

Faker库——测试数据源

Python 中有个测试数据库，叫做 Faker，它可以自动帮我们来生成各种各样的看起来很真的“假”数据

1 安装

```
pip install faker
```

2 使用

```
from faker import Faker

faker=Faker()
print('name:',faker.name())
print('address:',faker.address())
print('text:',faker.text())

'''
name: Steve Thomas
address: 10082 Hall Meadow Apt. 382
South Justinchester, IN 40238
text: More moment how understand fast party. After administration visit now deep
candidate. Floor oil on mission success eat office. Group of message strategy
certain question long.
'''
```

3 解析

3.1 数据语言

在创建Faker对象时，可以指定数据的语言代号，从而产生不同国家的测试数据。默认英文en-US

- 简体中文: zh_CN
- 繁体中文: zh_TW
- 美国英文: en_US
- 英国英文: en_GB
- 德文: de_DE
- 日文: ja_JP
- 韩文: ko_KR
- 法文: fr_FR

```
faker=Faker('zh-CN')
```

3.2 随机种子

```
fake.seed(integer)
```

3.3 Provider

这个 faker 库在设计上，为了解耦，将 Provider 对象做成了 Faker 对象的“插件”。Faker 可以添加一个个 Provider 对象，Provider 对象为 Faker 对象提供了生成某项数据的核心实现。就相当于 Faker 对象是一个生成器，它的生成功能依赖于 Provider，是 Provider 提供给了 Faker 对象生成某项数据的能力。其内置了如下默认的 Provider 对象。

自定义Provider

```
from faker import Faker
fake = Faker()

# first, import a similar Provider or use the default one
from faker.providers import BaseProvider

# create new provider class. Note that the class name _must_ be ``Provider``.
class Provider(BaseProvider):
    def foo(self):
        return 'bar'

# then add new provider to faker instance
fake.add_provider(Provider)

# now you can use:
fake.foo()
# 'bar'
```

第三方Provider

```
from faker import Faker
from faker.providers import internet

print('image url:', faker.image_url())
print('explorer:', faker.internet_explorer())
print('IPv4:', faker.ipv4())
print('IPv6:', faker.ipv6())
print('IPv4 private:', faker.ipv4_private())
print('IPv4 public:', faker.ipv4_public())
```

Address

用于生成一些和地址相关的数据，如地址、城市、邮政编码、街道等内容

```
faker.providers.address
```

```
fake.address()
# '河北省兰州县沙湾兴城路Q座 403829'
```

```
fake.building_number()
# 'Y座'

fake.city()
# '巢湖县'

fake.city_name()
# '六安'

fake.city_suffix()
# '县'

fake.country()
# '威克岛'

fake.country_code(representation="alpha-2")
# 'DE'

fake.district()
# '大兴'

fake.postcode()
# '438682'

fake.province()
# '河北省'

fake.street_address()
# '陆路t座'

fake.street_name()
# '马路'

fake.street_suffix()
# '街'
```

Color

用于生成和颜色相关的数据，如 HEX、RGB、RGBA 等格式的颜色

```
faker.providers.color
```

```
fake.color_name()
# 'DarkOrchid'

fake.hex_color()
# '#4d9393'

fake.rgb_color()
# '57,27,251'

fake.rgb_css_color()
# 'rgb(50,239,246)'

fake.safe_color_name()
# 'lime'
```

```
fake.safe_hex_color()
# '#77ee00'
```

Company

用于生成公司相关数据，如公司名、公司前缀、公司后缀等内容

```
faker.providers.company
```

```
fake.color_name()
# 'DarkOrchid'

fake.hex_color()
# '#4d9393'

fake.rgb_color()
# '57,27,251'

fake.rgb_css_color()
# 'rgb(50,239,246)'

fake.safe_color_name()
# 'lime'

fake.safe_hex_color()
# '#77ee00'
```

