

3-1 オブジェクトの長さや数を取得する len 関数



図1 オブジェクトの長さや要素の数を取得する

● len 関数の利用目的

文字列やバイト列、リストや辞書からオブジェクトの長さまたは要素の数を返します(図1)。for文やif文、while文の条件式の中でこれらの数値を比較対象やカウンタなどとして利用します。

● len 関数の書式

len 関数の書式は次になります。

len(オブジェクト)

len 関数の引数には、シーケンス(文字列、バイト列、タプル、リスト、rangeなど)またはコレクション(辞書、集合、凍結集合など)を代入して、オブジェクトの長さまたは要素の数を返します。

● 実行例

▶ 文字列の長さ

len 関数のオブジェクトに文字列を代入すると文字数が返されます。

図2の例ではそれをprint関数に渡して画面出力しています。英字だけでなく、空白も文字列としてカウントされるので結果として11が表示されます。

図2
長さや数を取得する len 関数 1
…文字列

```
print(len('Hello World'))
11 ← 文字以外にも空白もカウントされます
```

▶ リストの要素数

リストの要素数をカウントできます。図3の1行目はリストlist1に4つの要素を代入します。2行目でlen関数でlist1の要素数が返され、print関数で画面出力します。リストには4つの要素がありますので、画面には4が表示されます。

```
list1 = ["apple", "orange", "lemon", "banana"]
print(len(list1))
4
```

for文のループ処理でリストの要素分、処理を繰り返すときに、ループ回数を決めるためによく利用します

図3 長さや数を取得する len 関数 2…リスト

▶ 辞書のオブジェクト数

辞書のオブジェクトをカウントできます。

図4の1行目で変数dict1に4つのキーとオブジェクトとを組み合わせで代入します。

2行目のlen関数で変数dict1をオブジェクトとして代入すると辞書のオブジェクト数が返され、print関数で画面出力します。

辞書のオブジェクトは4つありますので画面には4が出力されます。

```
dict1 = {1:"apple", 2:"orange", 3:"lemon", 4:"banana"}
print(len(dict1))
4
```

オブジェクトを追加や削除する辞書では、オブジェクト数をプログラムでチェックできます

図4 長さや数を取得する len 関数 3…辞書

3-2 リストを扱う list 関数

● list 関数の利用目的

リストの作成に利用する関数です。リストは内部要素の値が書き換え可能なオブジェクトであり、ミュータブルなシーケンスと言えます(図5)。一般的に同種の項目の集まりを配列構造として格納するために使われます。

例えば、国語のテストの成績を格納するなどの使い方ができます。

1つのリストに同じ性質のデータ(オブジェクト)としてテストの成績を生徒番号順のように並べる順番のルールを決めて、まとめて格納できます。

また、生徒別に変数を用意してテストの成績を格納するのと比べて、リストを使った方が短いソースコードの記述で済みます。

リストに格納した成績を参照するときにも、生徒番号順に並んでいることが分かれば、リストから目的の生徒の成績を参照したり計算に活用したりするのが簡単です。

● list 関数の書式

list 関数の書式は次になります。

list([イテラブルなオブジェクト])