Maintainer), sofern dies nicht der Autor selbst ist
maintainer_email	die E-Mail-Adresse des Paketverwalters

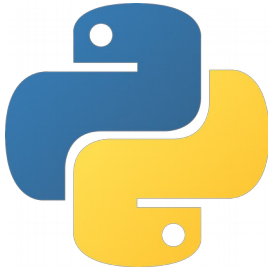
Python
Erstellen einer Distribution



- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Schreiben des Moduls
  - Das Installationsscript
    - distutils.core.setup(arguments)

Parametername	Beschreibung
url	die URL einer Homepage mit weiteren Informationen zur Distribution
download_url	die URL, unter der die Distribution direkt heruntergeladen werden kann
packages	eine Liste von Strings, die die Namen aller Pakete enthält, die in der Distribution enthalten sein sollen
package_dir	Ein Dictionary, über das Pakete in Unterverzeichnissen in die Distribution aufgenommen werden können. Näheres zur Verwendung von *package_dir* folgt weiter unten.

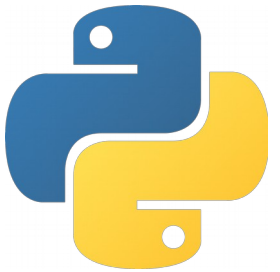
Python
Erstellen einer Distribution



- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Schreiben des Moduls
  - Das Installationsscript
    - distutils.core.setup(arguments)

Parametername	Beschreibung
package_data	Ein Dictionary, über das Dateien, die zu einem Paket gehören, mit in die Distribution aufgenommen werden können. Näheres zur Verwendung von *package_data* finden Sie weiter unten.
py_modules	eine Liste von Strings, die die Namen aller Python-Module enthält, die in der Distribution enthalten sein sollen
scripts	eine Liste von Strings, die die Namen aller Scriptdateien enthält, die in der Distribution enthalten sein sollen
data_files	Eine Liste von Tupeln, über die zusätzliche Dateien in die Distribution mit aufgenommen werden können. Näheres zur Verwendung von *data_files* finden Sie weiter unten.

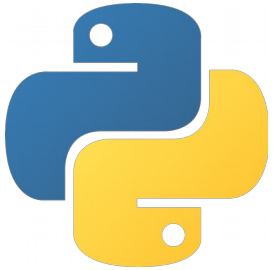
Python
Erstellen einer Distribution



- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Schreiben des Moduls
  - Das Installationsscript
    - distutils.core.setup(arguments)

Parametername	Beschreibung
ext_modules	Eine Liste von `distutils.core.Extension`-Instanzen, die die Namen aller Python-Erweiterungen enthält, die kompiliert werden und in der Distribution enthalten sein sollen. Näheres zu diesem Thema erfahren Sie in Abschnitt 25.2, »Schreiben von Extensions«.
script_name	Der Name des Installationsscripts, das in der Distribution verwendet werden soll. Dieser Parameter ist mit `sys.argv[0]`, also dem Namen des Scripts, vorbelegt, das gerade ausgeführt wird.
license	ein String, der die Lizenz angibt, unter der die Distribution veröffentlicht wird

Python
Erstellen einer Distribution

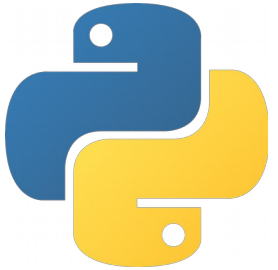


- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Distribution von Paketen

    - Wenn Ihr Projekt statt aus einzelnen Modulen aus einem oder mehreren Paketen besteht, müssen Sie die Namen aller Pakete, die in die Distribution aufgenommen werden sollen, über den Schlüsselwortparameter packages angeben

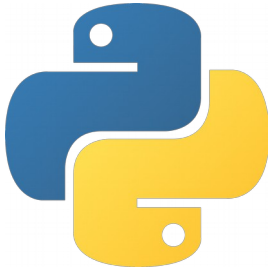
```
from distutils.core import setup

setup(
        [...]
        packages = ["paket1", "paket2",
                                        "paket1.unterpaket1"]
        )
```

