PyZufall Dokumentation

Release 0.13.1

davidak

23.07.2015

Inhaltsverzeichnis

Die Python-Bibliothek **PyZufall** beinhaltet Funktionen für das **Generieren zufälliger Daten**.

Die Entwicklung begann als Satzgenerator und wird noch in diese Richtung weitergeführt. Durch die große Sammlung an Funktionen kann sie inzwischen vielfältig eingesetzt werden. So wird sie bereits von anderen Projekten verwendet, wie dem Random VCard-Generator.

PyZufall ist Freie Software. Der Quellcode ist Open Source und steht unter der GPL Version 3.

Das Code Repository und ein Bugtracker sind auf Github.

Jeder ist eingeladen zum Projekt beizutragen!

Ausführliche Informationen sind in der Dokumentation zu finden.

KAPITEL 1

Installation

Mit der Paketverwaltung pip lässt sich die neuste stabile Version installieren.

```
# pip install pyzufall
```

Es läuft mit Python 2.7 und Python 3.

KAPITEL 2

Verwenden

Hier ein einfaches Beispiel, wie du mit **PyZufall** einen zufälligen Namen generieren kannst:

```
from pyzufall.generator import vorname, nachname
name = vorname() + ' ' + nachname()
print(name)
```

In der Dokumentation findest du weitere Beispiele und eine Referenz aller Funktionen.

KAPITEL 3

Inhalt der Dokumentation

3.1 Entstehung

3.1.1 Das Spiel

Als Kind hatte ich bei meiner Oma öfter das Spiel Opa plätschert lustig in der Badewanne gespielt.

Dabei hat jeder Spieler ein Blatt Papier, dass er quer nimmt und als erstes einen Namen oder eine Person darauf schreibt, es an der Stelle faltet, so dass man es nicht mehr lesen kann und es im Uhrzeigersinn weitergibt. Dann schreibt jeder auf das erhaltene Blatt ein Verb, gibt es weiter und schribt ein Adjektiv und nach nochmaligem Weitergeben einen Ort darauf.

So entstehen absurde und zufällige Sätze. Das war immer sehr witzig.

3.1.2 Das Programm

Als ich älter wurde, begann ich Programmieren zu lernen. Dabei hat mich immer begeistert, durch Zufall etwas zu generieren, was teilweise einen Sinn ergibt, aber oft sehr absurd und dadurch lustig ist.

So ist ein Personendatengenerator entstanden, mit dem eine Personendatenbank befüllt wurde.

Auch hab ich ein Script geschrieben, dass Sätze nach dem Muster des Spiels generiert. Erst in Perl und dann in Python. Diese Sätze werden natürlich auf Dauer langweilig.

Da mittlerweile Python die Programmiersprache meiner Wahl ist, habe ich das Script darin weiterentwickelt, mit diversen Satz-Schemata und anderen tollen Funktionen.

3.1.3 Der Satzgenerator

Inzwischen gibt es satzgenerator.de. Auf dieser Webseite werden zufällige Sätze generiert, die bewertet und geteilt werden können.

Für die Generierung der Sätze wird PyZufall genutzt. Die Seite ist auch in Python programmiert, benutzt das Web-Framework Bottle und eine MySQL-Datenbank für die Speicherung der Sätze und Bewertungen.

PyZufall ist Open Source und wird bereits für andere Projekte benutzt.

3.2 Installation

Mit der Paketverwaltung pip lassen sich Python-Pakete vom Python Package Index (PyPI) herunterladen und installieren.

Die neuste stabile Version von **PyZufall** installierst du mit:

```
# pip install pyzufall
```

Hinweis: Pip benötigt Admin-Rechte für die Installation neuer Pakete. Wenn du nicht root bist benutze sudo.

Oder du installierst **PyZufall** manuell, indem du die neuste Version vom PyPI runterlädst, diese entpackst und installierst.

```
# wget https://pypi.python.org/packages/source/P/PyZufall/PyZufall-0.11.tar.gz
# tar -xzf PyZufall-0.11.tar.gz
# cd PyZufall-0.11/
# python3 setup.py install
```

Hinweis: Auch hier brauchst du Admin-Rechte.

