

This documentation was generated from the Python documentation available by typing `help(dict)` in the Python shell. In the documentation found here the variable *D* is a reference to a dictionary. A few methods were omitted here for brevity. The official Python 3 documentation is at <http://docs.python.org/3/>.

Method	Returns	Comments
<code>dict()</code>	dict	new empty dictionary
<code>dict(mapping)</code>	dict	new dictionary initialized from a mapping object's (key, value) pairs
<code>dict(seq)</code>	dict	new dictionary initialized as if via: D = {} for k, v in seq: D[k] = v
<code>dict(**kwargs)</code>	dict	new dictionary initialized with the name=value pairs in the keyword arg list. For example: <code>dict(one=1, two=2)</code>
<code>k in D</code>	bool	True if D has key k, else False
<code>del D[k]</code>		Deletes key k from dictionary D
<code>D1==D2</code>	bool	Returns True if dictionaries D1 and D2 have same keys mapped to same values
<code>D[k]</code>	value type	Returns value k maps to in D. If k is not mapped, it raises a <code>KeyError</code> exception
<code>iter(D)</code>	iterator	Returns an iterator over D
<code>len(D)</code>	int	Returns the number of keys in D
<code>D1!=D2</code>	bool	Returns True if D1 and D2 have any different keys or keys map to different values
<code>repr(D)</code>	str	Returns a string representation of D
<code>D[k]=e</code>		Stores the key, value pair k,e in D
<code>D.clear()</code>	None	Remove all items from D
<code>D.copy()</code>	dict	a shallow copy of D
<code>D.get(k[,e])</code>	value type	D[k] if k in D, else e. e defaults to None
<code>D.items()</code>	items	a set-like object providing a view on D's items

(continued)

(continued)

Method	Returns	Comments
D.keys()	keys	a set-like object providing a view on D's keys
D.pop(k[,e])	v	remove specified key and return the corresponding value. If key is not found, e is returned if given, otherwise <code>KeyError</code> is raised
D.popitem()	(k, v)	remove and return some (key, value) pair as a 2-tuple; but raise <code>KeyError</code> if D is empty
D.setdefault(k[,e])	D.get(k,e)	Returns D.get(k,e) and also sets d[k]=e if k not in D
D.update(E, **F)	None	Update D from dict/iterable E and F If E has a .keys() method, does: for k in E: D[k] = E[k] If E lacks .keys() method, does: for (k, v) in E: D[k] = v In either case, this is followed by: for k in F: D[k] = F[k]
D.values()	values	an object providing a view on D's values