

Python Mapping for Structures

A Slice [structure](#) maps to a Python class with the same name. For each Slice data member, the Python class contains a corresponding attribute. For example, here is our [Employee](#) structure once more:

Slice

```
struct Employee {
    long number;
    string firstName;
    string lastName;
};
```

The Python mapping generates the following definition for this structure:

Python

```
class Employee(object):
    def __init__(self, number=0, firstName='', lastName=''):
        self.number = number
        self.firstName = firstName
        self.lastName = lastName

    def __eq__(self, other):
        # ...

    def __ne__(self, other):
        # ...

    def __str__(self):
        # ...

    def __hash__(self):
        # ...

    # ...
```

The constructor initializes each of the attributes to a default value appropriate for its type. You can also declare different [default values](#) for members of primitive and enumerated types.

The `__eq__` method returns true if all members of two structures are (recursively) equal, and `__ne__` returns true if any member differs.

The `__str__` method returns a string representation of the structure.

For structures that are also [legal dictionary key types](#), the mapping also generates relational operators (`__lt__`, `__le__`, `__gt__`, `__ge__`) and a `__hash__` method. The `__hash__` method returns a hash value for the structure based on the value of all its data members.

See Also

- [Structures](#)
- [Dictionaries](#)
- [Python Mapping for Identifiers](#)
- [Python Mapping for Modules](#)
- [Python Mapping for Built-In Types](#)
- [Python Mapping for Enumerations](#)
- [Python Mapping for Sequences](#)
- [Python Mapping for Dictionaries](#)
- [Python Mapping for Constants](#)
- [Python Mapping for Exceptions](#)