

1. 格式

```
165 格式语法:
166 变量名 = 数据;
167
168 int num;
169 num = 100;
```

2. 赋值符号

```
171 1). 这个等号和你数学中的等号是完全不同的。
172 叫做赋值符号。
173 赋值符号的作用：将右边的数据 存储到 左边的变量所代表的空间中去。
174
```

3. int型变量赋值注意点

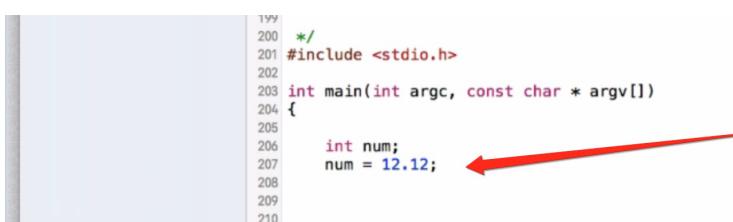
1>超过范围--报错



A screenshot of an Xcode code editor. The file is named 'main.c'. Line 199 contains the assignment statement `num = 22212121212121212121212121212123334546576;`. A red arrow points from the number 222... to a red error bar at the end of the line. The error message 'Integer literal is too large to be represented in any integer type' is displayed in a tooltip.

```
179 int num;
180 num = 199;
181
182
183
184
185
186
187
188
189
190
191
192 */
193 #include <stdio.h>
194
195 int main(int argc, const char * argv[])
196 {
197
198     int num;
199     num = 22212121212121212121212121212123334546576;
200
201 }
```

2>赋值小数--不报错



A screenshot of an Xcode code editor. The file is named 'main.c'. Line 207 contains the assignment statement `num = 12.12;`. A red arrow points from the decimal value 12.12 to the assignment operator =. The code editor shows no errors or warnings for this line.

```
199 */
200 #include <stdio.h>
201
202 int main(int argc, const char * argv[])
203 {
204
205     int num;
206     num = 12.12;
207
208
209
210
```

总结

- a. 如果给的数据太太太大，直接报语法错误。
- b. 如果我们为int变量赋值了1个小数，只会存储这个小数的整数部分。

验证

A screenshot of a C IDE showing a program and its output. The code is as follows:

```
201 #include <stdio.h>
202
203 int main(int argc, const char * argv[])
204 {
205     int num;
206     num = 1234567890;
207
208     printf("%d\n", num);
209
210     printf("A \n");
211     return 0;
212 }
```

The variable `num` is assigned the value `1234567890`. A red arrow points from this assignment to the output window. The output window shows the value `12` and the character `A`, with the message `Program ended with exit code: 0`.

3>超过范围比较小

A screenshot of a C IDE showing a program and its output. The code is as follows:

```
201 */
202 #include <stdio.h>
203
204 int main(int argc, const char * argv[])
205 {
206
207     int num;
208     num = -2200000000;
209
210
211     printf("%d\n", num);
212
213     printf("A \n");
214 }
```

The variable `num` is assigned the value `-2200000000`. A red arrow points from this assignment to the output window. The output window shows the value `-2094967296` and the character `A`, with the message `Program ended with exit code: 0`.

因为变量的类型是int类型的，所以为int变量赋值的时候，赋值的数据不能超过指定的范围。

- a. 如果给的数据太太太，直接报语法错误。
- b. 如果我们为int变量赋值了1个小数，只会存储这个小数的整数部分，不会做四舍五入。
- c. 如果给的数据超过的范围并不多，这个时候存储的是随机值。

3.float型变量赋值注意点

1>

A screenshot of a C IDE showing a list of notes for float assignment. The notes are:

- 3). 为float类型的变量赋值。
 - a. 在C语言中如果我们直接写1个小数，那么这个小数的类型是double类型的

现在给一个float变量用赋值double小数赋值，会丢失精度

```
209 */  
210 #include <stdio.h>  
211  
212 int main(int argc, const char * argv[])  
213 {  
214     float f1;  
215     f1 = 12.12;
```