Python
Erstellen einer Distribution



- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Distribution von Paketen
    - In diesem Fall werden die Pakete paket1 und paket2, die sich im Hauptverzeichnis befinden müssen, in die Distribution aufgenommen
    - Zusätzlich wird noch das Paket unterpaket1 aufgenommen, das sich innerhalb des Pakets paket1 befindet
    - Sie können durchaus sowohl Pakete über packages als auch einzelne Module über py_modules in die Distribution aufnehmen

Python
Erstellen einer Distribution

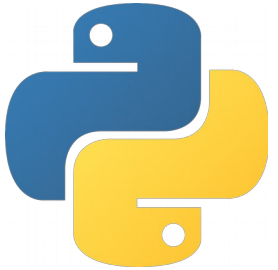


- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Distribution von Paketen

    - Oftmals existiert im Hauptordner neben dem Installationsscript ein Ordner src oder source, in dem sich dann die Module oder Pakete der Distribution befinden

    - Um solch einen Unterordner im Installationsscript bekannt zu machen, übergeben Sie den Schlüsselwortparameter package_dir beim Aufruf von setup

Python
Erstellen einer Distribution

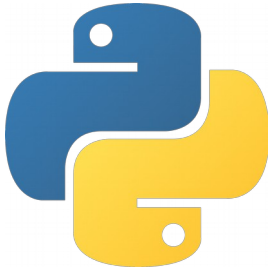


- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Distribution von Paketen

```
from distutils.core import setup

setup(
        [...]
        package_dir = {"" : "src"},
        packages = ["paket1", "paket2",
                                        "paket1.unterpaket1"]
)
```

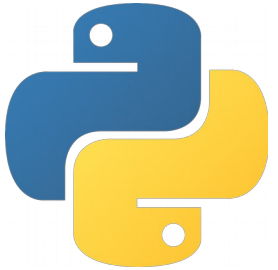
  - Damit wird das Programmverzeichnis ("") auf das Verzeichnis src gelegt

Python
Erstellen einer Distribution



- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Distribution von Paketen
    - Diese Angabe kann auch für einzelne Pakete getätigt werden
    - So können Sie beispielsweise über einen weiteren Eintrag in diesem Dictionary mit dem Schlüssel "paket3" und dem Wert "pfad/zu/meinem/paket/paket3" ein drittes Paket einbinden, das sich in einem anderen Verzeichnis befindet als die beiden Pakete paket1 und paket2
    - Danach kann paket3 über die Liste packages in die Distribution aufgenommen werden
    - Auch Unterpakete von paket3 brauchen dann nicht mehr über den vollständigen Pfad angesprochen zu werden

Python
Erstellen einer Distribution



- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Distribution zusätzlicher Dateien

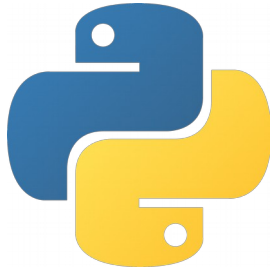
    - Neben Modulen und Paketen gehören möglicherweise weitere Dateien zu Ihrem Projekt und sollten damit auch Platz in der Distribution finden

    - Dazu zählen zunächst einfache Scriptdateien

    - Diese implementieren beispielsweise ein Tool, das im Zusammenhang mit Ihrem Paket steht

    - Der Unterschied zwischen einem Modul und einer Scriptdatei ist, dass das Modul selbst keinen Python-Code ausführt, sondern nur Funktionen oder Klassen bereitstellt, während eine Scriptdatei ein lauffähiges Programm enthält

Python
Erstellen einer Distribution



- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Distribution zusätzlicher Dateien

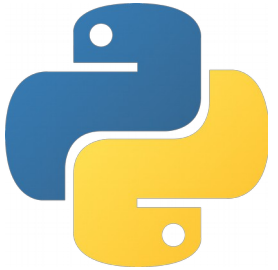
    - Das distutils-Paket installiert Scriptdateien in ein Verzeichnis, in dem sie systemweit ausführbar sind

    - Solche Scriptdateien können beim Aufruf von setup durch den Schlüsselwortparameter scripts übergeben werden

