HTML Parser

1. **What's the job of HTML parser?**

   Parse the HTML markup into a parse tree.

2. **How does HTML parser Defined?**

   — HTML cannot easily be defined by a context free grammar to be parsed using the regular top down or bottom up parsers. The reasons are:

   - The forgiving nature of the language.
   - The fact that browsers have traditional error tolerance to support well known cases of invalid HTML.
   - The parsing process is reentrant(可折返的). In HTML, dynamic code (such as script elements containing document.write() calls) can add extra tokens, so the parsing process actually modifies the input.

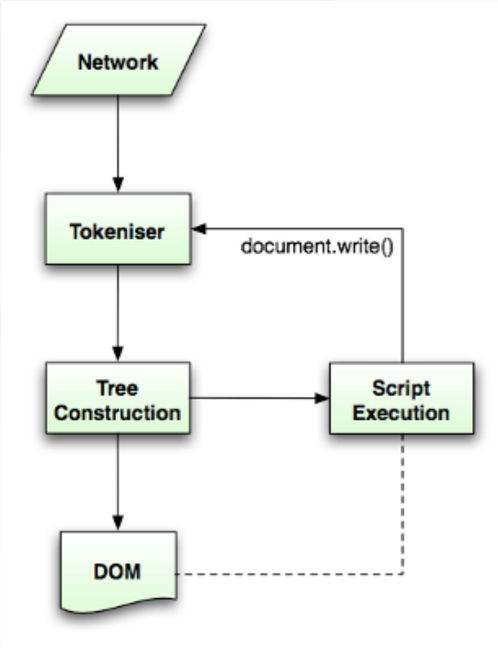
   — A formal format for defining HTML is DTD.

   — HTML DTD(Document Type Definition):

   The format contains definitions for all allowed elements, their attributes and hierarchy. There are a few variations of the DTD. The strict mode conforms solely to the specifications but other modes contain support for markup used by browsers in the past. The purpose is backwards compatibility with older content.

3. **How does HTML parser work?**

   - Tokenization is the lexical analysis, parsing the input into tokens. Among HTML tokens are start tags, end tags, attribute names and attribute values.
   - The tokenizer recognizes the token, gives it to the tree constructor, and consumes the next character for recognizing the next token, and so on until the end of the input.
   - DOM(Document Object Model):The output tree (the "parse tree") is a tree of DOM element and attribute nodes. It is the object presentation of the HTML document and the interface of HTML elements to the outside world like JavaScript.



THE HTML PARSING ALGORITHM

Tokenization

1. **A state machine of tokenization**
   — Each state consumes one or more characters of the input stream and updates the next state according to those characters.

   —The decision is influenced by the current tokenization state and by the tree construction state.

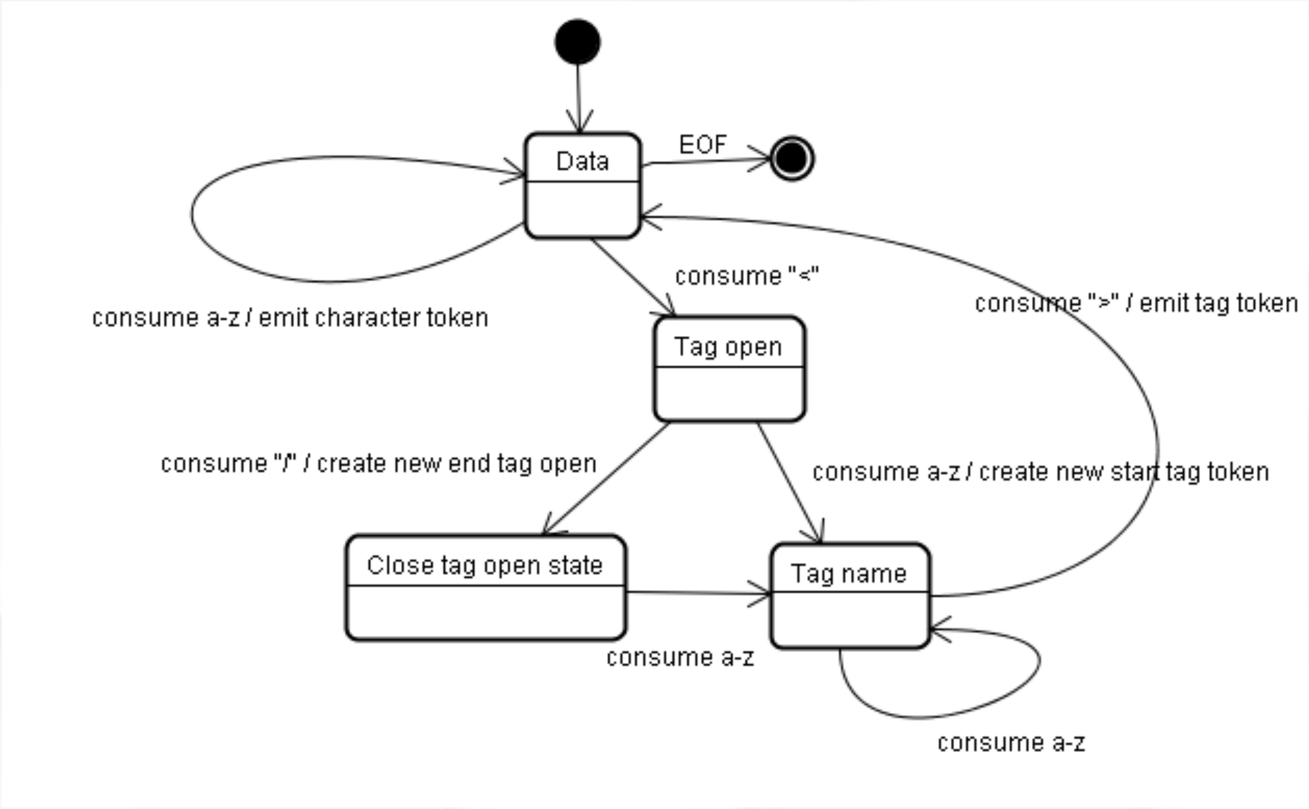
   — The same consumed character will yield different results for the correct next state, depending on the current state.

2. **An example of tokenization**
   — Tag open state

   — close tag open state

   — data state

```
<!DOCTYPE html>
<html>
<body>
    Hello world
</body>
</html>
```



STATE MACHINE OF TOKENIZATION