

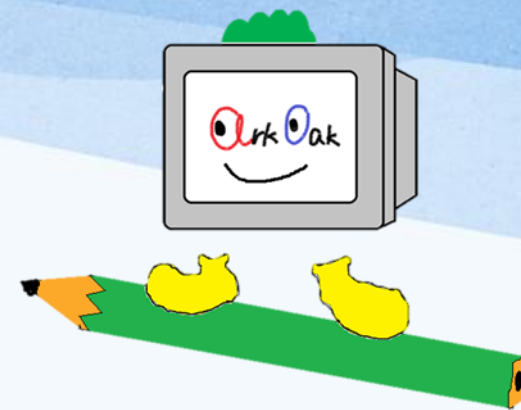
Ark Oak企画

# C言語入門

第6回



# 構造体



おーく君



# 構造体とは

複数の異なる型の変数を1つにまとめたもの

例) 成績の処理

```
int id;           //出席番号
char name[20];   //氏名
int kokugo;      //国語の点数
int suugaku;     //数学の点数
int eigo;        //英語の点数
```



1人の学生のデータ毎  
にまとめて扱いたい



# 構造体の宣言

構造体タグ

```
struct student{  
    int id; //出席番号  
    char name[20]; //氏名  
    int kokugo; //国語の点数  
    int suugaku; //数学の点数  
    int eigo; //英語の点数  
};  
  
struct student taro;
```

メンバ

struct student 型の taro という変数が定義された



# 構造体の代入・参照

```
struct student taro;  
struct student jiro;  
  
taro.id=10;  
strcpy(taro.name,"Yamada");  
taro.kokugo=100;  
taro.suugaku=85;  
taro.eigo=60;  
  
jiro=taro;  
jiro.id=11;  
strcpy(jiro.name,"次郎");  
jiro.kokugo=99;
```

構造体の変数に値を  
代入するときには

構造体の名前.変数名

というようにあいだ  
に.を入れて書く

構造体を代入すると、  
構造体の変数すべて  
が代入される。



# 構造体の代入・参照

```
void print_student(struct student s){  
    printf("出席番号:%d\n",s.id);  
    printf("氏名:%s\n",s.name);  
    printf("国語:%d\n",s.kokugo);  
    printf("数学:%d\n",s.suugaku);  
    printf("英語:%d\n",s.eigo);  
    printf("合計:%d\n",s.kokugo+s.suugaku+s.eigo);  
    printf("\n");  
}
```

参照の時にも

構造体の名前.変数名

というようにあいだに.を入れて書く



# 文字列

C言語では文字列を記憶する変数がない



文字変数char を配列として使うことで文字列を扱える

```
#include <stdio.h>
Int main(void) {
    char str[6] = {'M','A','R','I','O','¥0'};
    printf("%s¥n",str);
    return 0;
}
```

¥0 は文字列の終わりを示している



# 文字列

```
strcpy(taro.name, "Yamada");
```



```
name[0]='Y';  
name[1]='a';  
name[2]='m';  
name[3]='a';  
name[4]='d';  
name[5]='a';  
name[6]='¥0';
```

