# 其他

## 1. XML和JSON的区别

### 1.1. XML

* 应用广泛，可扩展性强，被广泛应用各种场合；
* 读取、解析没有JSON快；
* 可读性强，可描述复杂结构。

### 1.2. JSON

* 结构简单，都是键值对；
* 读取、解析速度快，很多语言支持；
* 传输数据量小，传输速率大大提高；
* 描述复杂结构能力较弱。

## 2. 开源协议

Free Software Licenses

Free Software Licenses

[Licenses | Choose a License](https://choosealicense.com/licenses/)

## 3. 方法中什么时候需要进行参数校验，什么时候不需要

需要进行参数校验的场景：

* 调用频次低的方法。
* 执行时间开销很大的方法，参数校验时间几乎可以忽略不计。
* 需要极高稳定性和可用性的方法。
* 对外提供的开放接口。

不需要进行参数校验的场景：

* 极有可能被循环调用的方法。
* 底层方法（调用频率高，且不太可能出错）。
* 被声明为private，只会被自己代码调用的方法。

## 4. 版本号命名方式

版本格式：主版本号.次版本号.修订号

规则：

1. 主版本号：当做了不兼容的API修改。
2. 次版本号：当做了向下兼容的功能性新增。
3. 修订号：当做了向下兼容的问题修正。

## 5. 如何理解非对称加密

首先，每个用户都有两把钥匙，一把公钥一把私钥。公钥是对外发布的，所有人都看的到所有人的公钥，私钥是自己保存，每个人都只知道自己的私钥而不知道别人的。

用该用户的公钥加密后只能该用户的私钥才能解密。这种情况下，公钥是用来加密信息的，确保只有特定的人（用谁的公钥就是谁）才能解密该信息。

还有第二种情况，公钥是用来解密信息的，确保让别人知道这条信息是真的由我发布的，是完整正确的。接收者由此可知这条信息确实来自于拥有私钥的某人，这被称作数字签名，公钥的形式就是数字证书。

## 6. Java算法题模板

import java.io.\*;
import java.util.\*;
import java.text.\*;
import java.math.\*;
import java.util.regex.\*;

public class Main {
 public static void main(String[] args) {
 Scanner scanner = new Scanner(System.in);

 System.out.println("result");
 }
}