3.3 Verwenden

Anhand von Beispielen werden die verschiedenen Module von **PyZufall** vorgestellt.

3.3.1 Name generieren

Um einen Namen zu generieren wird das Modul *pyzufall.generator* verwendet, dass eine Vielzahl von Funktionen für das Generieren von Daten bereitstellt.

```
from pyzufall.generator import vorname, nachname
name = vorname() + ' ' + nachname()
print(name)
```

```
Samira Reschke
```

Siehe auch:

Eine Übersicht aller Funktionen findest du in der Referenz.

3.3.2 Mahlzeit zusammenstellen

Indem du mehrere Funktionen kombinierst, kannst du eine ganze Mahlzeit generieren.

```
from pyzufall.generator import essen, beilage, trinken
s = "Heute Abend gibt es {} mit {} und dazu ein Glas {}.".format(essen(), beilag
print(s)
```

```
Heute Abend gibt es Salzkartoffeln mit Rösti und dazu ein Glas Zuckerschnaps.
```

3.3.3 Meine Band

Auch Funktionen für eine Band gibt es.

```
from pyzufall.generator import bandart, band, vorname
s = "Meine {} heißt '{}' und besteht aus {}, {} und mir.".format(bandart(), band
print(s)
```

```
Meine Ambientband heißt 'Enten bei den Affen im Zoo' und besteht aus Dietmar, Ki
```

3.3.4 Satz generieren

Das Modul *pyzufall.satz* generiert zufällige Sätze nach diversen Schemata.

```
from pyzufall.satz import satz
s = satz()
print(s)
```

```
Der anonyme Vater fällt auf dem Straßenfest auf.
```

Es kann auch eine spezielle Art von Satz generiert werden.

```
from pyzufall.satz import satz_absurde_farbfunktion
s = satz_absurde_farbfunktion()
print(s)
```

```
Violett ist aggressiver als Gelb.
```

3.3.5 Person generieren

Die Klasse *Person* des Moduls *pyzufall.person* generiert eine Person mit diversen Daten. Du kannst den Datensatz der Person ausgeben oder direkt auf jede einzelne Variable zugreifen.

```
from pyzufall.person import Person
p1 = Person()
p2 = Person()
```

```
print(p1)

print("{} und {} sitzen auf einer Bank im Park.".format(p1.vorname, p2.vorname))
print("{} ({}) wohnt in {} und isst gerne {}.".format(p1.vorname, p1.alter, p1.w

del p1, p2
```

```
********************************************************************************
Name: Fred Wittke (die_taube)
Geburtsname: Dornemann
Geschlecht: männlich
Geburtsdatum: 06.01.1912 (103)
Geburtsort: Wittlich
Wohnort: Kaltenkirchen
Beruf: Rentner
E-Mail: fred@die-taube.de
Homepage: http://die-taube.de/
Interessen: Brieftauben züchten
Lieblingsfarbe: Gelb
Lieblingsessen: Frischbraten mit Röstzwiebeln
Motto: Reden ist Silber, Schweigen ist Gold.
********************************************************************************

Fred und Franz sitzen auf einer Bank im Park.
Fred (103) wohnt in Kaltenkirchen und isst gerne Frischbraten mit Röstzwiebeln.
```

Ein praktischer Einsatzweck dafür ist der Random VCard-Generator.

3.4 Beitragen

3.4.1 Datensätze bzw. Worte hinzufügen

Die Vielfalt und Anzahl der möglichen Sätze steigt mit den Datensätzen. Es ist einfach neue Worte oder Satzteile beizutragen und hilft enorm die Sätze noch abwechslungsreicher zu machen.

Bemerkung: An einer einfachen Möglichkeit, Daten hinzuzufügen, wird gearbeitet.

Zu tun

Dokuwiki auf satzgenerator.de/beitragen einrichten mit Kopie der Datensätze. Bearbeiten nach Registrierung möglich.