文字変数の配列nameに  
Yamadaを代入する

注意

strcpyを使うときにはプ  
ログラムの始めに

```
#include <string.h>
```

と書いておく。





# 文字列を扱う上での注意

文字列を文字変数の配列で扱うときに、配列の大きさよりも多い文字数を扱おうとするとプログラムがおかしくなってしまう。

具体的に言うと、

```
char str[20]
```

と定義された文字変数の配列に¥0を含めて20より多くの文字を入れることはできない。



# 練習（１）

## 結果

本田君と香川君の出席番号と成績をまとめたい。

結果として次のように入力するようにプログラムを書く。

出席番号:10

氏名:本田

国語:100

数学:85

英語:60

合計:245

出席番号:11

氏名:香川

国語:82

数学:73

英語:63

合計:218



# 練習（１）答え

```
#include <stdio.h>
#include <string.h>

struct student{
    int id;
    char name[20];
    int kokugo;
    int suugaku;
    int eigo;
}honda,kagawa;
void print_student(struct student s);
int main(void){
    honda.id=10;
    strcpy(honda.name,"本田");
    honda.kokugo=100;
    honda.suugaku=85;
    honda.eigo=60;
}
```

```
kagawa.id=11;
strcpy(kagawa.name,"香川");
kagawa.kokugo=82;
kagawa.suugaku=73;
kagawa.eigo=63;

print_student(honda);
print_student(kagawa);
return 0;}
void print_student(struct student s){
    printf("出席番号:%d\n",s.id);
    printf("氏名:%s\n",s.name);
    printf("国語:%d\n",s.kokugo);
    printf("数学:%d\n",s.suugaku);
    printf("英語:%d\n",s.eigo);
    printf("合計:%d\n",s.kokugo+s.suugaku+s.eigo);
    printf("\n");
}
```

# 配列と構造体

構造体を配列として扱う

```
struct student data[]={  
    {10,"Yamada",100,85,60},  
    {11,"Sato",85,95,80},  
    {12,"Takahashi",72,68,78},  
};
```

構造体の配列の初期化



## 練習（2）

山田、佐藤、高橋の3人の出席番号と成績をまとめる。  
構造体を配列にして次のような結果を出力する。

出席番号:10	出席番号:11	出席番号:12
氏名:山田	氏名:佐藤	氏名:高橋
国語:100	国語: 85	国語: 72
数学:85	数学: 95	数学: 68
英語:60	英語: 80	英語: 76



# 練習 (2) 答え

```
#include <stdio.h>
#include <string.h>

struct student{
    int id;           //出席番号
    char name[20];   //氏名
    int kokugo;      //国語の点数
    int suugaku;     //数学の点数
    int eigo;        //英語の点数
};
void print_student(struct student s);
struct student taro;int main(void){ /*
    int i;
    struct student data[]={
        {10,"Yamada",100,85,60},
        {11,"Sato",85,95,80},
        {12,"Takahashi",72,68,78},
    };
};
```

```
        for(i=0;i<3;i++){
            print_student(data[i]);
        }
    return 0;
}
void print_student(struct student s){
    printf("出席番号:%d¥n",s.id);
    printf("氏名:%s¥n",s.name);
    printf("国語:%d¥n",s.kokugo);
    printf("数学:%d¥n",s.suugaku);
    printf("英語:%d¥n",s.eigo);
    printf("合計:%d¥n",s.kokugo+s.suugaku+s.eigo);
    printf("¥n");
}
```



# 入力関数

- scanf

文字の入力

```
char c[20];  
scanf("%s",&c);  
printf("%s",c);
```

数字の入力

```
int a;  
scanf("%d",&a);  
printf("%d",a);
```

- gets

文字の入力

```
char c[20];  
gets(c);  
printf("%s",c);
```

数字の入力

```
char buffer[20];  
int a;  
gets(buffer);  
a=atoi(buffer);  
printf("%d",a);
```



# 練習(3)

標準入力から自分の名前と年齢を得て、次のように表示するプログラムを書く。

氏名:佐藤翼  
年齢:20





# 練習(3) 答え

```
char name[256];  
int age;  
  
scanf("%s",&name);  
scanf("%d",&age);  
printf("氏名:%s¥n",name);  
printf("年齢:%d¥n",age);
```

```
char name[256];  
char buffer[256];  
int age;  
  
gets(name);  
gets(buffer);  
age=atoi(buffer);  
printf("氏名:%s¥n",name);  
printf("年齢:%d¥n",age);
```



# 宿題

自分の課程・名前・年齢を入力すると、それを構造体でまとめ、次のように出力するプログラムを書く。

課程:バイオエンジニアリング課程3年  
氏名:佐藤翼  
年齢:20

