Overview of list and set

There are 4 types of collections in Python. While `list` and `set` fundamentally contain homogeneous elements, `dict` and `tuple` contain heterogeneous elements.

- Homogeneous means of same type.
- Examples of collections with homogeneous elements.
  - Collection of employees - `list`
  - Collection of unique employees - `set`
  - Collection of integers - `list`
  - Collection of unique integers - `set`
- Based up on the requirement we should use appropriate type of collection.
- `list`
  - Group of homogenous elements.
  - There can be duplicates in the `list`.
  - `list` can be created by enclosing elements in `[]` - example `[1, 2, 3, 4]`.
  - Empty `list` can be initialized using `[]` or `list()`.
- `set`
  - Group of homogenous elements
  - No duplicates allowed in the `set`. Even if you add same element more than once, such elements will be ignored.
  - `set` can be created by enclosing elements in `{}` - example `{1, 2, 3, 4}`.
  - Empty `set` can be initialized using `set()`. We cannot initialize empty set using `{}` as it will be treated as empty `dict`.
- `list` and `set` can be analogous to Table with columns and rows while `dict` and `tuple` can be analogous to a row with in a table.
- `list` can hold duplicate values while `set` can only hold unique values.
- If you want to have a row with column names then we use `dict` otherwise we use `tuple`.
- We will deep dive into all types of collections to get better understanding about them.

```
l = [1, 2, 3, 3, 4, 4]
```

```
l
```

```
[1, 2, 3, 3, 4, 4]
```

```
l = []
```

```
l
```

```
[]
```

```
type(l)
```

```
list
```

```
l = list()
```

```
l
```

```
[]
```

```
s = {1, 2, 3, 3, 4, 4}
```

```
s
```

```
{1, 2, 3, 4}
```

```
type(s)
```

```
set
```

```
s = set() # Initializing empty set
```

```
s
```

```
set()
```

```
s = {} # s will be of type dict
```

```
type(s)
```

```
dict
```