

词法结构

- 字符集

- JS使用Unicode字符集编写

字符集有ASCII, GB2312, BIG5, GB18030, Unicode等

http://www.cnblogs.com/skynet/archive/2011/05/03/2035105.html#_3.1.UCS_&_UNICODE

- JS区分大小写, 但HTML不区分, 但XHTML区分

XHTML严格的说是区分大小写的, 但即便没有大小写, 浏览器还是能解析渲染

- JS忽略标识符之间的空格和换行符 (除一种情况)

- JS会忽略空格符 (\u0020), 水平制表符 (\u0009), 垂直制表符 (\u000B), 换页符 (\u000C), 不间断空白 (\u00A0), 子节序标记 (\uFEFF), 以及所有Zs类别的字符

Zs类别代表没有标识符号但不属于控制或格式字符的空格字符

- JS将一下字符识别为行结束符: 换行符 (\u000A), 回车符 (\u000D), 行分隔符 (\u2028), 段分隔符 (\u2029)。回车符和换行符在一起被解析为一个单行结束符

- Unicode转义序列: 使用6个ASCII字符来表示任意16位Unicode内码。以\u为前缀, 其后跟随4个十六进制数。转义写法可以用在字符串直接量、正则表达式、标识符 (除关键字外)。'A\u0041' === 'AA' // => true

- JS注释:

- //单行注释
 - /*一段注释*/ //另一段注释
 - /* *多行注释 *多行注释 */

- 直接量: 程序中直接使用的数据值

- 数字直接量: 11
 - 字符串直接量: 'javascript'
 - 布尔值: true
 - 正则表达式直接量: /javascript/gi
 - null
 - 数组直接量: [1,2,3,4]
 - 对象直接量: {x:1, y:2}

还有一些复杂的直接量

- 标识符和保留字

- 标识符: 用来对变量和函数进行命名, 或某些循环语句中的跳转位置的标记
 - 标识符必须以字母, 下划线 (_), 或美元符 (\$) 开始
 - 后续的字符可以是字母, 数字, 下划线或美元符
 - 数字不允许作为首字母
 - 出于可移植性和易于书写的考虑, 一般只用ASCII字母是数字写标识符, 但JS允许标识符中出现Unicode字符全集中的字母和数字。

- 保留字: JS中的某些标识符作为关键字, 这些关键字就不能再作为其他的标志符了
break, delete, function, return, typeof, case, do, if, switch, var, catch, else, in, this, void, continue, false, instanceof, throw, while, debugger, finally,

new , true , with , default , for , null , try , (class , const , export , extends , import , super) (es6) , enum

- 一些关键字只在严格模式下为保留字
implements , let , private , public , yield , interface , package , protected , static
- 在严格模式下, arguments , eval虽不是保留字,但不能作为变量名,函数名和参数名
- JS在ES3中将Java所有的关键字都作为保留字,应当避免使用这些关键字作为标识符
- JS还预定义了很多全局变量和函数,应当避免将这些作为变量名和函数名

- 可选的分号

- JS使用 (;) 将语句分隔开
- JS并不在所有换行处都填补分号,只有在缺少了分号就无法正确解析的时候,JS才会填补分号
因此,若语句以 "(" , "[" 开头的话,若前一句没有分号,则很容易被作为函数或数组调用。因此用分号明确的标记语句的结束
- 涉及return , break和continue的语句,若其后紧随着换行,则JS会填补分号。因此,在return , break , continue和随后的表达式之间不能有换行
- "++" , "--" 运算符作为表达式后缀应与表达式在同一行,否则,行尾将填补分号