
pyzufall Dokumentation

Release 0.11

dauidak

21. 09. 2013

Inhaltsverzeichnis

1	Entstehung	2
1.1	Das Spiel	2
1.2	Das Programm	2
1.3	Der Satzgenerator	2
2	Installation	4
2.1	Manuelle Installation	4
2.2	Unittests	4
3	Verwenden	6
3.1	Name generieren	6
3.2	Satz generieren	6
3.3	Person generieren	6
4	Beitragen	8
4.1	Datensätze bzw. Worte hinzufügen	8
4.2	Fehler berichten	8
4.3	Programmieren	8
5	Benutzer	9
6	Entwicklung	10
6.1	Repository	10
6.2	Vor dem Release	10
6.3	Release	10
6.4	Nach dem Release	11
7	Referenz der Module	12
7.1	pyzufall.helfer	12
7.2	pyzufall.generator	13
7.3	pyzufall.person	16
7.4	pyzufall.satz	16

8	Changelog	19
8.1	Version 0.11	19
8.2	Version 0.10.3	19
8.3	Version 0.10.2	20
8.4	Version 0.10.1	20
8.5	Version 0.10	20
8.6	Version 0.9	20
8.7	Version 0.8	21
	Python-Modulindex	22
	Stichwortverzeichnis	23

Das Python-Paket **pyzufall** generiert zufällige Daten wie Namen, Berufe, Bandnamen, ein Datum, Sätze oder den Datensatz einer Personen.

Es wurde ursprünglich für satzgenerator.de entwickelt, kann aber vielfältig eingesetzt werden.

Pyzufall ist Open Source, der Quelltext kann auf github.com heruntergeladen werden.

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, 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.

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.

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

Die aktuellste Version von **pyzufall** findest du auf [github](#).

Eine stabile Version wird es auf [PyPI](#) geben.

Bemerkung: Derzeit befindet sich **pyzufall** in aktiver Entwicklung. Daher kann sich die Struktur oder Funktionsnamen noch ändern.

2.1 Manuelle Installation

Klone das git-Repository:

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

Bemerkung: Dafür muss [git](#) installiert sein.

Danach kopierst du das Paket **pyzufall** in dein Projekt-Verzeichnis:

```
$ mkdir meinprojekt
$ mv pyzufall/pyzufall meinprojekt
$ cd meinprojekt
```

2.2 Unittests

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

```
imac:pyzufall davidak$ nosetests -v
test.test_test ... ok
test_helfer.test_lese ... ok
```

```
test_helfer.test_erste_gross ... ok
test_satz.test_satz ... ok
test_satz.test_satz_frage ... ok
```

```
-----
Ran 5 tests in 0.822s
```

OK

Bemerkung: Für die Unittests wird `nose` verwendet und muss installiert sein.

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, dass 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(p1.vorname + " und " + p2.vorname + " sitzen auf einer Bank im Park.\n")
print(p2.vorname + " (" + str(p2.alter) + ") wohnt in " + p2.wohnort + ".")

*****
Name: Kornelia Eismann
Geschlecht: weiblich
Geburtsdatum: 27.04.1974 (39)
Geburtsort: Geesthacht
Wohnort: Halle
Beruf: Technische Zeichnerin
Interessen: Würfelspiele
Lieblingsfarbe: Weiß
Lieblingsessen: Steak
Motto: Augen auf beim Eierkauf.
*****

Kornelia und Thorge sitzen auf einer Bank im Park.

Thorge (58) wohnt in Stadthagen.
```

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

Dein Projekt füge ich auch gerne hinzu.

Einfach eine E-Mail mit Beschreibung und Link an 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

<https://github.com/davidak/pyzufall>

6.2 Vor dem Release

- Versionsnummer in `version.py` überprüfen
- *Changelog* aktualisieren, Versionsnummer und Datum überprüfen
- Unittests ausführen:

```
$ make test
```

- Dokumentation bauen:

```
$ make docs
```

6.3 Release

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

