

Tuples (“immutable lists”)

```
>>> foo = (42, 'hello', (5, 'spam'), 'penguin')
>>> foo
(42, 'hello', (5, 'spam'), 'penguin')
>>> foo[0]
42
>>> foo[-1]
'penguin'
>>> foo[0:2]
(42, 'hello')
>>> foo[0:1]
(42,)
```

Tuples (“immutable lists”)

```
>>> foo = (42, 'hello', (5, 'spam'), 'penguin')
```

```
>>> foo
```

```
(42, 'hello', (5, 'spam'), 'penguin')
```

```
>>> foo[0]
```

```
42
```

```
>>> foo[-1]
```

```
'penguin'
```

```
>>> foo[0:2]
```

```
(42, 'hello')
```

```
>>> foo[0:1]
```

```
(42,)
```

```
>>> foo[0] = 100
```

```
BARF!!! (that's California-speak for 'error')
```

Dictionaries

```
>>> D = {}
>>> D["Ran"] = "spam"
>>> D["Zach"] = "donuts"
>>> D["Alien"] = 42
>>> D["Ran"]
'spam'
>>> D["Alien"]
42
>>> D["Napoleon Dynamite"]
BARE!
```

“Ran”, “Zach”, and “Alien” are called the “keys” in the dictionary. Any *immutable* object can be a key.

Dictionaries

```
>>> D = {}
>>> D["Ran"] = "spam"
>>> D["Zach"] = "donuts"
>>> D["Alien"] = 42
>>> D["Ran"]
'spam'
>>> D["Alien"]
42
>>> D["Napoleon Dynamite"]
BARF!
>>> D
{ 'Ran': 'spam', 'Zach': 'donuts', 'Alien': 42 }
```

Dictionaries - summary

A dictionary associates values with keys.

```
D = {}      # create an empty dictionary
D[k]= v     # make key k have value v
            # (replace old value if k already in D)
D[k]        # get value under key k
k in D      # whether k is a key in D
```

Example: { 'cat':3, 'avatar':1, 'sprite':42 }

v can be any value

k must be an immutable type (string, int,
tuple of immutables)