Credit card

用于生成信用卡相关数据，如过期时间、银行卡号、安全码等内容

```
faker.providers.credit_card
```

```
fake.credit_card_expire(start="now", end="+10y", date_format="%m/%y")
# '08/21'

fake.credit_card_full(card_type=None)
# 'Diners Club / Carte Blanche\n慧 李\n36612062559471 09/27\nCVC: 454\n'

fake.credit_card_number(card_type=None)
# '4730083086312'

fake.credit_card_provider(card_type=None)
# 'VISA 16 digit'

fake.credit_card_security_code(card_type=None)
# '503'
```

Date time

用于生成时间相关数据，如年份、月份、星期、出生日期等内容，可以返回 `datetime` 类型的数据

```
faker.providers.date_time
```

```
fake.am_pm()
# 'PM'

fake.century()
# 'X'

fake.date(pattern="%Y-%m-%d", end_datetime=None)
# '2017-01-16'

fake.date_between(start_date="-30y", end_date="today")
# datetime.date(1995, 11, 11)

fake.date_between_dates(date_start=None, date_end=None)
# datetime.date(2019, 8, 7)

fake.date_object(end_datetime=None)
# datetime.date(1998, 9, 6)

fake.date_of_birth(tzinfo=None, minimum_age=0, maximum_age=115)
# datetime.date(1964, 10, 11)

fake.date_this_century(before_today=True, after_today=False)
# datetime.date(2018, 2, 18)

fake.date_this_decade(before_today=True, after_today=False)
# datetime.date(2017, 4, 9)

fake.date_this_month(before_today=True, after_today=False)
# datetime.date(2019, 8, 1)

fake.date_this_year(before_today=True, after_today=False)
# datetime.date(2019, 6, 11)

fake.date_time(tzinfo=None, end_datetime=None)
# datetime.datetime(1990, 4, 24, 15, 20, 45)

fake.date_time_ad(tzinfo=None, end_datetime=None, start_datetime=None)
# datetime.datetime(1531, 9, 20, 12, 12, 12)

fake.date_time_between(start_date="-30y", end_date="now", tzinfo=None)
# datetime.datetime(2011, 2, 2, 9, 37, 1)

fake.date_time_between_dates(datetime_start=None, datetime_end=None,
tzinfo=None)
# datetime.datetime(2019, 8, 7, 14, 30, 25)

fake.date_time_this_century(before_now=True, after_now=False, tzinfo=None)
# datetime.datetime(2015, 6, 18, 8, 48, 6)

fake.date_time_this_decade(before_now=True, after_now=False, tzinfo=None)
# datetime.datetime(2015, 4, 23, 10, 45, 30)

fake.date_time_this_month(before_now=True, after_now=False, tzinfo=None)
# datetime.datetime(2019, 8, 2, 19, 48, 10)

fake.date_time_this_year(before_now=True, after_now=False, tzinfo=None)
# datetime.datetime(2019, 7, 14, 12, 19, 6)
```

```
fake.day_of_month()
# '13'

fake.day_of_week()
# 'wednesday'

fake.future_date(end_date="+30d", tzinfo=None)
# datetime.date(2019, 8, 17)

fake.future_datetime(end_date="+30d", tzinfo=None)
# datetime.datetime(2019, 8, 13, 13, 11, 44)

fake.iso8601(tzinfo=None, end_datetime=None)
# '1977-05-15T14:46:13'

fake.month()
# '01'

fake.month_name()
# 'December'

fake.past_date(start_date="-30d", tzinfo=None)
# datetime.date(2019, 7, 11)

fake.past_datetime(start_date="-30d", tzinfo=None)
# datetime.datetime(2019, 7, 13, 7, 26, 20)

fake.time(pattern="%H:%M:%S", end_datetime=None)
# '15:21:15'

fake.time_delta(end_datetime=None)
# datetime.timedelta(0)

fake.time_object(end_datetime=None)
# datetime.time(2, 45, 45)

fake.time_series(start_date="-30d", end_date="now", precision=None,
distrib=None, tzinfo=None)
# <generator object Provider.time_series at 0x7f648857b480>

fake.timezone()
# 'Africa/Banjul'

fake.unix_time(end_datetime=None, start_datetime=None)
# 1014648347

fake.year()
# '1975'
```

