

## 補足資料：構造体

構造体とは型の一種です。型なのですが、int や float などの型とは異なる点が 2 つあります。

1. 型の名前を自分で決められる
2. 複数の変数を 1 つの変数にまとめることができる

表：int 型と構造体の違い

型の種類	名前	変数の数
整数型	int	1 つ
構造体	任意	複数可

「型の名前を自分で決められる」というのはどういうことでしょうか。具体的に見ていきましょう。構造体を作成する書式は次の通りです。

書式：構造体の作成

```
struct 構造体の名前 {  
    型 変数 A;  
    型 変数 B;  
    .  
    .  
};
```

例えば、iPhone 画面を扱う構造体「Screen」を考えてみましょう。2 つの変数（幅と高さ）を持つ Screen を作成する場合は次のコードを書きます。

List：構造体 Screen の作成

```
struct Screen{  
    int width;  
    int height;  
};
```

このように構造体を作成する際に、名前を自由に決めることができます。変数や関数の定義と似ています。「Screen」は筆者が付けた名前です。もちろん別の名前でも構いません。

また、int や float などの型では、1 つの型に対して 1 つの変数しか利用できませんが、上記の構造体 Screen を見ると「width」（幅）と「height」（高さ）の 2 つの変数が定義されていることがわかります。

## □ 構造体の宣言

構造体を宣言する場合は、次の書式で指定します。

書式：構造体の宣言

```
struct 構造体の名前 変数名;
```

例えば、先ほど紹介した構造体 `Screen` を、変数名 `myScreen` として宣言する場合は次のように書きます。変数名はこれまでの変数と同様に自由につけることができます。

List：構造体の宣言例

```
struct Screen myScreen;
```

`int` 型などの宣言と異なるのは、宣言時に「**struct**」(structure の省略形) が付くという点です。このように指定することで、構造体の宣言であるということを明示しています。

## □ 「typedef」を使用するとより便利に

構造体を宣言する際は上記のように文頭に「`struct`」を指定しますが、この「`struct`」は構造体の作成時に次のように書くことで省略できるようになります。

List：構造体 `Screen` の作成 (typedef 使用時)

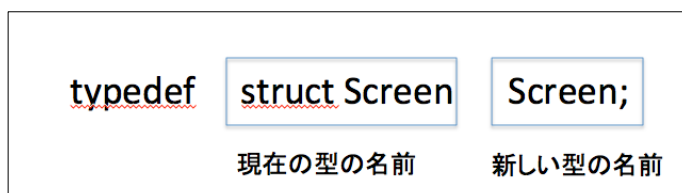
```
struct Screen{
    int width;
    int height;
};
typedef struct Screen Screen;
```

「typedef」(タイプデフ：type definition) は、型の名前を追加することができる仕組みです。書式は次のようになります。

書式：typedef の使用

```
typedef 現在の型の名前 新しい型の名前;
```

図：typedef



このように構造体の作成時に `typedef` を書いておけば、通常の変数と同様に構造体名と変数名だけの指定で、構造体を利用できるようになります。

List : 構造体 Size の宣言例

```
Screen myScreen;
```

## □ 構造体の使用方法

宣言した構造体型の変数を使用する場合は、次の書式を使用します。「.」（ドット）を指定することで構造体の中にある変数を操作します。

書式 : 構造体の使用

```
構造体型の変数名.構造体の中にある変数名;
```

構造体 Size 型の変数 `screenSize` の中にある変数に値を代入したり、代入されている値を参照する場合は、次のように指定します。

List : 変数 `screenSize` 内の変数を使用する

```
Screen myScreen; //構造体の変数の宣言
myScreen.width = 320; //幅
myScreen.height = 568; //高さ
int area = myScreen.width * myScreen.height; //面積
NSLog(@"%d", area); //面積を出力
```

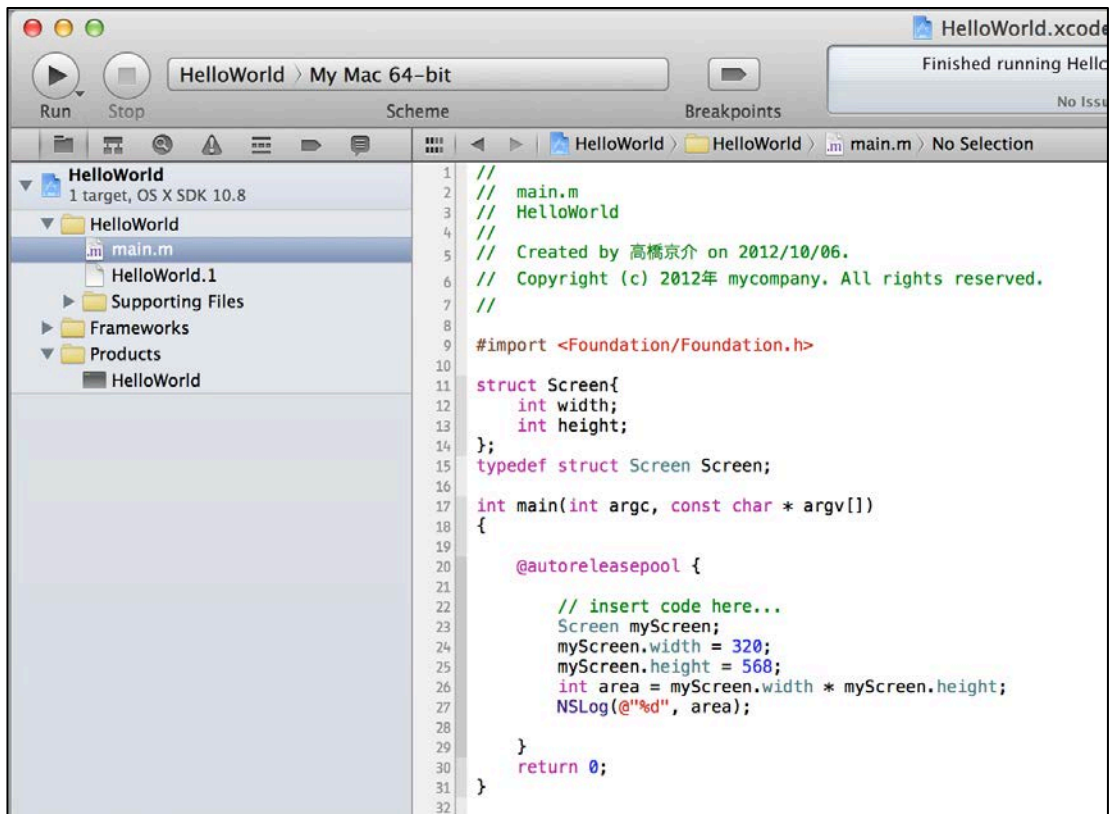
この例のように関係が深い値（横幅と高さなど）をまとめて使用する際に構造体を利用します。

## □ 構造体を使ってみよう！

それでは実際に構造体を使ってみましょう。上記で解説したコードを、xcode を使用して書いて実行します。

### 【STEP 1】

ナビゲータエリアで [main.m] を選択して以下のように構造体を使用するコードを書きます。コードを書き直したら [Run] ボタンをクリックして実行します。



### 【STEP 2】

するとデバッガエリアに「181760」と出力されます。

