

基本数据类型

基本数据类型

- 整型
- 浮点型
- 字符型
- 布尔型
- id类型
- 基本数据类型转换

基本数据类型

整型

- int
- short
- long
- long long

基本数据类型

浮点型

- float
- double
- 科学计数法

基本数据类型

字符型

- char
- 特殊的转义字符

\n : 换行

\t : 制表符

\\ : 反斜杠

\": 双引号

\': 单引号

%% : %

基本数据类型

布尔型

- YES
- NO
- 定义后当做整数一样进行赋值运算和输出

基本数据类型

id类型

- 表示存储任意类型的对象

基本数据类型

基本数据类型转换

- 隐式转换
- 强制转换

运算符和表达式

运算符和表达式

- 算数运算符
- 赋值运算符
- 扩展后的赋值运算符
- 比较运算符
- 逻辑运算符
- 三目运算符

运算符和表达式

算数运算符

- +
- -
- *
- /
- %
- ++
- --



运算符和表达式

赋值运算符

- =

运算符和表达式

扩展后的赋值运算符

- `+=`
- `--`
- `*=`
- `--`
- `%=`

运算符和表达式

比较运算符

- >
- >=
- <
- <=
- ==
- !=

运算符和表达式

逻辑运算符

- &&
- ||
- !

运算符和表达式

三目运算符

- `(expression) ? if-true-statement :if-false-statement;`

循环语句

循环语句

- for
- while
- do...while
- 循环跳转语句

循环语句

- for

```
for ( init_expression; loop_condition; loop_expression) {  
    program statment  
}
```

init_expression: 循环开始前设置初始值，仅执行一次

loop_condition :循环所需条件，每次进行判断

loop_expression:循环表达式，每次循环执行一次

循环语句

- while

```
while(expression) {  
    program statment  
}
```

循环语句

- **do...while**

```
do {  
    program statement  
} while (expression);
```

循环语句

循环跳转语句

- break
- continue

选择语句

选择语句

- if
- switch

选择语句

- if

```
if (expression) {  
    program statement  
} else if( expression) {  
    program statement  
} else {  
    program statement  
}
```

选择语句

- switch

```
switch(expression): {  
    case value1: {  
        program statement  
        break;  
    } case value2: {  
        program statement  
        break;  
    } default: {  
        program statement  
        break;  
    }  
}
```

预处理程序

预处理程序

- 文件包含
- 宏定义
- 条件编译
- 错误、警告处理
- 编译器控制

预处理程序

文件包含

- `#include` 有重复包含的问题
- `#import` 解决了重复包含的问题

■ 预处理程序

宏定义

- 格式: `#define TRUE 1`

预处理程序

条件编译

- `#if + #elif + #else + #endif`
- `#ifdef = #if define`
- `#elif = #else if`

预处理程序

错误、警告处理

- #error 预报编译前的错误
- #warning 编译器提醒这个警告

■ 预处理程序

编译器控制

- #pragma mark

枚举

枚举

- 用处
- NS_ENUM
- NS_OPTIONS

枚举

用处

- 为了方便阅读，养成良好的习惯，避免在代码中临时定义常量，应该多用枚举类型
- 定义后当做整数一样进行赋值运算输出

枚举

- NS_ENUM

用连续的数字来表示单枚举项

枚举

- NS_OPTIONS

使用位移的整数表示可以多选的枚举项

枚举

NS_ENUM和NS_OPTIONS并没有多大的区别只是，为了代码统一，约束下编码规则