```
$ python setup.py sdist upload
```

6.4 Nach dem Release

- Versionsnummer inkrementieren
- am nächsten Release arbeiten :)

Referenz der Module

Die Module mit ihren Klassen und Funktionen.

7.1 pyzufall.helfer

`pyzufall.helfer.chance` (*wahrscheinlichkeit, wert*)

Der übergebene Wert wird mit der gewählten Wahrscheinlichkeit zurückgegeben. Neu in Version 0.11.

`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.

`pyzufall.helfer.lese` (*dateiname*)

Liest die Textdatei mit dem übergebenen Namen aus data/ zeilenweise ein und gib eine Liste zurück.

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

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

Rückgabe eine Liste, die den Zeilen der Textdatei entspricht

Rückgabetyp list

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

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

Beispiele:

- blöd + e = blöde
- feige + e = feige
- feige + er = feiger

Neu in Version 0.11.

7.2 pyzufall.generator

`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.

`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.

`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/>

`pyzufall.generator.interesse()`
Gibt ein zufälliges Interesse bzw Hobby zurück. Veraltet ab Version 0.11: Wird durch `pyzufall.generator.interessen_liste()` ersetzt.

`pyzufall.generator.interessen_liste(anzahl=1)`
Gibt eine Liste von Interessen als String zurück.

Ohne Angabe der Anzahl wird ein Interesse zurückgegeben. Neu in Version 0.11.

`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.

7.3 pyzufall.person

class `pyzufall.person.Person`

Generiert Daten einer zufälligen und fiktiven Person. Neu in Version 0.9.

7.4 pyzufall.satz

`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_spruchwort()`

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 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 Selv 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_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 beisst 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 in jeder Version dokumentiert.

8.1 Version 0.11

Veröffentlicht am

- Funktion `pyzufall.helfer.chance()` hinzugefügt und `pyzufall.helfer.e25()` etc dadurch ersetzt
- Funktion `pyzufall.generator.interesten_liste()` hinzugefügt. Sie ersetzt `pyzufall.generator.interesse()`.
- Funktion `pyzufall.helfer.str_add()` mit Unittests hinzugefügt #14
- Ungleiche Elemente aus Listen werden jetzt mit der Funktion `random.sample()` generiert.
- Unittests aufgeteilt
- viele kleine Fehlerbehebungen und Verbesserungen
- Dokumentation und README angepasst
- Seite *Entwicklung* zur Dokumentation hinzugefügt
- Mit der Sphinx-Erweiterung `sphinxcontrib-issuetracker` können jetzt Issues von github in der Dokumentation verlinkt werden
- Makefile erstellt #3

8.2 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

8.3 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

8.4 Version 0.10.1

Veröffentlicht am 13.09.2013

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

8.5 Version 0.10

Veröffentlicht am 13.09.2013

- Projekt umstrukturiert: pyzufall als Paket in mehrere Module aufgeteilt

8.6 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

8.7 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.

Python-Modulindex

p

`pyzufall.generator`, [13](#)
`pyzufall.helfer`, [12](#)
`pyzufall.person`, [16](#)
`pyzufall.satz`, [16](#)

Stichwortverzeichnis

A

adjektiv() (in Modul pyzufall.generator), 13

B

band() (in Modul pyzufall.generator), 13

bandart() (in Modul pyzufall.generator), 13

baum() (in Modul pyzufall.generator), 13

beilage() (in Modul pyzufall.generator), 13

beruf_m() (in Modul pyzufall.generator), 13

beruf_w() (in Modul pyzufall.generator), 13

C

chance() (in Modul pyzufall.helfer), 12

color() (in Modul pyzufall.generator), 13

D

datum() (in Modul pyzufall.generator), 13

E

e16() (in Modul pyzufall.helfer), 12

e25() (in Modul pyzufall.helfer), 12

