



JÖNKÖPING UNIVERSITY

School of Engineering

DICTS IN PYTHON

Peter Larsson-Green

Jönköping University

Autumn 2018

DICTS (DICTIONARIES)

Used to store multiple values.

- Each value is associated with a key.
- Expressions creating dicts:

```
{ }
```

```
{ <key-expr> : <val-expr> }
```

```
{ <key-expr> : <val-expr> , <key-expr> : <val-expr> }
```

DICTS (DICTIONARIES)

Used to store multiple values.

- Expression retrieving an item from a dict:

```
<dict-expr> [ <key-expr> ]
```

- Statement adding an item to a dict (or update an existing):

```
<dict-expr> [ <key-expr> ] = <val-expr>
```

EXAMPLE

```
my_dict = {'one': 1, 'two': 2}
```

```
my_dict['three'] = 3
```

```
four = my_dict['one'] + my_dict['three']
```

```
w = {0:0}[0]
```

```
z = {0:{0:0}}[0][0]
```

```
two = {0:{1:2}}[0][1]
```

<dict-expr> [<key-expr>]



{0:0}

[0]



{ 0 : 0 }

EXAMPLE

```
def get_digit_as_string(digit):  
    strings = ["zero", "one", "two", "three", ...]  
    return strings[digit]
```

`get_digit_as_string(1)` → "one"

`get_digit_as_string(3)` → "three"

```
def get_string_as_digit(string):  
    digits = {"zero": 0, "one": 1, "two": 2, "three": 3}  
    return digits[string]
```

`get_string_as_digit("one")` → 1

`get_string_as_digit("three")` → 3

DICTS (DICTIONARIES)

- The key must be "constant"/"immutable"/"hashable".
 - Numbers and strings work.
 - Lists and dicts do not work.
- Expression checking if a key exists in dict:
 - `<key-expr> in <dict-expr>`
 - `'a' in {'a': 1} → True`
 - `1 in {'a': 1} → False`
- Expression checking if a key doesn't exist in dict:
 - `<key-expr> not in <dict-expr>`
 - `'a' not in {'a': 1} → False`
 - `1 not in {'a': 1} → True`

EXAMPLE

```
def number_of_occurrences(the_list):  
    occurrences = {}  
    for e in the_list:  
        if e not in occurrences:  
            occurrences[e] = 1  
        else:  
            occurrences[e] += 1  
    return occurrences
```

```
def number_of_occurrences(the_list):  
    occurrences = {}  
    for e in the_list:  
        occurrences[e] = 0  
    for e in the_list:  
        occurrences[e] += 1  
    return occurrences
```

```
number_of_occurrences([1, 4, 2, 2, 1]) → {1: 2, 2: 2, 4: 1}
```


MORE ABOUT DICTS

- Remove item with a specific key:
 - `the_dict.pop(<key-expr>)`
- Iterate over the keys:

```
for key in the_dictionary:  
    value = the_dictionary[key]
```

MORE ABOUT DICTS

- Remove all entries:
 - `the_dict.clear()`
- Set a value for a key if it doesn't have one:
 - `the_dict.setdefault(<key-expr>, <value-expr>)`

```
if the_key not in the_dict:  
    the_dict[the_key] = <value-expr>
```

- Documentation:
 - `help({})`