

address.

```
zipCode(format?: string)
city(format?: number)
cityPrefix()
citySuffix()
streetName()
streetAddress(useFullAddress?: boolean)
streetSuffix()
streetPrefix()
secondaryAddress()
county()
country()
countryCode()
state(useAbbr?: boolean)
stateAbbr()
latitude()
longitude()
```

commerce.

```
color()
department()
productName()
price(
  min?: number,
  max?: number,
  dec?: number,
  symbol?: string)
productAdjective()
productMaterial()
product()
```

company.

```
suffixes(): string[];
companyName(format?: number)
companySuffix()
catchPhrase()
bs()
catchPhraseAdjective()
catchPhraseDescriptor()
catchPhraseNoun()
bsAdjective()
bsBuzz()
bsNoun()
```

database.

```
column()
type()
collation()
engine()
```

date. (return Date)

```
past(years?: number, refDate?: string | Date)
future(years?: number, refDate?: string | Date)
between(
  from: string | number | Date,
  to: string | Date )
recent(days?: number)
month(
  options?: { abbr?: boolean, context?: boolean } )
weekday(
  options?: { abbr?: boolean, context?: boolean } )
```

hacker.

```
abbreviation()
adjective()
noun()
verb()
ingverb()
phrase()
```

phone.

```
phoneNumber(format?: string)
phoneNumberFormat(
  phoneFormatsArrayIndex?: number )
phoneFormats()
```

finance.

```
account(length?: number)
accountName()
mask(
  length?: number,
  parens?: boolean,
  elipsis?: boolean )
amount(
  min?: number,
  max?: number,
  dec?: number,
  symbol?: string )
transactionType()
currencyCode()
currencyName()
currencySymbol()
bitcoinAddress()
iban(formatted?: boolean): string
bic(): string
```

image.

```
image()
avatar()
imageUrl(
  width?: number,
  height?: number,
  category?: string,
  randomize?: boolean,
  https?: boolean )
abstract(width?: number, height?: number)
animals(width?: number, height?: number)
business(width?: number, height?: number)
cats(width?: number, height?: number)
city(width?: number, height?: number)
food(width?: number, height?: number)
nightlife(width?: number, height?: number)
fashion(width?: number, height?: number)
people(width?: number, height?: number)
nature(width?: number, height?: number)
sports(width?: number, height?: number)
technics(width?: number, height?: number)
transport(width?: number, height?: number)
dataUri(width?: number, height?: number)
```

system.

```
fileName(ext?: string, type?: string)
commonFileName(ext: string, type?: string)
mimeType()
commonFileType()
commonFileExt()
fileType()
fileExt(mimeType: string)
directoryPath()
filePath()
semver()
```

helpers.

```
randomize<T>(array: T[]): T;
randomize()
slugify(string?: string)
replaceSymbolWithNumber(
  string?: string,
  symbol?: string )
replaceSymbols(string?: string)
shuffle<T>(o: T[]): T[];
shuffle(): string[];
mustache(
  str: string,
  data: {
    [key: string]: string | (
      (substring: string,
       ...args: any[]) => string
    )
  })
createCard(): Card;
contextualCard(): ContextualCard;
userCard(): UserCard;
createTransaction(): Transaction;
```

internet.

```
avatar()
email(
  firstName?: string,
  lastName?: string,
  provider?: string )
exampleEmail(firstName?: string, lastName?: string)
userName(firstName?: string, lastName?: string)
protocol()
url()
domainName()
domainSuffix()
domainWord()
ip()
ipv6()
userAgent()
color(
  baseRed255?: number,
  baseGreen255?: number,
  baseBlue255?: number )
mac()
password(
  len?: number,
  memorable?: boolean,
  pattern?: string | RegExp,
  prefix?: string )
```

lorem.

```
word()
words(num?: number)
sentence(wordCount?: number, range?: number)
slug(wordCount?: number)
sentences(sentenceCount?: number)
paragraph(sentenceCount?: number)
paragraphs(paragraphCount?: number, separator?: string)
text(times?: number)
lines(lineCount?: number)
```

name.

```
firstName(gender?: number)
lastName(gender?: number)
findName(
  firstName?: string,
  lastName?: string,
  gender?: number )
jobTitle()
prefix()
suffix()
title()
jobDescriptor()
jobArea()
jobType()
```

random.

```
number(max?: number): number;
number(
  options?: {
    min?: number,
    max?: number,
    precision?: number
  }): number;
arrayElement()
arrayElement<T>(array: T[]): T;
arrayElement<T>(array: ReadonlyArray<T>): T;
objectElement(object?: { [key: string]: any }, field?: "key")
objectElement<T>(object?: { [key: string]: T }, field?: any): T;
uuid()
boolean(): boolean;
word(type?: string)
words(count?: number)
image()
locale()
alphaNumeric(count?: number)
```

fake(str: string): string;