一、

为了方便管理，把parser单独拆到文件中

Parser接受html文本作为参数，返回一颗dom树

二、

使用FSM来实现Html的分析

在html标准中，已经规定了html的状态

Toy-browser只挑选其中一部分状态，完成一个最简版本

三、  
解析标签：开始标签、结束标签、自封闭标签

四：

在状态机中，除了状态迁移，我们还会加入业务逻辑

我们在标签结束状态提交标签token

五：

属性值分为单引号、双引号、无引号三种写法，因此需要较多状态处理

处理属性的方式跟标签类似

属性结束时，我们把属性加到标签token上

六：

从标签构建DOM树的基本技巧是使用栈

遇到开始标签时创建元素并入栈，遇到jie’shu

结束标签shi时chu出栈

自封闭节点可视为入后立刻出栈

任何元素的父元素是它入栈前的栈顶

七：

文本节点与自封闭标签处理类似

多个文本节点需要合并

Css计算

第一步：

遇到style标签时，把css规则保存起来

这里我们调用css parser来分析css规则

这里我们必须要仔细研究此庫分析css规则的格式

第二步：

当我们创建一个元素后，立即计算css

理论上，当我们分析一个元素时，所有css规则已经收集完毕

在真实浏览器中，可能遇到写在body的style标签，需要重新css计算的情况，这里我们忽略

第三步：

在computeCSS函数中，需要知道元素的所有父元素才能判断元素与规则是否匹配

我们从上一步骤的statck，可以获取本元素所有的父元素

因为我们首先获取的是当前元素，所以我们获得和计算父元素匹配的顺序是从内向外

第四步：

选择器也要从当前元素向外排列

复杂选择器要拆成针对单个元素的选择器，用循环匹配父元素队列

第五步：

根据选择器的类型和元素属性，计算是否与当前元素匹配

这里仅仅实现可三种基本选择器，实际的浏览器中要处理复合选择器

第六步：

一旦选择匹配，就应用选择器到元素上，形成computedStyle

第7步

Css规则根据specificity和后来优先规则覆盖

Specificity是个四元组，越左边权重越高

一个css规则的specificity根据包含的简单选择器相加而成