Geo

用于生成和地理位置相关的数据，包括经纬度，时区等等信息

```
faker.providers.geo
```

```
fake.coordinate(center=None, radius=0.001)
# Decimal('137.852599')

fake.latitude()
# Decimal('-19.9540745')

fake.latlng()
# (Decimal('-88.937396'), Decimal('65.246894'))

fake.local_latlng(country_code="US", coords_only=False)
# ('39.78504', '-85.76942', 'Greenfield', 'US', 'America/Indiana/Indianapolis')

fake.location_on_land(coords_only=False)
# ('-38.65333', '178.00417', 'Gisborne', 'NZ', 'Pacific/Auckland')

fake.longitude()
# Decimal('113.209515')
```

Internet

用于生成和互联网相关的数据，包括随机电子邮箱、域名、IP 地址、URL、用户名、后缀名等内容

`faker.providers.internet`

```
fake.ascii_company_email(*args, **kwargs)
# 'zwei@78.net'

fake.ascii_email(*args, **kwargs)
# 'jinna@hotmail.com'

fake.ascii_free_email(*args, **kwargs)
# 'guiyinglei@hotmail.com'

fake.ascii_safe_email(*args, **kwargs)
# 'jingdong@example.org'

fake.company_email(*args, **kwargs)
# 'tangtao@xiao.cn'

fake.domain_name(levels=1)
# 'li.cn'

fake.domain_word(*args, **kwargs)
# 'yang'

fake.email(*args, **kwargs)
# 'xia56@gmail.com'

fake.free_email(*args, **kwargs)
# 'juanliang@gmail.com'

fake.free_email_domain(*args, **kwargs)
# 'gmail.com'

fake.hostname(*args, **kwargs)
# 'srv-70.fangna.cn'
```

```
fake.image_url(width=None, height=None)
# 'https://placeimg.com/318/385/any'

fake.ipv4(network=False, address_class=None, private=None)
# '1.216.79.223'

fake.ipv4_network_class()
# 'c'

fake.ipv4_private(network=False, address_class=None)
# '192.168.186.133'

fake.ipv4_public(network=False, address_class=None)
# '124.100.210.40'

fake.ipv6(network=False)
# '4ec9:9df:dd92:aec4:2b2b:7c29:a486:7961'

fake.mac_address()
# 'a4:5d:79:b9:12:24'

fake.safe_email(*args, **kwargs)
# 'yan04@example.net'

fake.slug(*args, **kwargs)
# ''

fake.tld()
# 'org'

fake.uri()
# 'http://www.yongqiang.cn/login/'

fake.uri_extension()
# '.asp'

fake.uri_page()
# 'author'

fake.uri_path(deep=None)
# 'explore/categories/category'

fake.url(schemes=None)
# 'https://www.qi.com/'

fake.user_name(*args, **kwargs)
# 'hzhu'
```

