
Inventory Documentation

Release 0.0.0

Luke Sloop

Dec 06, 2017

Class Reference

1 Inventory API

3

inventory.py is a simple game inventory save system written in python. The inventory allows you to control what describes the items and store them. It also handles saving and loading inventories. I created this for practice in writing python and in practice in documenting with [readthedocs](#) and [sphinx](#)

1.1 Modules

1.1.1 inventory.py

Item

Inherits: object

Module: *inventory*

Brief Description

Represents an inventory item instance for storing and retrieving item properties.

Instance Methods

<i>Item</i>	<i>Item</i> (dict kwargs)
None	<i>set</i> (str keyword, str arg)
str	<i>get</i> (str keyword)

Description

Represents an item that can be stored in an inventory. The item takes in as many named arguments as necessary that describe it.

Instance Variables

- **dict kwargs** - A list of keywords describing the item

Instance Method Descriptions

- **Item Item** (dict kwargs)

Construct a new inventory item with as many keyword arguments describing the item as needed.

- **None set** (str keyword, str arg)

Set the value for the keyword argument in *kwargs*.

- **str get** (str keyword)

Return the argument for the dictionary key keyword in kwargs. Returns None if the keyword is not in kwargs

Supported Magic Methods

- **str __str__** ()

Returns *kwargs* as a string.

Inventory

Inherits: object

Module: *inventory*

Brief Description

Represents a basic inventory for item management

Instance Methods

<i>Inventory</i>	<i>Inventory</i> (int size)
None	<i>load</i> (str path)
None	<i>save</i> (str path)
None	<i>sort</i> (str keyword=None)
None	<i>add</i> (<i>Item</i> item)
None	<i>set</i> (int slot, <i>Item</i> item)
None	<i>set_size</i> (int size)
int	<i>find</i> (<i>Item</i> item)
int	<i>find_first_item</i> ()
int	<i>find_first_empty</i> ()
<i>Item</i>	<i>get</i> (int slot)
int	<i>get_empty_count</i> ()
int	<i>get_items_count</i> ()
List	<i>get_items</i> ()
List	<i>get_items_with_kwargs</i> (str keyword, str args=None)
int	<i>get_size</i> ()
bool	<i>is_full</i> ()
bool	<i>is_empty</i> ()
<i>Item</i>	<i>remove</i> (<i>Item</i> item)
List	<i>remove_all</i> ()
<i>Item</i>	<i>remove_from</i> (int slot)

Instance Method Descriptions

- **Inventory** *Inventory* (int size)

Construct a new inventory of the passed in *size*.

- **None** *load* (str path)

Loads an inventory from the passed in filepath.

- **None** *save* (str path)

Saves the current inventory to the passed in filepath

- **None** *sort* (str keyword=None)

Moves the items with the *keyword* to the start of the inventory sorting them by *keyword* in either alphabetical order or numerical order (If only contains numbers). If *keyword* is not passed in then it just moves all of the items to the start of the inventory.

- **None** *add* (*Item* item)

Add the passed in item instance to the inventory at the next empty slot. If no slot is empty then it will raise an *OverflowError*. Use *is_full* before adding an item without slot specified to ensure that no error is thrown.

- **None** *set* (int slot, *Item* item)

Add the *item* to the specific inventory slot, The new *item* will replace the current *item* if one exists.

- **None** *set_size* (int size)

Sets the number of items that the inventory can hold. This will cut off any items that are past the current size, or add empty slots to extend the size.

- *Item* *get* (int slot)

Return the item instance at the inventory `slot` index.

- **int** `get_empty_count()`

Returns the number of empty slots in the inventory.

- **int** `get_items_count( Item item )`

Returns the number of items if `item` is `None`. If `item` is specified then returns the number of instances of that item in the inventory.

- **List** `get_items()`

Returns a *List* of items in the inventory. Alternative to `Item().items_list` that will also return empty *items*.

- **List** `get_items_with_kwargs( str keyword, str arg=None )`

Returns a list of inventory items with the specific `keyword` argument. If `arg` is not `None` then it will return only the items that have the keyword argument and the arguments value is `args`

- **List** `get_size()`

Returns the the amount of items the inventory can hold

- **int** `find( Item item )`

Returns the first slot index of the `item` if it exists in the inventory. If no *Item* is found returns `-1`.

- **int** `find_first_item()`

Returns the slot index of the first item in the inventory if the inventory is not empty. If the inventory is empty returns `-1`.

- **int** `find_first_empty()`

Returns the slot index of the first empty slot in the inventory. If the inventory is full returns `-1`.

- **bool** `is_full()`

Returns `True` if the inventory has no empty slots. Otherwise returns `False`.

- **bool** `is_empty()`

Returns `True` if the inventory does not contain any items. If the inventory contains one or more items, returns `False`

- *Item* `remove( Item item )`

Remove and return the first instance of the `item`. If the item is not found it will raise a *SlotEmptyError*.

- **List** `remove_all()`

Removes all *Items* and returns them as a *List*.

- *Item* `remove_from( int slot )`

Remove and return the `item` from the inventory slot index. If the slot index is empty it raises a *SlotEmptyError*.

Supported Magic Methods

- **None** `__iter__()`

Iterates through the inventory items.

- **bool** `__contains__()`

Returns true if the item is in the inventory.

- **Item** `__getitem__()`

Calls *get* on the item.

- **int** `__len__()`

Returns `items_count`

- **Inventory** `__add__()`

Returns this inventory with the second inventory concatenated to the end.

- **bool** `__eq__()`

Returns `True` if this inventory has the same items and length as other inventory. Otherwise returns `False`.

- **bool** `__lt__()`

Returns `True` if this inventory has less items than the other inventory. Otherwise returns `False`.

- **bool** `__le__()`

Returns `True` if this inventory has less or equal number of items than the other inventory. Otherwise returns `False`

- **str** `__str__()`

Returns this inventory as a `string`.

SlotEmptyError

Inherits: exception

Module: *inventory*

Brief Description

Raised on an attempt to remove an item from an empty inventory slot