

Default ctor

```
fraction::fraction() : numerator(0), denominator(1)
```

默认初始化为 0/1。

Ctor takes two integers as parameters

```
fraction::fraction(int a, int b) : numerator(a), denominator(b)
```

初始化为 a/b，加了分母不等于 0 的断言。自动约分，分子为 0 时分母置为 1。

Copy ctor

```
fraction::fraction(const fraction& p)
```

复制构造函数

```
const fraction& fraction::operator= (const fraction& p)
```

赋值符重载

Arithmetical operators: + - * /

```
const fraction& fraction::operator+(const fraction& p) const
```

加法重载，返回值为化简后的结果。

```
const fraction& fraction::operator-(const fraction& p) const
```

减法重载，返回值为化简后的结果。

```
const fraction& fraction::operator*(const fraction& p) const
```

乘法重载，返回值为化简后的结果。

```
const fraction& fraction::operator/(const fraction& p) const
```

除法重载，返回值为化简后的结果。

Relational operators: < <= == != >= >

```
bool fraction::operator==(const fraction& p) const
```

==号重载。

```
bool fraction::operator!=(const fraction& p) const
```

!=号重载。

```
bool fraction::operator<(const fraction& p) const
```

<号重载。

```
bool fraction::operator<=(const fraction& p) const
```

<=号重载。

```
bool fraction::operator>(const fraction& p) const
```

>号重载。

`bool fraction::operator>=(const fraction& p) const`
>=号重载。

Insertor and extractor for streams

`istream& operator>>(istream& in, fraction& p)`
输入流重载，会提示输入分母和分子，自动化简，有分母不等于 0 的断言。
`ostream& operator<<(ostream& out, fraction& p)`
输出流重载，以(-)a/b 的形式输出，当分母为 1 时只输出分子。

Typecast to double

`fraction::operator double() const`

将 fraction 类型转化为 double 类型。

To string

`fraction::operator std::string() const`

将 fraction 类型转化为 string 类型。

Conversion from a finite decimal string

`fraction::fraction(string s)`

将 string 类型转化成 fraction 类型。