python-party-pack Documentation

Release 1.0.0

Daniel Greenfeld

Sep 27, 2017

Contents

1 Reference: Party 3

2 Indices and tables 5

Python Module Index 7

Combining a basic yet fun Python project with Sphinx Docs, Coverage.py, and some other tools, plus some notes on good practices. This is a jump start to get involved with the party fun of Document Driven Development (DDD) and Test Driven Development (TDD).

Contents:

CHAPTER 1

Reference: Party

exception party.**DontHaveThatDrink**
This is thrown when someone asks for a drink not supplied.

exception party.**EveryoneHasPartyHat**
This is thrown when everyone has managed to get their hat on.

exception party.**OutOfThatDrink**
This is thrown when someone asks for a drink whose supply has been exhausted.

class party.**Party**(*attendees=None*, ***kwargs*)
A programmatic representation of a party. Accepts the following parameters:

- attendees (numeric)
- beers (numeric)
- sangrias (numeric)
- wines (numeric)
- lemonades (numeric)

Note: If you don't supply a value for any of the parameters, the Party will act like all parties do and these parameter will show up in random quantities.

---

drink(*substance_name*)
Drink an available substances specified in VALID_DRINKS constant

inventory()
Display an inventory of guests and available drinks

put_hat_on_attendee()
Put a hat on an attendee

party.**random_party**()
Use this function to hold a completely party.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

party, 3

Index

D

DontHaveThatDrink, 3
drink() (party.Party method), 3

E

EveryoneHasPartyHat, 3

I

inventory() (party.Party method), 3

O

OutOfThatDrink, 3

P

Party (class in party), 3
party (module), 3
put_hat_on_attendee() (party.Party method), 3

R

random_party() (in module party), 3