# 第8章

编译预处理

# 编译预处理

- 以#开头
- 在编译之前执行
  - 宏定义
  - 文件包含
  - 条件编译

# 宏定义

# 不带参数的宏定义

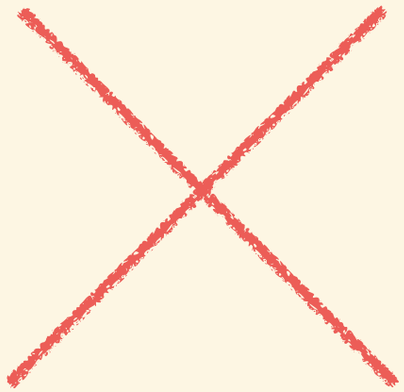
```
#define <标识符> <替换序列>
#define PI 3.14
```

- 宏名一般用全大写字母表示
- 只做替换，不做语法检查
- 不替换字符串中的宏
- 用\换行
- 层层替换

```c
#include <stdio.h>

#define PI 3.14159265358979

int main()
{
    float r =10.0;

    printf("PI: %f\n", PI);
    printf("PI/2: %f\n", PI/2);
    printf("area: %f\n", PI*r*r);
    printf("circumference: %f\n", 2*PI*r);

    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    float r =10.0;

#define PI 3.14159265358979
    printf("PI: %f\n", PI);
    printf("PI/2: %f\n", PI/2);
    printf("area: %f\n", PI*r*r);
    printf("circumference: %f\n", 2*PI*r);

    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    float r =10.0;

    printf("PI: %f\n", PI);
#define PI 3.14159265358979
    printf("PI/2: %f\n", PI/2);
    printf("area: %f\n", PI*r*r);
    printf("circumference: %f\n", 2*PI*r);

    return 0;
}
```



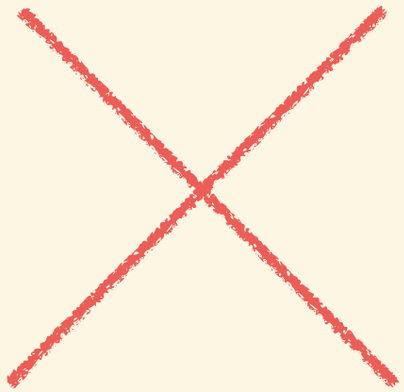
```
pi.c:9:24: error: use of undeclared identifier 'PI'
    printf("pi: %f\n", PI);
                       ^
1 error generated.
```

```c
#include <stdio.h>

#define X 3.14
#define PI X

int main()
{
    float r =10.0;

    printf("pi: %f\n", PI);
    printf("pi/2: %f\n", PI/2);
    printf("area: %f\n", PI*r*r);
    printf("circumference: %f\n", 2*PI*r);

    return 0;
}
```

```c
#include <stdio.h>

#define PI X
#define X 3.14

int main()
{
    float r =10.0;

    printf("pi: %f\n", PI);
    printf("pi/2: %f\n", PI/2);
    printf("area: %f\n", PI*r*r);
    printf("circumference: %f\n", 2*PI*r);

    return 0;
}
```

```c
#include <stdio.h>

#define PI X

int main()
{
    float r =10.0;

#define X 3.14
    printf("pi: %f\n", PI);
    printf("pi/2: %f\n", PI/2);
    printf("area: %f\n", PI*r*r);
    printf("circumference: %f\n", 2*PI*r);

    return 0;
}
```

```c
#include <stdio.h>

#define PI X

int main()
{
    float r =10.0;

    printf("pi: %f\n", PI);
#define X 3.14
    printf("pi/2: %f\n", PI/2);
    printf("area: %f\n", PI*r*r);
    printf("circumference: %f\n", 2*PI*r);

    return 0;
}
```



```
pi.c:9:24: error: use of undeclared identifier 'X'
    printf("pi: %f\n", PI);
                       ^
pi.c:3:12:       expanded from macro 'PI'
#define PI X
           ^
1 error generated.
```

```c
#include <stdio.h>

#define M 3
#define N M + 1
#define NN N * N / 2

int main()
{
    printf("%d\n", NN);
    printf("%d\n", 5 * NN);

    return 0;
}
```

```c
#include <stdio.h>

#define M 3
#define N M + 1
#define NN N * N / 2

int main()
{
    printf("%d\n", NN);
    printf("%d\n", 5 * NN);

    return 0;
}
```

```
6
18
```

```c
#include <stdio.h>

#define VALUES 1, 2, 3, 4, 5

int main()
{
    int a[] = {VALUES};
    int i;
    int n = sizeof(a) / sizeof(a[0]);

    for (i = 0; i < n; i++)
        printf("%d\n", a[i]);

    return 0;
}
```