e50() (in Modul pyzufall.helfer), 12

e75() (in Modul pyzufall.helfer), 12

erste_gross() (in Modul pyzufall.helfer), 12

essen() (in Modul pyzufall.generator), 13

F

farbe() (in Modul pyzufall.generator), 13

firma() (in Modul pyzufall.generator), 13

G

geburtsdatum() (in Modul pyzufall.generator),
14

gegenstand() (in Modul pyzufall.generator),
14

geschlecht() (in Modul pyzufall.generator), 14

I

interesse() (in Modul pyzufall.generator), 14

interessen_liste() (in Modul pyzufall.generator), 14

K

koerperteil() (in Modul pyzufall.generator), 14

L

lese() (in Modul pyzufall.helfer), 12

N

nachname() (in Modul pyzufall.generator), 14

O

objekt() (in Modul pyzufall.generator), 14

objekt_m() (in Modul pyzufall.generator), 14

objekt_w() (in Modul pyzufall.generator), 14

ort() (in Modul pyzufall.generator), 14

P

Person (Klasse in pyzufall.person), 16

person() (in Modul pyzufall.generator), 15

person_m() (in Modul pyzufall.generator), 15

person_objekt_m() (in Modul pyzufall.generator), 15

person_objekt_w() (in Modul pyzufall.generator), 15

person_w() (in Modul pyzufall.generator), 15

pflanze() (in Modul pyzufall.generator), 15

pyzufall.generator (Modul), 13

pyzufall.helfer (Modul), 12

pyzufall.person (Modul), 16

pyzufall.satz (Modul), [16](#)

S

[satz\(\)](#) (in Modul pyzufall.satz), [16](#)
[satz_absurde_farbfunktion\(\)](#) (in Modul pyzufall.satz), [16](#)
[satz_adjektiv_am_ort\(\)](#) (in Modul pyzufall.satz), [16](#)
[satz_adjektiv_spruchwort\(\)](#) (in Modul pyzufall.satz), [16](#)
[satz_arbeit\(\)](#) (in Modul pyzufall.satz), [16](#)
[satz_band\(\)](#) (in Modul pyzufall.satz), [16](#)
[satz_band_besetzung\(\)](#) (in Modul pyzufall.satz), [17](#)
[satz_band_gegruendet\(\)](#) (in Modul pyzufall.satz), [17](#)
[satz_band_mitglied\(\)](#) (in Modul pyzufall.satz), [17](#)
[satz_baum\(\)](#) (in Modul pyzufall.satz), [17](#)
[satz_essen\(\)](#) (in Modul pyzufall.satz), [17](#)
[satz_farbe\(\)](#) (in Modul pyzufall.satz), [17](#)
[satz_folgehandlung\(\)](#) (in Modul pyzufall.satz), [17](#)
[satz_frage\(\)](#) (in Modul pyzufall.satz), [17](#)
[satz_frage_1\(\)](#) (in Modul pyzufall.satz), [17](#)
[satz_frage_2\(\)](#) (in Modul pyzufall.satz), [17](#)
[satz_frage_3\(\)](#) (in Modul pyzufall.satz), [17](#)
[satz_frage_4\(\)](#) (in Modul pyzufall.satz), [17](#)
[satz_frage_5\(\)](#) (in Modul pyzufall.satz), [18](#)
[satz_freunde_lieben\(\)](#) (in Modul pyzufall.satz), [18](#)
[satz_kloster\(\)](#) (in Modul pyzufall.satz), [18](#)
[satz_koerperteil\(\)](#) (in Modul pyzufall.satz), [18](#)
[satz_nulltransitiv\(\)](#) (in Modul pyzufall.satz), [18](#)
[satz_standard\(\)](#) (in Modul pyzufall.satz), [18](#)
[satz_standard_1\(\)](#) (in Modul pyzufall.satz), [18](#)
[satz_standard_2\(\)](#) (in Modul pyzufall.satz), [18](#)
[satz_standard_3\(\)](#) (in Modul pyzufall.satz), [18](#)
[satz_standard_4\(\)](#) (in Modul pyzufall.satz), [18](#)
[satz_thema\(\)](#) (in Modul pyzufall.satz), [18](#)
[spruchwort\(\)](#) (in Modul pyzufall.generator), [15](#)
[stadt\(\)](#) (in Modul pyzufall.generator), [15](#)
[stadt_bl\(\)](#) (in Modul pyzufall.generator), [15](#)
[str_add\(\)](#) (in Modul pyzufall.helfer), [13](#)

T

[tier\(\)](#) (in Modul pyzufall.generator), [15](#)

[trinken\(\)](#) (in Modul pyzufall.generator), [15](#)

V

[verbd\(\)](#) (in Modul pyzufall.generator), [15](#)
[verbi\(\)](#) (in Modul pyzufall.generator), [15](#)
[verbi2\(\)](#) (in Modul pyzufall.generator), [15](#)
[verbn\(\)](#) (in Modul pyzufall.generator), [15](#)
[verbt\(\)](#) (in Modul pyzufall.generator), [16](#)
[verbt2\(\)](#) (in Modul pyzufall.generator), [16](#)
[vorname\(\)](#) (in Modul pyzufall.generator), [16](#)
[vorname_m\(\)](#) (in Modul pyzufall.generator), [16](#)
[vorname_w\(\)](#) (in Modul pyzufall.generator), [16](#)

W

[wort\(\)](#) (in Modul pyzufall.generator), [16](#)

Z

[zahl\(\)](#) (in Modul pyzufall.generator), [16](#)