

zip 関数による結合

次に学習するのは、複数のリストやタプルにまたがって格納されたデータを、一つのオブジェクトにまとめる方法です。

具体的には、同一インデックスをもつ要素を取り出して作られたタプルを要素とする **zip 型**の **zip オブジェクト**を生成します。この作業を“**zip 化する**”といい、それを行うのが、組込み関数である **zip 関数**です。

… という言葉での解説を理解するのは、かなり大変です。以下に示す例と、そのイメージ図である **Fig.8-6**を対比して、理解していきましょう。

例 8-4 zip関数によるリストの結合

```

>>> p1 = [75, 56, 89]
>>> p2 = [42, 85, 77]
>>> name = ['渡辺', '西田', '杉田']
>>> pl = list(zip(p1, p2))
>>> pl
[(75, 42), (56, 85), (89, 77)]
>>> pt = tuple(zip(p1, p2, name))
>>> pt
((75, 42, '渡辺'), (56, 85, '西田'), (89, 77, '杉田'))

```

最初に生成されている **p1**、**p2**、**name** は、それぞれ、3人の英語の点数、数学の点数、名前を要素としてもつ、要素数3のリストです。

a リストをzip化してリストを生成

b リストをzip化してタプルを生成

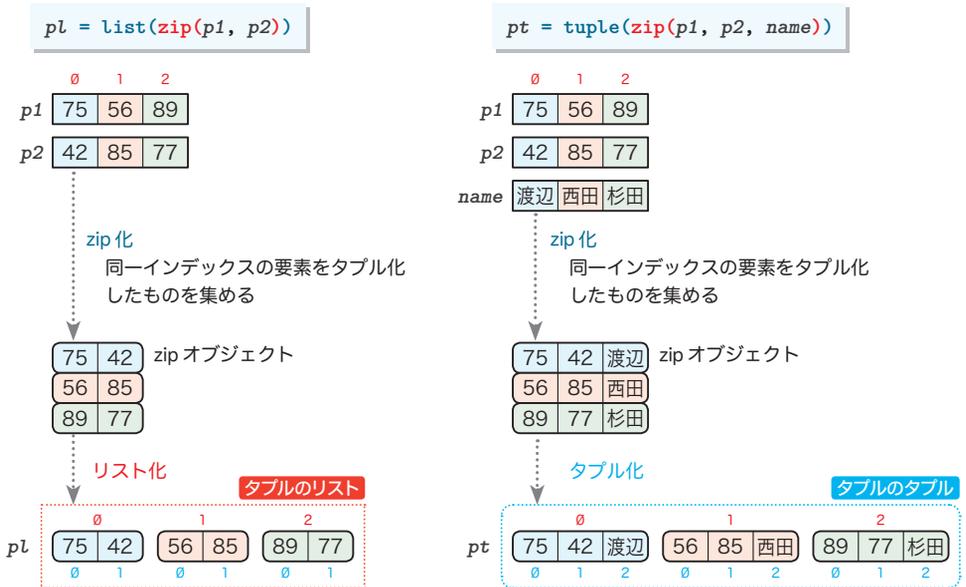


Fig.8-6 zip 関数によるリストの結合

図8aは、`p1`と`p2`をzip化した（英語と数学の点数のタプルをまとめた）リストを生成する例です。そして、図8bは、`p1`と`p2`と`name`をzip化した（英語と数学と名前のタプルをまとめた）タプルを生成する例です。

- ▶ 生成後のリストあるいはタプルの要素数は、いずれも3です。

*

いずれの例でも、インデックスが0の要素、1の要素、2の要素を取り出したものをタプル化する作業が行われます。たとえば、生成されるzipオブジェクト内の(75, 42)あるいは(75, 42, '渡辺')は、インデックスが0の要素をタプル化したものです。

別々のリストに格納されていた点数と名前が、学生ごとにまとめられ、渡辺君のデータのタプル、西田君のデータのタプル、杉田君のデータのタプルが作られました。

タプル化したものをまとめて、zipオブジェクトを作るのがzip化です。

zip化されたオブジェクトの型は、zipオブジェクト型です。そのため、そのままでは、リストやタプルとしては使えません。そこで、`list`関数あるいは`tuple`関数を使って、リスト化あるいはタプル化を行います。

- a … zipオブジェクトをリスト化して、全要素がタプルであるリストを生成。
- b … zipオブジェクトをタプル化して、全要素がタプルであるタプルを生成。

リスト`pl`と、タプル`pt`の要素と構成要素の、値と型を調べてみましょう。

例 8-5 zip関数で結合されたリストとタプルの要素と構成要素（例 8-4の続き）

```
>>> pl[0]
(75, 42)
>>> type(pl[0])
<class 'tuple'>
>>> pl[0][1]
42
>>> type(pl[0][1])
<class 'int'>
```

```
>>> pt[1]
(56, 85, '西田')
>>> type(pt[1])
<class 'tuple'>
>>> pt[1][2]
'西田'
>>> type(pt[1][2])
<class 'str'>
```

リスト`pl`の要素（たとえば`pl[0]`）、タプル`pt`の要素（たとえば`pt[1]`）のいずれもがタプル型（`tuple`型）です。

また、リスト`pl`も、タプル`pt`も、その要素がタプルですから、構成要素をアクセスするインデックス式は、`[]`を2重に適用したものとなります。

左ページに示すように、前側のインデックスが0, 1, 2で、後ろ側のインデックスは、リスト`pl`では0, 1、タプル`pt`では0, 1, 2です。

*

ここまでは、zip化する前のリストの全要素数が一致する例で学習してきました。要素数が異なる場合は、最も短い要素数の分だけが、先頭側から取り出されてzip化されます。

- ▶ たとえば、要素数が5と2と3のリストをまとめると、先頭側の2個の要素のみがzip化されます。なお、`itertools`モジュールの`zip_longest`関数を利用すると、最も長い要素数にあわせてzip化が行えます（不足分は`None`で埋められますが、`fillvalue`での値の指定も可能です）。