3.4.2 Fehler berichten

Das Projekt verwendet den Bugtracker auf github um Fehler und Verbesserungsvorschläge zu sammeln.

3.4.3 Programmieren

Wenn du auf github angemeldet bist, kannst du **PyZufall** forken, das Repository runterladen und anfangen neue Funktionen zu programmieren oder den bestehenden Code zu verbessern.

Dann stellst du einen Pull-Request mit deinen Änderungen.

3.5 Benutzer

Hier ist eine Liste mit Projekten, die **PyZufall** verwenden:

- satzgenerator.de
- Random VCard-Generator

Dein Projekt füge ich auch gerne hinzu.

Einfach eine E-Mail mit Beschreibung und Link an post at davidak punkt de.

3.6 Entwicklung

Bei der Entwicklung von **PyZufall** wird git für die Versionierung eingesetzt.

Die Dokumentation wird mit Sphinx erzeugt und die Unittests durch nose ausgeführt.

3.6.1 Code Status

Hier wird das Ergebnis der automatischen Unittests und die Coverage des Codes im Repository angezeigt:

3.6.2 Repository runterladen

```
$ git clone https://github.com/davidak/pyzufall.git
```

3.6.3 Dokumentation erzeugen

Mit folgendem Befehl wird die HTML-Version sowie LaTeX samt PDF erzeugt.

```
$ make docs
```

Einzeln geht das mit:

```
$ make html
$ make pdf
```

3.6.4 Unittests ausführen

Um den Code auf deinem System zu testen, führe folgenden Befehl im heruntergeladenen Repository aus:

```
imac:PyZufall davidak$ nosetests --with-doctest
..................................................
----------------------------------------------------------------------
Ran 50 tests in 0.867s

OK
```

3.6.5 Vor dem Release

- Versionsnummer in `version.py` überprüfen, niemals 'dev' auf PyPI laden!
- Changelog aktualisieren, Versionsnummer und Datum überprüfen
- Unittests ausführen:

```
$ nosetests --with-doctest
```

- Dokumentation bauen und überprüfen:

```
$ make docs
```

- Installation von PyPI mit pip testen: https://wiki.python.org/moin/TestPyPI

  Account registrieren: https://testpypi.python.org/pypi?%3Aaction=register_form (wird von Zeit zu Zeit gelöscht)

  Paket nach testpypi hochladen:

```
$ python setup.py register --repository https://testpypi.python.org/pypi

$ python3 setup.py sdist upload -r https://testpypi.python.org/pypi
```

  Prüfen auf Syntax-Fehler in der README: https://testpypi.python.org/pypi/PyZufall

  Einmal testweise installieren:

```
$ pip install -i https://testpypi.python.org/pypi pyzufall
```

3.6.6 Release

- letzte Änderungen in git einchecken
- git tag mit Versionsnummer setzen
- push auf github
- auf PyPI veröffentlichen:

```
$ python3 setup.py sdist upload
```

3.6.7 Nach dem Release

- Versionsnummer inkrementieren + 'dev'
- am nächsten Release arbeiten :)

3.7 Referenz der Module

Die Module mit ihren Klassen und Funktionen.

3.7.1 pyzufall.helfer

Stellt diverse Hilfsfunktionen bereit.

pyzufall.helfer.**alter**(*geburtsdatum*)
Berechnet das Alter in Jahren anhand des Geburtsdatums.

Rückgabetyp int

Neu in Version 0.12.

pyzufall.helfer.**aufzaehlung**(*liste*)
Erzeugt eine grammatikalisch korrekte Aufzählung aus einer Liste.

Beispiel:

```
>>> a = ['lesen', 'reiten', 'Freunde treffen']
```

```
>>> s = aufzaehlung(a)
```

```
>>> print(s)
lesen, reiten und Freunde treffen
```

```
>>> b = ['Überwachen', 'Strafen']
```

```
>>> s = aufzaehlung(b)
```

```
>>> print(s)
Überwachen und Strafen
```

```
>>> c = ['schlafen']
```

```
>>> s = aufzaehlung(c)
```

```
>>> print(s)
schlafen
```

Parameter **liste** (*list*) – Eine Liste von Strings.

