
PyZufall Dokumentation

Release 0.12

dauidak

14.03.2015

1	Entstehung	2
1.1	Das Spiel	2
1.2	Das Programm	2
1.3	Der Satzgenerator	2
2	Installation	4
3	Verwenden	5
3.1	Name generieren	5
3.2	Satz generieren	5
3.3	Person generieren	5
4	Beitragen	7
4.1	Datensätze bzw. Worte hinzufügen	7
4.2	Fehler berichten	7
4.3	Programmieren	7
5	Benutzer	8
6	Entwicklung	9
6.1	Repository runterladen	9
6.2	Dokumentation erzeugen	9
6.3	Unittests ausführen	9
6.4	Vor dem Release	10
6.5	Release	11
6.6	Nach dem Release	11
7	Referenz der Module	12
7.1	pyzufall.helfer	12
7.2	pyzufall.generator	14
7.3	pyzufall.person	17
7.4	pyzufall.satz	18
8	Changelog	21
8.1	Version 0.12	21

8.2	Version 0.11	21
8.3	Version 0.10.3	22
8.4	Version 0.10.2	22
8.5	Version 0.10.1	22
8.6	Version 0.10	22
8.7	Version 0.9	23
8.8	Version 0.8	23

Python-Modulindex **24**

Die Python-Bibliothek **PyZufall** beinhaltet diverse Funktionen für das **Generieren zufälliger Daten** wie Namen, Berufe, Bandnamen, ein Datum, Sätze oder den kompletten Datensatz einer Personen.

Die Entwicklung begann als Satzgenerator für <http://satzgenerator.de/> und wird noch in diese Richtung weitergeführt. Sie soll aber auch eine generelle Sammlung von Funktionen zur Generierung zufälliger Daten darstellen. So wird sie schon von [anderen Projekten](#) verwendet, wie dem [Random VCard-Generator](#).

PyZufall ist [freie und Open Source Software](#). Es steht unter der [GPL Version 3](#).

Den Quelltext findest du im [Repository](#) und Fehler sowie Verbesserungsvorschläge werden im [Bugtracker](#) gesammelt.

Jeder ist eingeladen zum Projekt beizutragen!

Entstehung

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, das 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 schreibt ein Adjektiv und nach nochmaligem Weitergeben einen Ort darauf.

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

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, das 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.

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.

Installation

Mit `pip` kannst du die neuste stabile Version von **PyZufall** ganz einfach auf deinem System installieren:

```
$ sudo pip install pyzufall
```

Bemerkung: Meist ruft der Befehl `pip` für Python 2.x auf. **PyZufall** wird hauptsächlich für Python 3 entwickelt, bei einem schnellen Test mit Python 2.6.x und 2.7.x sind allerdings keine Fehler aufgefallen.

Pip für **Python 3** wird auf vielen System (z.B OS X oder Ubuntu) wie folgt ausgeführt:

```
$ sudo pip-3.2 install pyzufall
```

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

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

Verwenden

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

3.1 Name generieren

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

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

Siehe auch:

Eine Übersicht aller Funktionen findest du in der *Referenz*.

3.2 Satz generieren

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

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

3.3 Person generieren

Die Klasse `Person` des Moduls `pyzufall.person` generiert eine Person mit diversen Daten. Du kannst alle Daten der Person auf einmal 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: Tilmann Ried
Geschlecht: männlich
Geburtsdatum: 19.04.1981 (32)
Geburtsort: Schwelm
Wohnort: Starnberg
Beruf: arbeitslos
Interessen: Sonnenstudio, Mofatuning
Lieblingsfarbe: Grün
Lieblingsessen: Schokoladen
Motto: Du sollst den Tag nicht vor dem Abend loben.
*****

Tilmann und Emre sitzen auf einer Bank im Park.
Tilmann (32) wohnt in Starnberg und isst gerne Schokoladen.
```

Ein praktischer Einsatzzweck dafür ist der [Random VCard-Generator](#).

Beitragen

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.

4.2 Fehler berichten

Das Projekt verwendet den [Bugtracker auf github](#) um Fehler und Verbesserungsvorschläge zu sammeln.

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.

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 dauidak punkt de](mailto:post@dauidakpunkt.de) oder das [Kontaktformular](#) benutzen.

Entwicklung

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

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

6.1 Repository runterladen

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

6.2 Dokumentation erzeugen

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

