1. var isFnRegex = /^\s\*(?:function)?\\*/;
2. "1 2 3".split(/ +/) //[1,2,3]
3. 判断是不是generator对象 就是执行出来的实例
   1. function isGenerator(obj) {
   2. return 'function' == typeof obj.next && 'function' == typeof obj.throw;
   3. }
4. 是不是Promise函数
   1. function isPromise(obj) {
   2. return 'function' == typeof obj.then;
   3. }
5. 是不是generator函数
   1. function isGeneratorFunction(obj) {
   2. var constructor = obj.constructor;
   3. if (!constructor) return false;
   4. if ('GeneratorFunction' === constructor.name || 'GeneratorFunction' === constructor.displayName) return true;
   5. return isGenerator(constructor.prototype);
   6. }
6. const attribute = /^\s\*([^\s"'<>\/=]+)(?:\s\*(=)\s\*(?:"([^"]\*)"+|'([^']\*)'+|([^\s"'=<>`]+)))?/
7. // could use https://www.w3.org/TR/1999/REC-xml-names-19990114/#NT-QName
8. const ncname = '[a-zA-Z\_][\\w\\-\\.]\*'
9. const qnameCapture = `((?:${ncname}\\:)?${ncname})`
10. const startTagOpen = new RegExp(`^<${qnameCapture}`)
11. const startTagClose = /^\s\*(\/?)>/
12. const endTag = new RegExp(`^<\\/${qnameCapture}[^>]\*>`)
13. const doctype = /^<!DOCTYPE [^>]+>/i
14. // #7298: escape - to avoid being pased as HTML comment when inlined in page
15. const comment = /^<!\--/
16. const conditionalComment = /^<!\[/
17. const decodingMap = {
18. '&lt;': '<',
19. '&gt;': '>',
20. '&quot;': '"',
21. '&amp;': '&',
22. '&#10;': '\n',
23. '&#9;': '\t'
24. }
25. const encodedAttr = /&(?:lt|gt|quot|amp);/g
26. const encodedAttrWithNewLines = /&(?:lt|gt|quot|amp|#10|#9);/g