Rückgabetyp string

Neu in Version 0.12.

pyzufall.helfer.**chance**(*wahrscheinlichkeit*, *wert*)
Der übergebene Wert wird mit der gewählten Wahrscheinlichkeit zurückgegeben.

Neu in Version 0.11.

Parameter

- **wahrscheinlichkeit** – int zwischen 1 und 100
- **wert** – string

pyzufall.helfer.**erste_gross**(*s*)
Macht den ersten Buchstaben gross.

Beispiele:

```
>>> a = erste_gross('das ist ein Beispiel?')
```

```
>>> print(a)
Das ist ein Beispiel?
```

```
>>> b = erste_gross('über Stock und Stein.')
```

```
>>> print(b)
Über Stock und Stein.
```

```
>>> c = erste_gross('älter als das Internet!')
```

```
>>> print(c)
Älter als das Internet!
```

pyzufall.helfer.**lese**(*dateiname*)
Liest die Textdatei mit dem übergebenen Namen aus data/ zeilenweise ein und gib eine Liste zurück.

Beispiel:

```
>>> liste = lese('baeume.txt')
```

http://stackoverflow.com/questions/10174211/make-an-always-relative-to-current-module-file-path

Parameter **dateiname** (*string*) – Dateiname inklusive Endung, z.B. *vornamen.txt*

Rückgabetyp list

pyzufall.helfer.**str_add**(*wort*, *string*)

Fügt einen String ans Ende eines Wortes an, ohne doppelte Buchstaben zu erzeugen.

Beispiele:

```
>>> a = str_add('feige', 'er')
```

```
>>> print(a)
feiger
```

```
>>> b = str_add('feige', 'e')
```

```
>>> print(b)
feige
```

```
>>> c = str_add('blöd', 'e')
```

```
>>> print(c)
blöde
```

Neu in Version 0.11.

pyzufall.helfer.**uml**(*s*)

Ersetzt Umlaute durch die entsprechenden 2 Buchstaben.

Beispiel:

```
>>> uml('Käse')
'Kaese'
>>> uml('Brötchen')
'Broetchen'
>>> uml('Gefühl')
'Gefuehl'
```

Neu in Version 0.13.

3.7.2 pyzufall.generator

Stellt diverse Generator-Funktionen zur Verfügung.

pyzufall.generator.**adjektiv**()

Gibt ein Adjektiv zurück.

pyzufall.generator.**band**()
Gibt einen fiktiven Bandnamen zurück.

pyzufall.generator.**bandart**()
Gibt eine Bandart zurück.

Beispiel: 'Gothic Metal Band'

pyzufall.generator.**baum**()
Gibt einen Baum zurück.

pyzufall.generator.**beilage**()
Gibt eine Beilage zum Essen zurück.

pyzufall.generator.**beruf_m**()
Gibt eine männliche Berufsbezeichnung zurück.

pyzufall.generator.**beruf_w**()
Gibt eine weibliche Berufsbezeichnung zurück.

pyzufall.generator.**color**()
Gibt eine Farbe auf englisch zurück.

pyzufall.generator.**datum**()
Gibt ein gültiges Datum zwischen <vor 50 Jahren> und <heute> zurück.

Rückgabetyp string

pyzufall.generator.**email**(*vorname*, *nachname*, *nick=''*, *domain=''*)
Generiert eine E-Mail-Adresse.

Beispiel: ismail@ismail-christ.eu, emelieeiru63@lebanese.cc, nic@copacabana.com

Neu in Version 0.13.

pyzufall.generator.**essen**()
Gibt ein Essen zurück.

pyzufall.generator.**farbe**()
Gibt eine Farbe zurück.

pyzufall.generator.**firma**()
Gibt einen fiktiven Firmenname zurück.

Zu tun

Funktion programmieren

---

pyzufall.generator.**geburtsdatum**()
Gibt ein gültiges Datum zwischen <vor 110 Jahren> und <heute> zurück.