```
$ make docs
```

Einzelne gehen so mit:

```
$ make html
```

```
$ make pdf
```

6.3 Unittests ausführen

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

```
imac:PyZufall davidak$ make test
Doctest: pyzufall.generator.adjektiv ... ok
Doctest: pyzufall.generator.band ... ok
Doctest: pyzufall.generator.bandart ... ok
Doctest: pyzufall.generator.baum ... ok
Doctest: pyzufall.generator.beilage ... ok
Doctest: pyzufall.generator.beruf_m ... ok
```

```
Doctest: pyzufall.generator.beruf_w ... ok
Doctest: pyzufall.generator.color ... ok
Doctest: pyzufall.generator.datum ... ok
Doctest: pyzufall.generator.essen ... ok
Doctest: pyzufall.generator.farbe ... ok
Doctest: pyzufall.generator.geburtsdatum ... ok
Doctest: pyzufall.generator.gegenstand ... ok
Doctest: pyzufall.generator.geschlecht ... ok
Doctest: pyzufall.generator.interesse ... ok
Doctest: pyzufall.generator.interesten_liste ... ok
Doctest: pyzufall.generator.koerperteil ... ok
Doctest: pyzufall.generator.nachname ... ok
Doctest: pyzufall.generator.ort ... ok
Doctest: pyzufall.generator.person_m ... ok
Doctest: pyzufall.generator.person_objekt_m ... ok
Doctest: pyzufall.generator.person_objekt_w ... ok
Doctest: pyzufall.generator.person_w ... ok
Doctest: pyzufall.generator.pflanze ... ok
Doctest: pyzufall.generator.spruchwort ... ok
Doctest: pyzufall.generator.stadt ... ok
Doctest: pyzufall.generator.stadt_bl ... ok
Doctest: pyzufall.generator.tier ... ok
Doctest: pyzufall.generator.trinken ... ok
Doctest: pyzufall.generator.verbd ... ok
Doctest: pyzufall.generator.verbi ... ok
Doctest: pyzufall.generator.verbi2 ... ok
Doctest: pyzufall.generator.verbn ... ok
Doctest: pyzufall.generator.verbt ... ok
Doctest: pyzufall.generator.verbt2 ... ok
Doctest: pyzufall.generator.vorname_m ... ok
Doctest: pyzufall.generator.vorname_w ... ok
Doctest: pyzufall.generator.wort ... ok
Doctest: pyzufall.generator.zahl ... ok
Doctest: pyzufall.helfer.erste_gross ... ok
Doctest: pyzufall.helfer.lese ... ok
Doctest: pyzufall.helfer.str_add ... ok
test.test_satz ... ok
test.test_satz_frage ... ok
test.test_spruchwort ... ok
```

Ran 45 tests in 0.907s

OK

6.4 Vor dem Release

- Versionsnummer in `version.py` überprüfen, niemals ‘dev’ auf PyPI laden

- *Changelog* aktualisieren, Versionsnummer und Datum überprüfen
- Unittests ausführen:

```
$ make test
```
- Dokumentation bauen und überprüfen:

```
$ make docs
```

6.5 Release

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

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

6.6 Nach dem Release

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

Referenz der Module

Die Module mit ihren Klassen und Funktionen.

7.1 pyzufall.helfer

Stellt diverse Hilfsfunktionen bereit.

`pyzufall.helfer.aufzaehlung` (*liste*)

Erzeugt eine grammatikalisch korrekte Aufzählung aus einer Liste.

Beispiel:

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

```
>>> aufzaehlung(a)
'lesen, reiten und Freunde treffen'
```

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

```
>>> aufzaehlung(b)
'Überwachen und Strafen'
```

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

```
>>> aufzaehlung(c)
'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.e16(wert)`

Der übergebene Wert wird mit einer Wahrscheinlichkeit von 16% zurückgegeben.

Veraltet ab Version 0.11: Wird durch `pyzufall.helfer.chance()` ersetzt.

`pyzufall.helfer.e25(wert)`

Der übergebene Wert wird mit einer Wahrscheinlichkeit von 25% zurückgegeben.

