Dictionaries

Quick Reference:

```
empty_dict = { } # Create an empty dictionary.
bb_alumni_victory = {
        2009:"Will Rice",
        2008:"Brown",
        2006:"Will Rice",
        1992:"Hanszen",
        1982:"Wiess" } # Create an explicit dictionary.
bb_alumni_victory[1984] = "Hanszen" # Add or update a key.
>>> bb_alumni_victory[2008]
'Brown'
```

Exercise 1 - Dictionary Creation

Write a function

```
def make_dict(keys, default_value):
    """
    Creates a new dictionary with the specified keys and default value.

    Arguments:
    keys          -- A list of keys to be included in the returned dictionary.
    default_value -- The initial mapping value for each key.

    Returns:
    A dictionary where every key is mapped to the given default_value.
    """
    ...
```

that takes a list of keys and a default value and creates a dictionary containing these keys with each key mapping to the given default value.

Assert your function produces the following output:

```
>>> make_dict(["a","b","c"], "z")
{'a': 'z', 'c': 'z', 'b': 'z'}
```

```
>>> make_dict([], [])
{}
```

Exercise 2 - Value Assertion

Write a function

```
def ensure_key_value_pair(pairs, key, expected_value):
    """
    Checks to ensure that the mapping of key in pairs matches the given expected value
.

    If the state of pairs is such that the given key is already mapped to the given ex
pected value
    this function in no way modifies the dictionary and returns the given dictionary.

    If the state of pairs is such that the given key does not map to the given expecte
d value
    (or no such key is contained in pairs) then update (or add) the mapping of key to
    the given expected value and return the given dictionary.

    Arguments:
    pairs          -- A dictionary to check for the expected mapping.
    key            -- The key of the expected mapping.
    expected_value -- The the value of the expected mapping.

    Returns:
    The given dictionary.
    """
    ...
```

that returns the given dictionary if the lookup value of the dictionary under the given `key` has the same value as `expected_value`. Otherwise, ensure that the value at `key` for the given dictionary is `expected_value` and return the updated dictionary.

Note: You may find the `dict` method `has_key(key)` which tests whether the given key is contained in the dictionary useful.

Assert your function produces the following output:

```
>>> pairs = { 1:"a", 2:"b" }
>>> ensure_key_value_pair(pairs, 1, "a")
{1: 'a', 2: 'b'}
```

```
>>> ensure_key_value_pair(pairs, 2, "z")
{1: 'a', 2: 'z'}
```

```
>>> ensure_key_value_pair(pairs, 3, "x")
{1: 'a', 2: 'z', 3: 'x'}
```

More information

Consult the Python documentation for more details about dictionaries.

Created Sat 23 Aug 20149:17 PM BST

Last Modified Sun 24 Aug 2014 5:42 AM BST