

スライド1 関数 「明快入門 C」 p.124

◇ 関数 【説明】

C プログラムは[ ]の寄せ集め。

プログラムは[ ]だけでも十分に機能する。

プログラムが大きくなると、巨大な一つの main 関数だけでは、プログラムの[ ]が悪くなる。そこでプログラムを[ ]大きさの[ ]に分割する。

```
void function_name (void)
{
}

```

関数には[ ]に名前をつけてかまわないが、“[ ]”だけは特別な名前  
プログラムの中には必ず“main”という名前の関数が必要。プログラムはこの main 関数から[ ]される。

スライド2 関数

```
#include <stdio.h>
void line(void) {
    printf("—");
}
int main(void) {
    printf( "静大¥n");
    [ ]; /* 関数 line()の呼び出し
    return 0;
}

```

は以下のプログラムと同じ働き

```
#include <stdio.h>
int main(void) {
    printf( "静大¥n");
    [ ]; /* line() */;
    return 0;
}

```

スライド3 関数の引数 「明快入門 C」 p.126

◇ 関数の引数 【説明】

関数の[ ]側と[ ]側でデータのやりとりをするには[ ]を用いる。引数は関数名に接続するかっこの中に、[ ]型と[ ]名を書く。たとえば、

```
void add(int a, int b)
{
    ...
}

```

と書く。このときの変数 a や b を「 」と呼ぶ。それはどんな値が渡されるか分からない仮の引数だ

から。この関数を呼び出す側では、

```
add(100, 200); /* 値を指定。 */
```

```
add(y, z); /* 変数名を指定 */
```

のように実際のデータを記述する。これを「 」と呼ぶ。

---

#### スライド4 関数の用例

##### 【用例】

```
void print(int n)
{ /* 引数を持つ関数を定義する。 */
    int i;
    for(i=0; i<n; i++) { /* 引数 n の値の回数だけ */
        printf("hello¥n"); /* printf() を実行 */
    }
}

int main(void) {
    int i;
    i = 5;
    [ ]; /* 定数を引数として関数を呼び出す。 */
    [ ]; /* 変数を引数として関数を呼び出す。 */
    return 0;
}
```

---

#### スライド5 関数の引数 用例 「明快入門 C」 p.124

リスト 2数の平均値を求めるプログラム

```
#include <stdio.h>
double ave(double x, double y); /* [ ] */
int main(void) {
    double a,b,avdt;
    a=11.11; b=33.33;
    avdt=ave(a,b);
    print( "a=%f b=%f 平均=%f¥n" ,a,b,avdt);
}

double ave(double x, double y){
    double wk;
    wk=(x+y)/2.0;
    [ ];
}
```

---

#### スライド6 値渡し 「明快入門 C」 p.132

数値を渡す

[ ]による呼び出し Call by value ([ ]による呼び出し Call by name ではない)