Veraltet ab Version 0.11: Wird durch `pyzufall.helfer.chance()` ersetzt.

`pyzufall.helfer.e50(wert)`

Der übergebene Wert wird mit einer Wahrscheinlichkeit von 50% zurückgegeben.

Veraltet ab Version 0.11: Wird durch `pyzufall.helfer.chance()` ersetzt.

`pyzufall.helfer.e75(wert)`

Der übergebene Wert wird mit einer Wahrscheinlichkeit von 75% zurückgegeben.

Veraltet ab Version 0.11: Wird durch `pyzufall.helfer.chance()` ersetzt.

`pyzufall.helfer.erste_gross(s)`

Macht den ersten Buchstaben gross.

Beispiele:

```
>>> erste_gross('das ist ein Beispiel?')
'Das ist ein Beispiel?'
```

```
>>> erste_gross('über Stock und Stein.')
'Über Stock und Stein.'
```

```
>>> erste_gross('älter als das Internet!')
'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*

Rückgabety `list`

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

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

Beispiele:

```
>>> str_add('feige', 'er')  
'feiger'
```

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

```
>>> str_add('blöd', 'e')  
'blöde'
```

Neu in Version 0.11.

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 dem 01.01.1950 und 31.12.2012 zurück.

Rückgabotyp string

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

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

`pyzufall.generator.firma()`
Gibt einen fiktiven Firmennamen zurück.

Zu tun

Funktion programmieren

`pyzufall.generator.geburtsdatum()`
Gibt ein gültiges Datum zwischen dem 01.01.1910 und 31.12.2012 zurück.

Rückgabotyp 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/>

Rückgabotyp int

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

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

`pyzufall.generator.interesten_liste(anzahl=2)`
Gibt eine Liste von Interessen als String zurück.

Ohne Angabe der Anzahl werden 2 Interessen zurückgegeben.

Neu in Version 0.11.

Veraltet ab Version 0.12.

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

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

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

`pyzufall.generator.objekt_m(s)`
Bringt ein Objekt in Bezug zu einer männlichen Person.

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

`pyzufall.generator.objekt_w(s)`
Bringt ein Objekt in Bezug 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.spruchwort()`

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.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 zufälligen Vornamen zurück.

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

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

`pyzufall.generator.wort()`
Gibt ein Fantasiewort zurück.

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

Rückgabety string

7.3 pyzufall.person

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

Es kann auf jedes Attribut einzeln zugegriffen werden oder mit `print(person)` alle auf einmal ausgegeben 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.

alter ()

Berechnet das Alter der übergebenen Person.

Rückgabety `int`

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 bipsst 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.

Changelog

Hier sind die Änderungen jeder Version dokumentiert.

8.1 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

8.2 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

8.3 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

8.4 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

8.5 Version 0.10.1

Veröffentlicht am 13.09.2013

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

8.6 Version 0.10

Veröffentlicht am 13.09.2013

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

8.7 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

8.8 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

`pyzufall.generator`, 14
`pyzufall.helfer`, 12
`pyzufall.person`, 17
`pyzufall.satz`, 18

A

adjektiv() (im Modul pyzufall.generator), 14
alter() (Methode von pyzufall.person.Person),