Rückgabetyp string

pyzufall.generator.**gegenstand**()
Gibt einen Gegenstand zurück.

pyzufall.generator.**geschlecht**()
Gibt ein zufälliges Geschlecht zurück.

1 = männlich 0 = weiblich

2011 gibt es laut Statistik 51,18% weibliche Personen in Deutschland: https://www.destatis.de/DE/ZahlenFakten/GesellschaftStaat/Bevoelkerung/Bevoelkerungsstand/Tabellen/Z

Rückgabetyp int

pyzufall.generator.**homepage**(*vorname*, *nachname*, *nick=''*)
Gibt die Domain einer persönlichen Homepage zurück.

Beispiel: lilim.eu, heruntergekommene-kastanie.net, damian-schuett.org

Neu in Version 0.13.

pyzufall.generator.**interesse**()
Gibt ein zufälliges Interesse bzw Hobby zurück.

Veraltet ab Version 0.11: Wird durch pyzufall.generator.interessen_liste() ersetzt.

pyzufall.generator.**koerperteil**()
Gibt ein Körperteil zurück.

pyzufall.generator.**nachname**()
Gibt einen Nachnamen zurück.

pyzufall.generator.**nickname**(*vorname=''*, *nachname=''*)
Generiert einen Nickname, Angabe von Vor- und Nachname optional.

Beispiel: dicker_falke, beate_brutal85, stinkender_panda24

Neu in Version 0.13.

pyzufall.generator.**objekt**()
Gibt ein Objekt zurück.

pyzufall.generator.**objekt_m**(*s*)
Bringt ein Objekt in Berzug zu einer männlichen Person.

Beispiel: 'der Bär' wird zu 'den Bären' oder 'seinen Bären'

pyzufall.generator.**objekt_w**(*s*)
Bringt ein Objekt in Berzug zu einer weiblichen Person.

Beispiel: 'der Bär' wird zu 'den Bären' oder 'ihren Bären'

pyzufall.generator.**ort**()
Gibt eine Ortsangabe zurück.

Beispiel: 'im Flur'

Zu tun

aufteilen in generator und zufällige aus liste

---

pyzufall.generator.**person**()
Gibt eine zufällige Person zurück.

pyzufall.generator.**person_m**()
Gibt eine männliche Person zurück.

pyzufall.generator.**person_objekt_m**()
Gibt eine Person als Objekt in Bezug auf eine männliche Person zurück.

Beispiel: seine Mitarbeiterin

pyzufall.generator.**person_objekt_w**()
Gibt eine Person als Objekt in Bezug auf eine weibliche Person zurück.

Beispiel: ihre Mutter

pyzufall.generator.**person_w**()
Gibt eine weibliche Person zurück.

pyzufall.generator.**pflanze**()
Gibt eine Pflanze zurück.

pyzufall.generator.**sprichwort**()
Gibt ein Sprichwort zurück.

pyzufall.generator.**stadt**()
Gibt eine Stadt zurück.

pyzufall.generator.**stadt_bl**()
Gibt eine Stadt mit Bundesland zurück.

pyzufall.generator.**tier**()
Gibt ein Tier zurück.

pyzufall.generator.**trinken**()
Gibt ein Getränk zurück.

pyzufall.generator.**url**(*domain*)

```
>>> url('davidak.de')
'http://davidak.de/'
```

Neu in Version 0.13.

pyzufall.generator.**verbd**()
Gibt ein ditransitives Verb zurück.

Beschreibung auf Wikipedia

pyzufall.generator.**verbi**()
Gibt ein intransitives Verb zurück.

Beschreibung auf Wikipedia

pyzufall.generator.**verbi2**()
Gibt ein intransitives, getrenntes Verb zurück.

Beschreibung auf Wikipedia

pyzufall.generator.**verbn**()
Gibt ein nullwertiges Verb zurück.

Beschreibung auf Wikipedia

pyzufall.generator.**verbt**()
Gibt ein transitives Verb zurück.

