

3 配 列

いくつかの同じ型の変数に対して、それぞれ異なった変数の名前をつけず、全体を一つの集まりとして、全体に一つの集まりの名前（配列名という）をつけたものである。配列を定義するのは、つぎのいずれかの方法で行う。

1. DIMENSION文による配列宣言。
2. 型宣言文による配列宣言。
3. COMMON文による配列宣言。

<p>3-1 配列宣言</p>	<p>配列の名前、大きさ、次元を示すために用いる。配列の名前は、変数の名前と同様に英字ではじまる6字以内の英数字である。</p> <p>配列名の型は、型宣言文が与えられない限り、整数型、実数型の型が暗黙の宣言にしたがって決定される。型宣言文が与えられれば、その型となる。</p> <p>配列宣言文は各プログラムのはじめの部分に書く。また、そのプログラム単位においてのみ有効である。</p> <p>次元の大きさは3次元までである。</p> <p>[例] DIMENSION A(10), B(5, 3), C(4, 10, 7) INTEGER A(10), B(5, 3) REAL M(5, 3), X(10, 5, 4) COMPLEX Y(2, 4, 2) DOUBLE PRECISION K(8, 9) LOGICAL L(8)</p>
<p>3-2 配列要素の記憶装置上での順序</p>	<p>配列を構成している一つのデータを配列要素(array element)というが、この配列要素は記憶装置上、つぎのような順になっている。</p> <p>(1) 1次元の配列の場合 X(10)と配列宣言されていれば、X(1), X(2), …… , X(10)の順となる。</p> <p>(2) 2次元以上の配列の場合 2次元以上の配列の場合では、はじめの添字がさきに変化する順になる。たとえば、X(3, 3)と配列宣言されていれば、X(1, 1), X(2, 1), X(3, 1), X(1, 2), X(2, 2), X(3, 2), X(1, 3), X(2, 3), X(3, 3)の順になる。</p>
<p>3-3 配列要素の識別</p>	<p>配列宣言された配列は、添字をつけて必要な配列要素を識別することができる。添字としては、つぎに示される添字式(subscript expression)のいずれかの形で与える。このようにして識別される配列要素は、添字もっているので、添字つき変数ともいわれる。ただし、2次元以上の配列においては、各添字式はコンマで区切らなければならない。また、各添字式の値は、それに対応する集まりの大きさよりも大きすぎたり、ゼロ、負の値になってはいけない。すなわち、添字の下限はつねに1であり、上限は配列宣言</p>