Job

用于生成和职业相关的数据

```
faker.job()
# '烫工'
```


Lorem

用于生成一些假文字数据，包括句子、自然段、长文本、关键词等，另外可以传入不同的参数来控制生成的长度

```
faker.providers.lorem
```

```
fake.paragraph(nb_sentences=3, variable_nb_sentences=True, ext_word_list=None)
# '只要联系虽然的是使用任何.但是方面方法不断.就是现在增加操作.'
```

```
fake.paragraphs(nb=3, ext_word_list=None)
# ['如果参加责任有些工作.如何为什经验.文章城市以及因此今天.自己个人标题更多当前因为.',
#  '电子注意方式评论部分个人质量.不断人民生活地方以后为了法律.一下还有质量能力大小.',
#  '非常有关支持程序学习.一起内容音乐积分不要今年表示.']
```

```
fake.sentence(nb_words=6, variable_nb_words=True, ext_word_list=None)
# '安全系列网络以后增加.'
```

```
fake.sentences(nb=3, ext_word_list=None)
# ['项目能力日期部门没有为了电子责任.', '可是全部美国论坛广告语言.', '所有因此进行有些.']
```

```
fake.text(max_nb_chars=200, ext_word_list=None)
# ('也是现在大学但是今天.价格发展如果用户一般产品事情.进行音乐日期计划.下载规定北京她的这样.\n'
#  '公司日本方法现在其实登录.管理注册其实信息其他起来完成.\n'
#  '看到说明浏览一个作者控制类型.大家所以标准.\n'
#  '计划也是孩子.同时单位部分的是管理比较怎么.\n'
#  '政府如果只有出现生产.系列当然更新行业汽车只要比较.精华支持来源日本.\n'
#  '完成而且浏览内容我们.计划比较网上技术报告.')
```

```
fake.texts(nb_texts=3, max_nb_chars=200, ext_word_list=None)
# ['中心处理手机他们什么不要参加.帮助完成部门技术可能你的管理.\n'
#  '服务必须目前规定发展日本.为了信息工作回复在线发展登录.是一地址生活在线发生.\n'
#  '经济由于评论类别原因因此有些不同.主题中国一个而且社区.\n'
#  '有关发表当然发现专业责任所以.音乐自己因此.质量浏览就是北京.\n'
#  '以上一下发布威望公司是否.质量标准客户.可是学生深圳成功情况政府客户.\n'
#  '新闻然后服务学习文化教育怎么查看.',
#  '有限都是他们喜欢只有他的的话.联系积分情况.\n'
#  '支持状态通过社会孩子提供没有标准.你们孩子不同业务虽然很多一样.\n'
#  '您的网站用户本站社会这里.能够希望不要情况.全部他的感觉电子点击您的发表.而且东西我的运行位置.\n'
#  '一样那个系列是一能够如此使用.这里不断网络文章而且.音乐东西帖子登录表示这个.\n'
#  '提供直接美国没有.社区标准免费内容国际所有分析.\n'
#  '一切单位法律更多开发社区.城市部分分析支持.',
#  '影响价格发生怎么有关开始.应用威望会员发表技术教育成为.\n'
#  '介绍方法网站一点.所有男人在线市场进行用户.影响还是主要参加建设.\n'
#  '功能我的孩子决定工程任何其他.因为认为方面不过查看登录.\n'
#  '准备需要为了会员中国国家各种.的人文件客户为了工作.以后需要更多今年是一用户操作不同.\n'
#  '时候上海能力一定最后设备关于.如果联系政府设计学生报告为什.是一计划一些会员为什女人.']
```

```
fake.word(ext_word_list=None)
# '通过'
```

```
fake.words(nb=3, ext_word_list=None, unique=False)
# ['当然', '系列', '这么']
```

Misc

`faker.providers.misc`

```
faker.boolean(chance_of_getting_true=50)
# True
faker.md5(raw_output=False)
# '3166fa26ffd3f2a33e020dfe11191ac6'
faker.null_boolean()
# False
faker.password(length=10, special_chars=True, digits=True, upper_case=True,
lower_case=True)
# 'w7Ln8La@%0'
faker.sha1(raw_output=False)
# 'c8301a2a79445439ee5287f38053e4b3a05eac79'
faker.sha256(raw_output=False)
# '1e909d331e20cf241aaa2da894deae5a3a75e5cdc35c053422d9b8e7ccfa0402'
faker.uuid4(cast_to=<class 'str'>)
# '6e6fe387-6877-48d9-94ea-4263c4c71aa5'
```

Person

`faker.providers.person`

```
fake.first_name()
# '瑜'

fake.first_name_female()
# '娜'

fake.first_name_male()
# '勇'

fake.first_romanized_name()
# 'Jun'

fake.last_name()
# '宋'

fake.last_name_female()
# '王'

fake.last_name_male()
# '张'

fake.last_romanized_name()
# 'Long'

fake.name()
# '薛建国'

fake.name_female()
# '郑丹'

fake.name_male()
```

```
# '赖佳'

fake.prefix()
# ''

fake.prefix_female()
# ''

fake.prefix_male()
# ''

fake.romanized_name()
# 'Min Cai'

fake.suffix()
# ''

fake.suffix_female()
# ''

fake.suffix_male()
# ''
```