Beschreibung auf Wikipedia

pyzufall.generator.**verbt2**()
Gibt ein intransitives, getrenntes Verb zurück.

Beschreibung auf Wikipedia

pyzufall.generator.**vorname**()
Gibt einen Vornamen zurück.

pyzufall.generator.**vorname_m**()
Gibt einen männlichen Vornamen zurück.

pyzufall.generator.**vorname_w**()
Gibt einen weiblichen Vornamen zurück.

pyzufall.generator.**wort**(*laenge=7*)
Gibt ein Fantasiewort zurück.

Parameter **laenge** – int

pyzufall.generator.**zahl**()
Gibt eine Zahl zwischen 0 und 100 zurück.

Rückgabetyp string

pyzufall.generator.**zeitpunkt**(*start*, *ende*)
Gibt einen zufälligen Zeitpunkt (datetime.date) zwischen zwei Zeitpunkten zurück. Es handelt sich dabei um ein gültiges Datum.

Neu in Version 0.13.

Param start, ende: datetime

Rückgabetyp datetime

3.7.3 pyzufall.person

Stellt die Klasse Person zur Verfügung. Mit ihr kann man ein Objekt erzeugen, dass eine Person mit zufällig generierten Daten darstellt.

Es kann auf jedes Attribut einzeln zugegriffen werden oder mit print(person) alle aufeinmal ausgeben werden.

Die generierten Daten basieren teilweise auf statistischen Werten und versuchen möglichst authentisch zu sein.

Quellen für Statistiken:

- https://www.destatis.de/
- http://de.statista.com/
- https://www.zensus2011.de/
- http://www.statistik2013.de/

class pyzufall.person.**Person**
Generiert Daten einer zufälligen und fiktiven Person.

Neu in Version 0.9.

3.7.4 pyzufall.satz

Stellt Funktionen bereit, die Sätze nach diversen Satz-Schemata generieren.

Die Funktion *pyzufall.satz.satz()* beinhaltet alle.

pyzufall.satz.**satz**()
Generiert einen zufälligen Satz.

20% Standard-Sätze, 20% Fragen und 60% Themen-Sätze

pyzufall.satz.**satz_absurde_farbfunktion**()
Generiert einen Satz nach folgendem Muster: Gelb ist brauner als Türkis.

pyzufall.satz.**satz_adjektiv_am_ort**()
Generiert einen Satz nach dem Muster: <Ort> <Verb> <Person> <Adjektiv>.

Beispiel: Auf dem Spielplatz ist die Freundin hilfsbereit.

pyzufall.satz.**satz_adjektiv_sprichwort**()
Generiert einen Satz nach dem Muster: Je untrainierter desto lächerlicher.

pyzufall.satz.**satz_arbeit**()
Generiert einen Satz über eine berufstätige Person.

Beispiel: Achmed, der Grafiker aus Waldheim, spielt den Nasenbär.

pyzufall.satz.**satz_band**()
Generiert einen zufälligen Satz zum Thema Band.

pyzufall.satz.**satz_band_besetzung**()
Generiert einen Satz mit den Mitgliedern einer Band.

Beispiel: Die Black Metal Band “Die Oralen Nudeln” besteht aus Marlene, Gert, Stefanie, Timm, Andrej, Friederike und Dorothea.

pyzufall.satz.**satz_band_gegruendet**()
Generiert einen Satz, der den Zeitpunkt einer Bandgründung zum Thema hat.

Beispiel: Die Electroband “Kartoffel auf dem Klo” wurde am 26.10.2009 in Selb gegründet.

pyzufall.satz.**satz_band_mitglied**()
Generiert einen Satz, in dem ein Bandmitglied vorgestellt wird.

Beispiel: Annelise ist Gitarristin von der Gothicband “Kräuter in der Innenstadt”.

pyzufall.satz.**satz_baum**()
Generiert einen Satz mit dem Thema Baum.

