

# プログラミング初級 講義実況 第6回

## 第6回講義内容

- 有効範囲
- プリプロセッサ
- 標準関数

2007/4/13

2/23

## 有効範囲

## 寿命

```
#include <stdio.h>
void func(void);

int main(void)
{
    int hoge;
    (略)
    return 0;
}

void func(void)
{
    int hoge;
    return 0;
}
```

2007/4/13

4/23

## 有効範囲

```
#include <stdio.h>
void func(void)

int main(void)
{
    int hoge;
    (略)
    return 0;
}
void func(void)
{
    hoge = 10;
}
```

2007/4/13

5/23

## 有効範囲

```
#include <stdio.h>
int hoge; // 1個目
void func(void)
int main(void)
{
    int hoge; // 2個目
    hoge = 20; (略)
    return 0;
}
void func(void)
{
    int x;
    hoge = 10; // 3個目
    for(x=0; x < 10; x++){
        int hoge; // 4個目
        hoge = x;
    }
}
```

2007/4/13

6/23

## これはできない

```
int main(void)
{
    int hoge;
    int hoge;

    return 0;
}
```

2007/4/13

7/23

## static

```
#include <stdio.h>

int main(void)
{
    static int x = 10;
    (略)
    return 0;
}
```

2007/4/13

8/23

## static

```
#include <stdio.h>
void func(void);
int main(void)
{
    func();
    func();
    func();
    return 0;
}
void func(void)
{
    static int hoge = 10;
    hoge++;
    printf("hoge = %d\n", hoge);
}
```

2007/4/13

9/23

# プリプロセッサ

## #define

- 使い方
- #define マクロ名A 文字列B

2007/4/13

11/23

## #define

```
#include <stdio.h>
#define NUMBER 10

int main(void)
{
    printf("%d", NUMBER);
    return 0;
}
```

2007/4/13

12/23

## #undef

- 使い方
- #undef 文字列A

2007/4/13

13/23

## #undef

```
#undef NUMBER
#define NUMBER 10
```

2007/4/13

14/23

## 引数つきマクロ

```
#include <stdio.h>
#define SUM( x, y ) (x+y)

int main(void)
{
    int sum;

    sum = SUM(3,5);
    printf("%d",sum);

    return 0;
}
```

2007/4/13

15/23

## 注意点

```
#define SQR(x) x*x

int x = SQR( a + b );
int y = SQR( a ) / SQR( b );
```

2007/4/13

16/23

## 展開後の注意

```
int x = SQR( a + b );
int y = SQR( a ) / SQR( b );

int x = a+b * a+b;
int y = a*a / b*b;
```

2007/4/13

17/23

## マクロにはカッコをつけるように

```
#define SQR(x) ((x)*(x))
```

2007/4/13

18/23

## #include

- 使い方
- #include <ファイル名>
- #include "ファイル名"

2007/4/13

19/23

# 標準関数

## sin関数

`double sin( double arg );`

### 使い方

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

int main(void)
{
    double value = -1.0;
    printf("%fのsinの値は%fです\n", value, sin(value));

    return 0;
}
```

2007/4/13

21/23

## strcpy関数

`char *strcpy(char* str1, char* str2);`

使い方  
#include <stdio.h>  
#include <string.h>

```
int main(void)
{
    char str[20];

    strcpy( str , "hello" );
    printf("コピーされた文字列は %s です\n",str);
    return 0;
}
```

2007/4/13

22/23

## まとめ

- 有効範囲

変数には寿命、範囲がある

- プリプロセッサ

#define や #includeのこと

- 標準関数

標準で用意されている関数がたくさん存在する

2007/4/13

23/23