Profile

用于生成一份详细简历和简要简历

`faker.providers.profile`

```
fake.profile(fields=None, sex=None)
# {   'address': '安徽省银川县新城宁德路s座 326738',
#     'birthdate': datetime.date(1981, 9, 15),
#     'blood_group': 'O+',
#     'company': '盟新信息有限公司',
#     'current_location': (Decimal('7.284600'), Decimal('55.451655')),
#     'job': '培训讲师',
#     'mail': 'xiuying43@gmail.com',
#     'name': '项明',
#     'residence': '西藏自治区香港市和平乌鲁木齐路d座 374197',
#     'sex': 'F',
#     'ssn': '610523193505161301',
#     'username': 'tao27',
#     'website': [   'http://www.minglei.cn/',
#                     'https://xiulan.cn/',
#                     'https://taoli.cn/']}

fake.simple_profile(sex=None)
# {   'address': '湖北省兴安盟市南溪天津街w座 233313',
#     'birthdate': datetime.date(1925, 9, 8),
#     'mail': 'kqiu@gmail.com',
#     'name': '阎兵',
#     'sex': 'M',
#     'username': 'tanyan'}
```

Python

用于生成python各类型数据

`faker.providers.python`

```
fake.pybool()
# True

fake.pydecimal(left_digits=None, right_digits=None, positive=False,
min_value=None, max_value=None)
# Decimal('-841064605917.3')

fake.pydict(nb_elements=10, variable_nb_elements=True, *value_types)
# {   '原因': 4346,
#     '可能': 'nPuAKzCaIcdQpdFJuyAz',
#     '因为': Decimal('43081594989819.7'),
#     '成为': 8232,
#     '文化': 3581,
#     '管理': 'GNDUtFzQdpGLZmysKLZR'}
```

```
fake.pyfloat(left_digits=None, right_digits=None, positive=False,
min_value=None, max_value=None)
# 465871829.168

fake.pyint(min_value=0, max_value=9999, step=1)
# 1454

fake.pyiterable(nb_elements=10, variable_nb_elements=True, *value_types)
# {'nUWERQQFUqBonfiAimFd', -564399395230.72, 'YEvjqrHLJeBccztUVzSz',
'hbdarCrbshYfMTsAqnPw', 'caembxoMqALKnHMorpJE', 'igshGFTGETmOfIbyImEZ',
'znBxLwxIrrpwTinBoJAZ', 'wcui@yahoo.com', datetime.datetime(1994, 5, 16, 10, 6,
23), 'mIin@hotmail.com', Decimal('2069972.768153'), 'xwtHqKMDpGxCfBEFH0EV',
4315, 'juanhu@fangna.cn'}
```

```
fake.pylist(nb_elements=10, variable_nb_elements=True, *value_types)
# [   'YqsFQkICRRYSQXwDEx11',
#     'https://66.cn/app/list/wp-content/faq.jsp',
#     'hanxia@mingping.cn',
#     'https://jia.cn/main/app/about/',
#     3341,
#     'taofeng@hotmail.com',
#     'http://mingjie.com/',
#     664331168.6089,
#     'JbreOshZeENONfabfNOa']

fake.pyset(nb_elements=10, variable_nb_elements=True, *value_types)
# {-2917442783.9, 'QouhOiccwPJpJSDCWRWJ', 'JsfgoDcEdYBDmWUXcBcg',
'evYuJQBZVQsIZlEdBU', datetime.datetime(1997, 11, 17, 23, 38, 55),
-40919792339.99, 5208165284718.0, 'https://xiang.cn/about.jsp',
'GkXcjYnUacOEsWkFRKcQ', 'oegDvViAMUJAjntMVPeX', 533, 'FgkCEWkBwMGtpJDQntNW',
'EKElqqLKQktAcDiTtXFA'}
```

```
fake.pystr(min_chars=None, max_chars=20)