18
aufzaehlung() (im Modul pyzufall.helfer), 12

B

band() (im Modul pyzufall.generator), 14
bandart() (im Modul pyzufall.generator), 14
baum() (im Modul pyzufall.generator), 14
beilage() (im Modul pyzufall.generator), 14
beruf_m() (im Modul pyzufall.generator), 14
beruf_w() (im Modul pyzufall.generator), 14

C

chance() (im Modul pyzufall.helfer), 12
color() (im Modul pyzufall.generator), 14

D

datum() (im Modul pyzufall.generator), 14

E

e16() (im Modul pyzufall.helfer), 13
e25() (im Modul pyzufall.helfer), 13
e50() (im Modul pyzufall.helfer), 13
e75() (im Modul pyzufall.helfer), 13
erste_gross() (im Modul pyzufall.helfer), 13
essen() (im Modul pyzufall.generator), 14

F

farbe() (im Modul pyzufall.generator), 14
firma() (im Modul pyzufall.generator), 14

G

geburtsdatum() (im Modul pyzufall.generator), 15

gegenstand() (im Modul pyzufall.generator),
15
geschlecht() (im Modul pyzufall.generator),
15

I

interesse() (im Modul pyzufall.generator), 15
interessen_liste() (im Modul pyzufall.generator), 15

K

koerperteil() (im Modul pyzufall.generator),
15

L

lese() (im Modul pyzufall.helfer), 13

N

nachname() (im Modul pyzufall.generator), 15

O

objekt() (im Modul pyzufall.generator), 15
objekt_m() (im Modul pyzufall.generator), 15
objekt_w() (im Modul pyzufall.generator), 15
ort() (im Modul pyzufall.generator), 15

P

Person (Klasse in pyzufall.person), 18
person() (im Modul pyzufall.generator), 16
person_m() (im Modul pyzufall.generator), 16
person_objekt_m() (im Modul pyzufall.generator), 16
person_objekt_w() (im Modul pyzufall.generator), 16
person_w() (im Modul pyzufall.generator), 16
pflanze() (im Modul pyzufall.generator), 16
pyzufall.generator (Modul), 14

pyzufall.helfer (Modul), 12
 pyzufall.person (Modul), 17
 pyzufall.satz (Modul), 18

S

satz() (im Modul pyzufall.satz), 18
 satz_absurde_farbfunktion() (im Modul pyzufall.satz), 18
 satz_adjektiv_am_ort() (im Modul pyzufall.satz), 18
 satz_adjektiv_sprichwort() (im Modul pyzufall.satz), 18
 satz_arbeit() (im Modul pyzufall.satz), 18
 satz_band() (im Modul pyzufall.satz), 18
 satz_band_besetzung() (im Modul pyzufall.satz), 18
 satz_band_gegruendet() (im Modul pyzufall.satz), 18
 satz_band_mitglied() (im Modul pyzufall.satz), 18
 satz_baum() (im Modul pyzufall.satz), 19
 satz_essen() (im Modul pyzufall.satz), 19
 satz_farbe() (im Modul pyzufall.satz), 19
 satz_folgehandlung() (im Modul pyzufall.satz), 19
 satz_frage() (im Modul pyzufall.satz), 19
 satz_frage_1() (im Modul pyzufall.satz), 19
 satz_frage_2() (im Modul pyzufall.satz), 19
 satz_frage_3() (im Modul pyzufall.satz), 19
 satz_frage_4() (im Modul pyzufall.satz), 19
 satz_frage_5() (im Modul pyzufall.satz), 19
 satz_freunde_lieben() (im Modul pyzufall.satz), 19
 satz_interessen() (im Modul pyzufall.satz), 19
 satz_kloster() (im Modul pyzufall.satz), 19
 satz_koerperteil() (im Modul pyzufall.satz), 20
 satz_nulltransitiv() (im Modul pyzufall.satz), 20
 satz_standard() (im Modul pyzufall.satz), 20
 satz_standard_1() (im Modul pyzufall.satz), 20
 satz_standard_2() (im Modul pyzufall.satz), 20
 satz_standard_3() (im Modul pyzufall.satz), 20
 satz_standard_4() (im Modul pyzufall.satz), 20

satz_thema() (im Modul pyzufall.satz), 20
 sprichwort() (im Modul pyzufall.generator), 16
 stadt() (im Modul pyzufall.generator), 16
 stadt_bl() (im Modul pyzufall.generator), 16
 str_add() (im Modul pyzufall.helfer), 13

T

tier() (im Modul pyzufall.generator), 16
 trinken() (im Modul pyzufall.generator), 16

V

verbd() (im Modul pyzufall.generator), 16
 verbi() (im Modul pyzufall.generator), 16
 verbi2() (im Modul pyzufall.generator), 16
 verbn() (im Modul pyzufall.generator), 17
 verbt() (im Modul pyzufall.generator), 17
 verbt2() (im Modul pyzufall.generator), 17
 vorname() (im Modul pyzufall.generator), 17
 vorname_m() (im Modul pyzufall.generator), 17
 vorname_w() (im Modul pyzufall.generator), 17

W

wort() (im Modul pyzufall.generator), 17

Z

zahl() (im Modul pyzufall.generator), 17