Beispiel: Die gnadenlose Kerstin tritt gegen den Apfelbaum.

pyzufall.satz.**satz_essen**()
Generiert einen Satz mit Essen und/oder Trinken.

Beispiel: Die Wärterin isst Orangen mit Mayonnaise und trinkt dazu Milch.

pyzufall.satz.**satz_farbe**()
Generiert einen Satz nach dem Muster: Braun ist eine unsittliche Farbe.

pyzufall.satz.**satz_folgehandlung**()
Generiert einen Satz, der eine Folgehandlung beschreibt.

Beispiel: Ohne dass Irmgard überlebt, bricht sie aggressiv ein.

pyzufall.satz.**satz_frage**()
Generiert eine zufällige Frage.

pyzufall.satz.**satz_frage_1**()
Generiert eine Frage nach dem Grund, aus dem eine Person eine Tätigkeit ausführt

Beispiel: Wieso fällt dein Partner in Gedanken hin?

pyzufall.satz.**satz_frage_2**()
Generiert eine Frage nach der Person, die eine Tätigkeit ausführt.

Beispiel: Wer telefoniert bewusstlos in der Abtei?

pyzufall.satz.**satz_frage_3**()
Generiert eine Frage nach dem Ort, an dem eine Person eine Tätigkeit ausführt.

Beispiel: Wo singt ein Siebdrucker?

pyzufall.satz.**satz_frage_4**()
Generiert eine Frage nach der Art, wie eine Person eine Tätigkeit ausführt.

Beispiel: Wie wird sie beim ersten Date angefasst?

pyzufall.satz.**satz_frage_5**()
Generiert eine Frage nach dem Zeitpunkt, an dem eine Person eine Tätigkeit ausführt.

Beispiel: Wann säuft eine Hure?

pyzufall.satz.**satz_freunde_lieben**()
Generiert einen Satz über eine Person mit Eigenschaften.

Beispiel: In der Garage ist das Mannsweib lesbisch.

pyzufall.satz.**satz_interessen**()
Generiert einen Satz über die Interessen einer Person.

pyzufall.satz.**satz_kloster**()
Generiert einen Satz über eine Person in einem Kloster.

Beispiel: Bruder Ludwig ist der böseste Mönch im Kloster.

pyzufall.satz.**satz_koerperteil**()
Generiert einen Satz zum Thema Körperteile.

Beispiel: Die ekelhafte Oma massiert ihren Fuß.

pyzufall.satz.**satz_nulltransitiv**()
Generiert einen Satz mit einem nulltransitiven Verb.

Beispiel: Im Park schneit es.

pyzufall.satz.**satz_standard**()
Generiert einen zufälligen Standard-Satz.

pyzufall.satz.**satz_standard_1**()
Generiert einen einfachen Satz nach dem Muster: <Person> <Verb> <Adjektiv> <Ort>.

Beispiel: Die Geschmacklose bepisst sich cool in der Kirche.

pyzufall.satz.**satz_standard_2**()
Generiert einen einfachen Satz nach dem Muster: <Ort> <Verb> <Person> <Adjektiv>.

Beispiel: Beim ersten Date flieht er.

pyzufall.satz.**satz_standard_3**()
Generiert einen einfachen Satz nach dem Muster: <Adjektiv> <Verb> <Person> <Ort>.

Beispiel: Gehirntot weint die Schädlingsbekämpferin in der Psychiatrie.

pyzufall.satz.**satz_standard_4**()
Generiert einen einfachen Satz nach dem Muster: <Person> <Verb> <Person/Objekt> <Adjektiv> <Ort>.

Beispiel: Der Ruhige raubt ein Schaf aus.

pyzufall.satz.**satz_thema**()
Generiert einen Satz zu einem zufälligen Thema.

3.8 Changelog

Hier sind die Änderungen jeder Version dokumentiert.