# 'GqWzPgqBfZvawNZWAQXu'

fake.pystruct(count=10, *value_types)
# (   [   'inbWOIOqPJvDCUiLiWtu',
```

```

# 'http://xr.cn/terms.html',
# datetime.datetime(1994, 9, 11, 4, 0, 46),
# datetime.datetime(1977, 6, 16, 3, 1, 41),
# Decimal('-16220.50097'),
# 'http://www.lugong.cn/category.html',
# 'juandai@hotmail.com',
# Decimal('-69628377501.36'),
# 'https://www.nali.cn/explore/index/',
# datetime.datetime(2018, 8, 15, 10, 47, 15)],
# { '北京': 'http://xiuyingyong.cn/wp-content/explore/explore/post/',
#   '就是': Decimal('15891497466771.0'),
#   '工程': 'zrkDKSjPFYusXTTFigfv',
#   '我们': 'jpsisvlgynlSwewkuzOF',
#   '控制': 'sGnGlmOhgnxRWKxqmGUP',
#   '最后': 'FkHSwFYxCKDWNKwbtswd',
#   '管理': 'ufCkWHdyGdYaUgTVTnOI',
#   '类型': 'exgo1PrCTDnuOmxbbWZA',
#   '能力': -77968229.9,
#   '论坛': datetime.datetime(1990, 8, 31, 16, 1, 36)},
# { '一点': { 1: Decimal('-661522809944.81'),
#             2: [3153, 9986, 7774],
#             3: { 1: datetime.datetime(1995, 3, 31, 14, 8, 11),
#                 2: 81686544894.1,
#                 3: [ 'PdbHbLJUHotoiqunsQEG',
#                     datetime.datetime(1979, 3, 16, 16, 21, 21)]}},
#   '企业': { 4: 'YLVrcNdmHetBIupLuVlh',
#             5: [ 'vKKPUILxvHRNdFovPdLv',
#                 -7290870725.3937,
#                 46779.9123533328],
#             6: { 4: 'https://www.fang.net/tags/wp-content/search.php',
#                 5: 'YGPvAKAWBrEohGHtycmh',
#                 6: [9117, 'pPdGCVrdprfQwvvootPI']}},
#   '你的': { 5: 'qCIQewXxZxeRkKbXgSUq',
#             6: [ 'tanming@guiyingxue.cn',
#                 'wkhIkrQUTvpPWJrPNzRZ',
#                 'uwihhgRLZpvTPaZiGJGQ'],
#             7: { 5: datetime.datetime(1994, 9, 5, 3, 38, 33),
#                 6: 'https://www.xuetan.net/index/',
#                 7: ['MNbBPNYSGHQcgvCydTgs',
#                     'cvaByZNITTvSwstfjKVH']}},
#   '信息': { 9: 'http://www.wuzhong.cn/home/',
#             10: [838, 6940, 'jBZHDNLKMxKbnxmgYshi'],
#             11: { 9: 'zhouxiulan@minjie.cn',
#                 10: 'fmeng@93.cn',
#                 11: [ 'xiexiuying@gmail.com',
#                     'jwwnLssYhvvIuVswDKQK']}},
#   '功能': { 0: 'http://lei.cn/tags/tags/main.php',
#             1: [8140, 'thfXfgXBupXvdrGOEqT', 6625],
#             2: { 0: 'gxue@xiayong.net',
#                 1: 6395,
#                 2: [ datetime.datetime(1998, 8, 23, 10, 50, 46),
#                     1677]}},
#   '单位': { 2: 9440,
#             3: [ 3136,
#                 -1125210.63288463,
#                 datetime.datetime(2018, 9, 25, 19, 37, 37)],
#             4: { 2: 'ztrAFpmrwgSVYAzPAkwv',
#                 3: datetime.datetime(1987, 5, 24, 12, 53, 50),