# 带参数的宏定义

*#define* <宏名>（<参数列表>）<替换列表>

# 求最大值

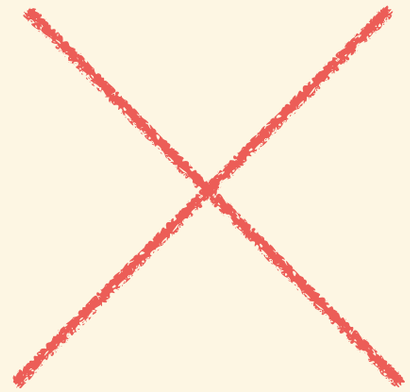
```c
#include <stdio.h>

#define MAX(x, y) x > y ? x : y

int main()
{
    printf("%d\n", MAX(3, 5));

    return 0;
}
```

# 求圆面积

```c
#include <stdio.h>

#define PI 3.14
#define AREA(r) PI * r * r

int main()
{
    int x = 1;

    printf("Area = %f", AREA(x + 1));

    return 0;
}
```

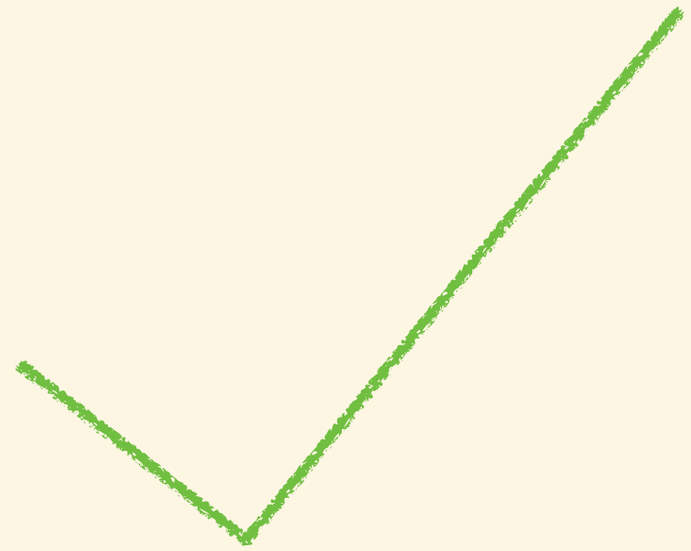
# 求圆面积

```c
#include <stdio.h>

#define PI 3.14
#define AREA(r) PI * r * r

int main()
{
    int x = 1;

    printf("Area = %f", AREA(x + 1));

    return 0;
}
```



# 求圆面积

```c
#include <stdio.h>

#define PI 3.14
#define AREA(r) PI * (r) * (r)

int main()
{
    int x = 1;

    printf("Area = %f", AREA(x + 1));

    return 0;
}
```



# 取消宏定义

```
#undef <宏名>

#undef PI
```

# 求圆面积

```c
#include <stdio.h>

#define PI 3.14

int main()
{
    int x = 1;

#define SQUARE(x) (x) * (x)
    printf("Area = %f", PI * SQUARE(x));
#undef SQUARE

    /* SQUARE not available here */

    return 0;
}
```