    - Dabei muss für scripts, wie für andere Parameter auch, eine Liste von Strings übergeben werden, die jeweils einen Dateinamen enthalten

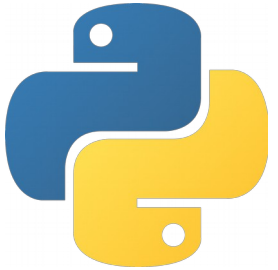
    - Ein kleiner Service, den das Paket distutils in Bezug auf Scriptdateien durchführt, ist das automatische Anpassen der Shebang-Zeile an das Betriebssystem, auf dem die Distribution installiert wird

Python
Erstellen einer Distribution



- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Distribution zusätzlicher Dateien
    - Die nächste Kategorie zusätzlicher Dateien sind Ressourcen, die von bestimmten Paketen benötigt werden und in diesen enthalten sind
    - Beispielsweise erfordert das Paket paket1 die beiden Dateien hallo.txt und welt.txt
    - In einem solchen Fall können diese Dateien über den Schlüsselwortparameter package_data in Form eines Dictionarys übergeben werden

Python
Erstellen einer Distribution

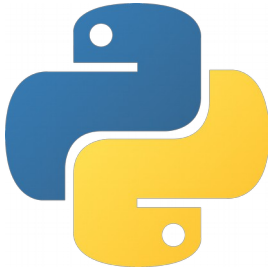


- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Distribution zusätzlicher Dateien

```
setup(
        [...]
        packages = ["paket1", "paket2", "paket1.unterpaket1"],
        package_data = {"paket1" : ["hallo.txt", "welt.txt"]}
        )
```

  - Anstatt jede Datei einzeln anzugeben, können auch Wildcards verwendet werden
  - So würde der Wert ["*.txt"] alle Textdateien einbinden, die sich im Verzeichnis des Pakets paket1 befinden
  - Zu guter Letzt ist es möglich, sonstige Dateien mit in die Distribution aufzunehmen

Python
Erstellen einer Distribution



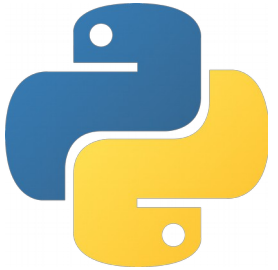
- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Distribution zusätzlicher Dateien

    - Dazu zählen alle Dateien, die in keine der vorherigen Kategorien passen, beispielsweise Konfigurationsdateien, Hilfeseiten oder Ähnliches

    - Diese Dateien können über den Schlüsselwortparameter data_files beim Funktionsaufruf von setup als Liste von Tupeln übergeben werden

```
setup(
        [...]
        data_files = [("grafiken", ["test1.bmp", "test2.bmp"])
                      ("config", ["programm.cfg"])]
        )
```

Python
Erstellen einer Distribution



- Distribution von Python-Projekten
  - Erstellen von Distributionen – distutils
  - Distribution zusätzlicher Dateien

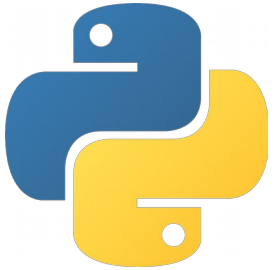
    - In diesem Fall werden die Dateien test1.bmp und test2.bmp aus dem Verzeichnis grafiken sowie die Datei programm.cfg aus dem Verzeichnis config in die Distribution übernommen

    - Die Verzeichnisse verstehen sich relativ zum Pfad des Installationsscripts

    - Hier können Sie durchaus auch absolute Pfade, beispielsweise für eine systemweite Konfigurationsdatei, angeben

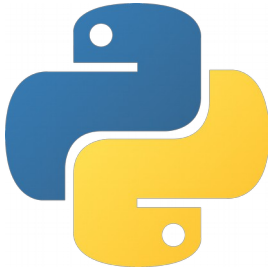
    - Keine Pfadangabe → Ablage Stammverzeichnis
      → Ordnerstruktur wird beibehalten