```

```

#         4: [ 'wei01@gmail.com',
#               datetime.datetime(2018, 7, 6, 3, 8, 46)]},
#
# '开发': {   3: datetime.datetime(2005, 2, 16, 6, 18, 21),
#             4: [-438531.19, 'ueBdqecQdaFtwcICNWOH', -1822721068802.0],
#             5: {   3: datetime.datetime(2017, 8, 11, 23, 52, 45),
#                   4: 'ifeng@wx.net',
#                   5: ['fgHxLVluztyDUNlApLUX', 6950]}},
#
# '知道': {   8: Decimal('255777920798845.0'),
#             9: [ 'IsqehAefSZcnKQqcOuEw',
#                   'https://xiulan.cn/app/wp-content/category/faq/',
#                   'eduEvDphZjewPfumKooY'],
#             10: {   8: 8070,
#                    9: 8182,
#                    10: [737450275326.4, -531799930141170.0]}},
#
# '行业': {   6: 237,
#             7: [ 'zliHEVSBNVwqKezrQNKO',
#                   'ozhjtEsLWDYLUKzPpkia',
#                   'kkmsUcmNuEcgLdmJTxd'],
#             8: {   6: 'nlu@qian.cn',
#                   7: 'maging@gong.cn',
#                   8: [7793, 'duuEchszCOQVTvmdQwiU']}},
#
# '语言': {   7: 'XDSOJwtdKdAvauWtQtRa',
#             8: [   3201,
#                   datetime.datetime(1975, 11, 29, 18, 6, 3),
#                   -2.83456],
#             9: {   7: 'dzwCLVOWKkQjjLcbihSd',
#                   8: 'ystKgAttzFljLMvWLhRn',
#                   9: [-185431619.375538, 5781342526604.19]}})

fake.pytuple(nb_elements=10, variable_nb_elements=True, *value_types)
# (   datetime.datetime(1984, 8, 30, 19, 13, 26),
#     8583,
#     'v\NHoffaesOBjxxiRpWz',
#     'pUXikEWbEhFJgEGjXdEp',
#     'http://www.chang.com/author/',
#     3288,
#     'MYstFTIFkENmUjDjdowr')

```

SSN

用于生成身份证号

```
faker.providers.ssn
```

```

fake.ssn(min_age=18, max_age=90)
# '23122219960103485X'

```

User agent

用于生成用户代理身份

```
faker.providers.user_agent
```

```
fake.android_platform_token()
```

```
# 'Android 2.3.2'

fake.chrome(version_from=13, version_to=63, build_from=800, build_to=899)
# ('Mozilla/5.0 (Macintosh; Intel Mac OS X 10_5_8) AppleWebKit/534.0 (KHTML, '
# 'like Gecko) Chrome/41.0.831.0 Safari/534.0')

fake.firefox()
# ('Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_3; rv:1.9.3.20) '
# 'Gecko/2013-04-29 16:06:30 Firefox/3.6.19')

fake.internet_explorer()
# 'Mozilla/5.0 (compatible; MSIE 9.0; windows NT 6.0; Trident/5.0)'

fake.ios_platform_token()
# 'iPad; CPU iPad OS 9_3_6 like Mac OS X'

fake.linux_platform_token()
# 'x11; Linux i686'

fake.linux_processor()
# 'i686'

fake.mac_platform_token()
# 'Macintosh; U; PPC Mac OS X 10_7_1'

fake.mac_processor()
# 'U; Intel'

fake.opera()
# 'Opera/8.63.(X11; Linux i686; ug-CN) Presto/2.9.165 Version/11.00'

fake.safari()
# ('Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_9 rv:3.0; pl-PL) '
# 'AppleWebKit/531.33.2 (KHTML, like Gecko) Version/4.0.3 Safari/531.33.2')

fake.user_agent()
# 'Mozilla/5.0 (compatible; MSIE 9.0; windows 98; Trident/5.1)'

fake.windows_platform_token()
# 'Windows NT 6.1'
```