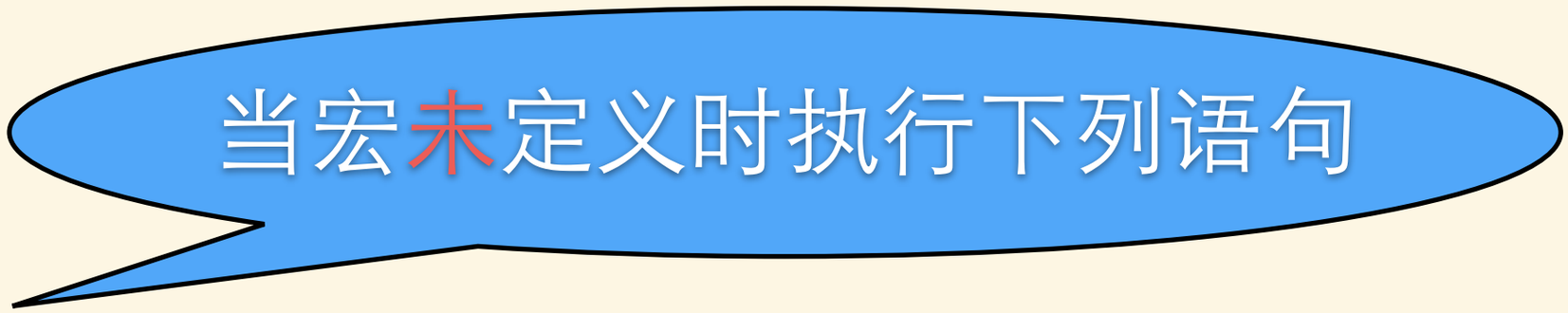
# 宏相对函数的优点？

# 条件编译

```c
#include <stdio.h>

int main()
{
#ifdef __unix__
    printf("Unix\n");
#elif defined __APPLE__
    printf("Apple\n");
#elif defined _WIN32
    printf("Windows\n");
#else
    printf("Other system\n");
#endif

    return 0;
}
```

当宏被定义时执行下列语句

```c
#include <stdio.h>

int main()
{
#ifndef _WIN32
    printf("Goodbye Windows!\n");
#else
    printf("Welcome to Windows\n");
#endif

    return 0;
}
```



```c
#include <stdio.h>

#define VERBOSE 2

int main()
{
#if VERBOSE == 1
    printf("Debug message\n");
#elif VERBOSE == 2
    printf("More messages\n");
#endif

    return 0;
}
```

当表达式为真时执行

```c
#include <stdio.h>

#define FLAG 0

int main()
{
#ifdef FLAG
    printf("FLAG defined\n");
#endif

#if FLAG
    printf("FLAG is true\n");
#endif

#undef FLAG

#ifdef FLAG
    printf("FLAG defined\n");
#endif

#if FLAG
    printf("FLAG is true\n");
#endif

    return 0;
}
```

```c
#include <stdio.h>

#define FLAG 0

void f()
{
#ifdef FLAG
    printf("FLAG defined\n");
#endif

#if FLAG
    printf("FLAG is true\n");
#endif
}

int main()
{
    f();

#undef FLAG

    f();

    return 0;
}
```

# 文件包含

# 文件包含

```
#include <foo.h>
```

```
#include “foo.h"
```

- 可包含任何文件
- 相当于在该位置插入文件的内容

# 如何组织你的工程?

- 声明放在.h文件
- 定义放在.c文件
- 相似功能放在同一模块

# 计算圆的面积和周长

# circle.h

```
#ifndef __CIRCLE_H_
#define __CIRCLE_H_

#define PI 3.14

float area(float r);

float circumference(float r);

#endif
```

# circle.c

```c
#include "circle.h"

float area(float r)
{
    return PI * r * r;
}

float circumference(float r)
{
    return 2 * PI * r;
}
```

# main.c

```c
#include <stdio.h>
#include "circle.h"

int main()
{
    float r;

    scanf("%f", &r);

    printf("area = %f\n", area(r));
    printf("circumference = %f\n", circumference(r));

    return 0;
}
```