# Dictionaries

- `Dict` is generalization of lists, but now indices don't have to be integers – can be values of any immutable type
- Refer to indices as **keys**, since arbitrary in form
- A `dict` is then a collection of `<key, value>` pairs
- Syntax
  - `monthNumbers = { 'Jan':1, 'Feb':2, 'Mar':3, 1:'Jan', 2:'Feb', 3:'Mar'}`

# We access by using a key

```
monthNumbers =
    {'Jan':1, 'Feb':2,
     'Mar':3, 1:'Jan',
     2:'Feb', 3:'Mar'}
```

```
monthNumbers['Jan']
```

returns

```
1
```

```
monthNumbers[1]
```

returns

```
'Jan'
```

Entries in a dict are unordered, and can only be accessed by a key, not an index

# Operations on dicts

- Insertion

```
monthNumbers['Apr'] = 4
```

- Iteration

```
collect = []

for e in monthNumbers:
  collect.append(e)
```

collect is now

```
[1, 2, 'Mar', 'Feb', 'Apr', 'Jan', 3]
```

Compare to

```
monthNumbers.keys()
```

# Keys can be complex

```
myDict = {(1,2): 'twelve',
          (1,3): 'thirteen'}

myDict[(1,2)]
```

returns

‘twelve’

Note that keys must be immutable, so have to use a tuple, not a list

We will return to dicts and their methods later