**help('dict')**

Help on class dict in module builtins:

class dict(object)

dict() -> new empty dictionary

dict(mapping) -> new dictionary initialized from a mapping object's

(key, value) pairs

dict(iterable) -> new dictionary initialized as if via:

d = {}

for k, v in iterable:

d[k] = v

dict(\*\*kwargs) -> new dictionary initialized with the name=value pairs

in the keyword argument list. For example: dict(one=1, two=2)

Methods defined here:

\_\_contains\_\_(self, key, /)

True if the dictionary has the specified key, else False.

\_\_delitem\_\_(self, key, /)

Delete self[key].

\_\_eq\_\_(self, value, /)

Return self==value.

\_\_ge\_\_(self, value, /)

Return self>=value.

\_\_getattribute\_\_(self, name, /)

Return getattr(self, name).

\_\_getitem\_\_(...)

x.\_\_getitem\_\_(y) <==> x[y]

\_\_gt\_\_(self, value, /)

Return self>value.

\_\_init\_\_(self, /, \*args, \*\*kwargs)

Initialize self. See help(type(self)) for accurate signature.

\_\_ior\_\_(self, value, /)

Return self =value.

\_\_iter\_\_(self, /)

Implement iter(self).

\_\_le\_\_(self, value, /)

Return self<=value.

\_\_len\_\_(self, /)

Return len(self).

\_\_lt\_\_(self, value, /)

Return self<value.

\_\_ne\_\_(self, value, /)

Return self!=value.

\_\_or\_\_(self, value, /)

Return self value.

\_\_repr\_\_(self, /)

Return repr(self).

\_\_reversed\_\_(self, /)

Return a reverse iterator over the dict keys.

\_\_ror\_\_(self, value, /)

Return value self.

\_\_setitem\_\_(self, key, value, /)

Set self[key] to value.

\_\_sizeof\_\_(...)

D.\_\_sizeof\_\_() -> size of D in memory, in bytes

clear(...)

D.clear() -> None. Remove all items from D.

copy(...)

D.copy() -> a shallow copy of D

get(self, key, default=None, /)

Return the value for key if key is in the dictionary, else default.

items(...)

D.items() -> a set-like object providing a view on D's items

keys(...)

D.keys() -> a set-like object providing a view on D's keys

pop(...)

D.pop(k[,d]) -> v, remove specified key and return the corresponding value.

If the key is not found, return the default if given; otherwise,

raise a KeyError.

popitem(self, /)

Remove and return a (key, value) pair as a 2-tuple.

Pairs are returned in LIFO (last-in, first-out) order.

Raises KeyError if the dict is empty.

setdefault(self, key, default=None, /)

Insert key with a value of default if key is not in the dictionary.

Return the value for key if key is in the dictionary, else default.

update(...)

D.update([E, ]\*\*F) -> None. Update D from dict/iterable E and F.

If E is present and has a .keys() method, then does: for k in E: D[k] = E[k]

If E is present and lacks a .keys() method, then does: for k, v in E: D[k] = v

In either case, this is followed by: for k in F: D[k] = F[k]

values(...)

D.values() -> an object providing a view on D's values

----------------------------------------------------------------------

Class methods defined here:

\_\_class\_getitem\_\_(...) from builtins.type

See PEP 585

fromkeys(iterable, value=None, /) from builtins.type

Create a new dictionary with keys from iterable and values set to value.

----------------------------------------------------------------------

Static methods defined here:

\_\_new\_\_(\*args, \*\*kwargs) from builtins.type

Create and return a new object. See help(type) for accurate signature.

----------------------------------------------------------------------

Data and other attributes defined here:

\_\_hash\_\_ = None