Python
Erstellen einer Distribution



- Distribution von Python-Projekten
  - Erstellen einer Quellcodedistribution
    - Nachdem Sie das Installationsscript geschrieben haben, können Sie mit dessen Hilfe eine Quellcodedistribution Ihres Pakets oder Moduls erstellen
    - Dazu wechseln Sie in das Verzeichnis, in dem das Installationsscript liegt, und führen es mit dem Argument sdist aus
      setup.py sdist
    - Dieser Befehl erzeugt die Quellcodedistribution im Unterordner dist nach dem Namensschema Projektname-Version.Format
    - Dabei können Sie das Format des Archivs über die Option --formats angeben

Python
Erstellen einer Distribution



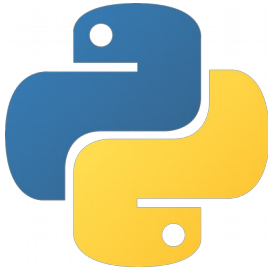
- Distribution von Python-Projekten
  - Erstellen einer Quellcodedistribution

    - Es ist zudem möglich, eine Distribution in mehreren Archivformaten zu erstellen
      setup.py sdist --formats=zip,gztar

    - Mögliche Werte sind dabei
      - zip für ein zip-Archiv (*.zip)
      - gztar für ein gz-komprimiertes tar-Archiv (*.tar.gz)
      - bztar für ein bz2-komprimiertes tar-Archiv (*.tar.bz2)
      - ztar für ein Z-komprimiertes tar-Archiv (*.tar.Z)
      - tar für ein unkomprimiertes tar-Archiv (*.tar)

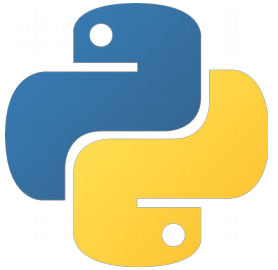
    - Wenn die Option --formats nicht angegeben wurde, wird unter Windows ein zip-Archiv und unter Unix-Systemen ein gz-komprimiertes tar-Archiv erstellt

Python
Erstellen einer Distribution



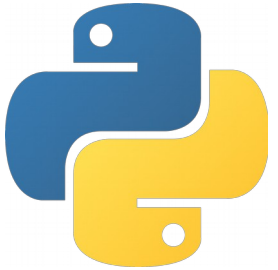
- Distribution von Python-Projekten
  - Erstellen einer Quellcodedistribution
    - In das Archiv werden alle Dateien aufgenommen, die im Installationsscript eingetragen wurden
    - Zusätzlich wird eine Datei namens README oder README.txt automatisch in das Archiv mit aufgenommen, sofern eine solche im selben Ordner wie das Installationsscript existiert
    - Das resultierende Archiv, die Quellcodedistribution, kann jetzt veröffentlicht und verbreitet werden
    - Der Benutzer, der diese Distribution herunterlädt, kann dann Ihr Modul bzw. Ihr Paket installieren

Python
Erstellen einer Distribution



- Distribution von Python-Projekten
  - Erstellen einer Quellcodedistribution
    - Hinweis
      - Beim Erstellen einer Distribution wird eine Datei namens MANIFEST erzeugt
      - Diese Textdatei enthält die Pfade zu allen Dateien, die in die Distribution aufgenommen werden
      - Beim erneuten Erstellen der Distribution werden diese Pfade aus der MANIFEST-Datei wieder ausgelesen, sofern die Datei existiert
      - Wenn das Installationsscript aktueller ist als die MANIFEST-Datei, wird die MANIFEST-Datei beim nächsten Erstellvorgang aktualisiert

Python
Erstellen einer Distribution



- Distribution von Python-Projekten
  - Erstellen einer Quellcodedistribution
    - Hinweis
      - Trotzdem ist es gelegentlich notwendig, dieses Aktualisieren explizit zu erzwingen

        setup.py sdist --force-manifest
        setup.py sdist --manifest-only

      - Mit diesen Aufrufen von setup.py wird das Aktualisieren der MANIFEST-Datei vor dem Erstellen der Distribution erzwungen bzw. ausschließlich die MANIFEST-Datei aktualisiert