Accessing Elements - dict

Let us see how we can access elements from the `dict` using Python as programming language.

- We can access a value of a particular element in `dict` by passing key `d[key]`. If the key does not exists, it will throw **KeyError**.
- `get` also can be used to access a value of particular element in `dict` by passing key as argument. However, if key does not exists, it will return None.
- We can also pass a default value to `get`.
- We can get all the keys in the form of set like object by using `keys` and all the values in the form of list like object by using `values`.
- We can also use `items` to convert a `dict` into a set like object with pairs. Each element (which is a pair) in the set like object will be a tuple.
- Let us see few examples.

```
d = {'id': 1, 'first_name': 'Scott', 'last_name': 'Tiger', 'amount': 1000.0}
```

```
d['id']
```

```
1
```

```
d['first_name']
```

```
'Scott'
```

```
d['commission_pct'] # throws key error
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-5-583ad3d358bd> in <module>
----> 1 d['commission_pct'] # throws key error

KeyError: 'commission_pct'
```

```
d.get?
```

```
Docstring: D.get(k[,d]) -> D[k] if k in D, else d.  d defaults to None.
Type:      builtin_function_or_method
```

```
d.get('first_name')
```

```
'Scott'
```

```
d.get('commission_pct') # Returns None
```

```
d.get('first_name', 'Some First Name')
```

```
'Scott'
```

```
d.get('commission_pct', 0)
```

```
0
```

```
d.keys?
```

```
Docstring: D.keys() -> a set-like object providing a view on D's keys
Type:      builtin_function_or_method
```

```
d.keys()
```

```
dict_keys(['id', 'first_name', 'last_name', 'amount'])
```

```
d.values?
```

```
Docstring: D.values() -> an object providing a view on D's values
Type:      builtin_function_or_method
```

```
d.values()
```

```
dict_values([1, 'Scott', 'Tiger', 1000.0])
```

```
d.items?
```

```
Docstring: D.items() -> a set-like object providing a view on D's items
Type:      builtin_function_or_method
```

```
d.items()
```

```
dict_items([('id', 1), ('first_name', 'Scott'), ('last_name', 'Tiger'), ('amount', 1000.0)])
```

```
list(d.items())[0]
```

```
('id', 1)
```

```
list(d.items())[1]
```

```
('first_name', 'Scott')
```

```
type(list(d.items())[1])
```

```
tuple
```