Tuples

Declaration :

```
> Tup = (4, 5)        , Not mutable, Ordred, indexed
> Tup = 4, 5
> ma-Tup = (4, )        # Tuple d'un element
```

Access :

```
> Tup [0]        # index, Tuple ordred indexed
        4
```
→ Pour modifier Tuple reflechir à creer une autre Tuple et le concatener ; par des slices
```
> a = (3, ) + a[1:2]        > a = b + c
```

Tuple destruct :

unpack Tuple.
```
> a = ("A", "B", "C")

> x, y, z = a        # x = "A", y = "B", z = "C"

> a = ("A", "B", 4, "C")
> x, y, _, 3 = a        # x = "A", y = "B", 3 = "c"
                ignore element index 2
```

Update Tuple.

```
> x = ("apple", "banana", "cherry")
> y = list (x)
> y [1] = "kiwi"
> x = tuple (y)
```

Tuple

Méthodes Tuples

count()

> t. count (value)
— o return the numbers of times a specified value.

Value : required. The item to search for.

index()

> t. index (value)
— o finds the first occurence of the specified value
— o raises an exception if the value i not found

Value : required, The item to search for.