2>正确写法

```
209 */  
210 #include <stdio.h>  
211  
212 int main(int argc, const char * argv[])  
213 {  
214     float f1;  
215     f1 = 12.12f;
```

总结

191 3). 为float类型的变量赋值.
192
193 a. 在C语言中如果我们直接写1个小数.那么这个小数的类型是double类型的。
194 如果我们就是希望这个小数是1个float类型的.那么就在这个小数的后面加1个f/F
195 代表前面的这个小数的类型是1个float类型的。
196
197 记住这句话：为float变量赋值的时候.赋值的小数后面要加1个f
198
199

3>float的精度问题

191 3). 为float类型的变量赋值.
192
193 a. 在C语言中如果我们直接写1个小数.那么这个小数的类型是double类型的。
194 如果我们就是希望这个小数是1个float类型的.那么就在这个小数的后面加1个f/F
195 代表前面的这个小数的类型是1个float类型的。
196
197 记住这句话：为float变量赋值的时候.赋值的小数后面要加1个f
198
199
200 b. float只能精确保存有效位数不超过7位的小数。
201 如果赋值的数据的有效位数超过了7位.那么超出的部分就不会精确保存。

```
222 */  
223 #include <stdio.h>  
224  
225 int main(int argc, const char * argv[])  
226 {  
227     float f1;  
228     f1 = 12.123458723f;  
229  
230     printf("%f\n", f1);  
231  
232  
233 //     int num:
```

```
12.123459  
Program ended with exit code: 0
```

4.double型变量赋值注意点

```
204 4). 为double类型的变量赋值。  
205  
206 a. 在C语言中如果我们直接写1个小数，那么这个小数的类型是double类型的。  
207 所以可以直接将1个小数赋值给1个double变量。  
208  
209 b. double只能精确保存有效位数不超过16位的小数。  
210 如果赋值的数据的有效位数超过了16位，那么超出的部分就不会精确保存。  
211
```

5.char型变量赋值注意点

1>

```
211  
212  
213 5). 为char类型的变量赋值。  
214  
215 a. char变量中只能存储单个字符，并且这单个字符还要用单引号引起来。  
216  
217  
218  
219
```

```
240 */  
241 #include <stdio.h>  
242  
243 int main(int argc, const char * argv[]){  
244  
245     char ch;  
246     ch = 'a'; //
```

2>

```
211  
212  
213 5). 为char类型的变量赋值。  
214  
215 a. char变量中只能存储单个字符，并且这单个字符还要用单引号引起来。  
216  
217 b. 如果给的字符太多，保存最后1个。  
218
```

```
211  
212  
213 Day02-IDE的使用与数据类型 > 08-数据的处理 > main.c > main0 | < ▲ >  
243 #include <stdio.h>  
244  
245 int main(int argc, const char * argv[]){  
246  
247     char ch;  
248     ch = 'abcx'; // Implicit conversion from 'int' to 'char' changes value from 1633837944 to 120  
249  
250  
251     printf("%c\n", ch);  
252  
253  
254 //     double d1;  
255 //     d1 = 12.12;  
256  
257 //     float f1;  
258 //     f1 = 12.123458723f;  
259 //  
260 //
```

x Program ended with exit code: 0

3>

219

c. 空格也是1个字符，所以也可以存储1个空格

```
245 #include <stdio.h>
246
247 int main(int argc, const char * argv[])
248 {
249
250     char ch; 
251     ch = ' ';
252 }
```

4>

220
221
222
223

d. char变量中无法存储中文。

在C语言中1个中文字符占3个字节 而1个char变量占1个字节。

```
245 #include <stdio.h>
246
247 int main(int argc, const char * argv[])
248 {
249
250     char ch; 
251     ch = '中';
252
253
254     printf("%c\n",ch);
255 }
```

@"无色大师的笔记"