3.8.1 Version 0.13.1

Veröffentlicht am 08.07.2015

- fix install with python3

3.8.2 Version 0.13

Veröffentlicht am 14.03.2015

- Kompatibilität mit Python 2.7
- automatische Unittests mit Travis-CI und Test Coverage
- *pyzufall.generator.zeitpunkt()* hinzugefügt
- *pyzufall.generator.nickname()* hinzugefügt
- *pyzufall.generator.homepage()* hinzugefügt
- *pyzufall.helfer.uml()* hinzugefügt
- *pyzufall.generator.url()* hinzugefügt
- *pyzufall.generator.email()* hinzugefügt
- veraltete Funktionen entfernt
- Dokumentation überarbeitet
- Viele Verbesserungen am Code und Fehler behoben

3.8.3 Version 0.12

Veröffentlicht am 14.01.2014

- *pyzufall.satz.satz_interessen()* hinzugefügt
- *pyzufall.helfer.aufzaehlung()* ersetzt `pyzufall.generator.interessen_liste`
- Keine Vergewaltigungen mehr. Das ist nicht witzig! Es braucht auch nicht über 20 Synonyme für Geschlechtsverkehr und Selbstbefriedigung. Durch diese Änderungen wird die Seriosität deutlich gesteigert!
- Dokumentation verbessert und aktualisiert
- Viele Verbesserungen am Code
- Viele kleine Fehler behoben

3.8.4 Version 0.11

Veröffentlicht am 22.09.2013

- Funktion *pyzufall.helfer.chance()* hinzugefügt und `pyzufall.helfer.e25()` etc dadurch ersetzt
- Funktion `pyzufall.generator.interessen_liste()` hinzugefügt. Sie ersetzt *pyzufall.generator.interesse()*.
- Funktion *pyzufall.helfer.str_add()* mit Unittests hinzugefügt

- Ungleiche Elemente aus Listen werden jetzt mit der Funktion `random.sample()` generiert.
- Doctests in Modulen hinzugefügt
- *Makefile* erstellt
- *setup.py* und *MANIFEST.in* hinzugefügt
- Dokumentation und README angepasst
- Seite Entwicklung zur Dokumentation hinzugefügt
- Sphinx Parameter zu Docstrings hinzufügen
- viele kleine Fehlerbehebungen und Verbesserungen

3.8.5 Version 0.10.3

Veröffentlicht am 15.09.2013

- Dateien mit Datensätzen die Endung .txt gegeben, um deren Erweiterbarkeit hervorzuheben und spätere Bearbeitung zu vereinfachen
- LICENSE wieder ohne .rst, weil es nicht in reStructuredText formatiert ist

3.8.6 Version 0.10.2

Veröffentlicht am 15.09.2013

- Changelog hinzugefügt und in Dokumentation eingebunden
- Dokumentation erweitert
- README und LICENSE auch mit reStructuredText formatiert anstatt Markdown, um einheitlich mit der Dokumentation zu sein

3.8.7 Version 0.10.1

Veröffentlicht am 13.09.2013

- Dokumentation an die neue Struktur angepasst
- Fehler in `pyzufall.person._gen_interessen()` behoben

3.8.8 Version 0.10

Veröffentlicht am 13.09.2013

- Projekt umstrukturiert: **pyzufall** als Paket in mehrere Module aufgeteilt

3.8.9 Version 0.9

Veröffentlicht am 23.08.2013

- jedes Satz-Schema als Funktion
- Unittests mit nose hinzugefügt
- Modul person hinzugefügt
- README.md hinzugefügt
- LICENSE.md hinzugefügt mit GPLv3
- TODO-Seite in Dokumentation hinzugefügt, auf der Hinweise im Quelltext aufgelistet werden
- Entstehung zur Dokumentation hinzugefügt
- Struktur der Dokumentation angepasst
- viele kleine Änderungen

3.8.10 Version 0.8

Veröffentlicht am 23.07.2013

- Dokumentation mit Sphinx hinzugefügt
- Docstring für jede Funktion hinzugefügt

Vor der Version 0.8 gab es keine Versionsnummern, sie wurde als gefühlter Entwicklungsstand vergeben.

Alle Änderungen können den Kommentaren der Commits im Repository entnommen werden.

Der erste Commit war am 27.08.2012.

p