Part I

Core JavaScript

Chapter 2

Lexical Structure

Cấu trúc ngữ pháp của ngôn ngữ là thiết lập các quy tắc mà bạn phải tuân theo khi sử dụng 1 ngôn ngữ. Nó là tầng syntax thấp nhất của ngôn ngữ.

**2.1 Character Set**

JavaScript programs được viết bằng việc sử dụng kí tự Unicode là tập hợp của ASCII, Latin-1 và hỗ trợ gần như mọi ngôn ngữ viết hiện tại được sử dụng trong bảng chữ cái.

**2.1.1 Case Sensitivity (phân biệt hoa thường)**

Javascript là 1 loại ngôn ngữ phân biệt hoa thường. Như biến, hàm hay các các định danh khác phải viết đúng cú pháp chuẩn. Như từ khóa *while* phải viết đúng là “while” không phải là “While” hoặc “WHILE” . Tương tự *online, Online, OnLine, ONLINE* là tên của 4 biến khác nhau.

Nhớ rằng HTML là không phân biệt hoa thường(XHTML cũng thế). Vì nó liên kết chặt chẽ với client-side JavaScript , sự khác biệt này có thể gây nhầm lẫn. Nhiều Object và property của *client-side* JavaScript thì có tên giống với các tag và attribute HTML. Trong khi những tên tag và attribute này có thể được dùng bất kỳ trường hợp nào trong HTML, trong javascript chúng phải tất cả bằng chữ thường, chẳng hạn attribute xử lí sự kiện onclick trong HTML phải viết đúng như thế này *onClick*, nhưng trong javascript hay trong XHTML phải viết đúng như thế này *onclick.*

**2.1.2 Whitespace, Line Breaks, format control characters**

Javascript bỏ những khoảng trắng xuất hiện ở giữa các tokens và xuống dòng trong chương trình. Bởi vì bạn có thể sử dụng khoảng trắng, xuống dòng tùy thích trong chương trình của mình, bạn có thể định dạng và thụt lề cho chương trình của bạn bằng cách gọn gàng và phù hợp sao cho điều đó giúp code của bạn dễ đọc, dễ hiểu.

Bên cạnh regular space character (\u0020), javascript cũng chấp nhận các kí tự liên tiếp như khoảng trắng: tab (\u0009), vertical tab (\u000B), bất kì kí tự dạng Unicode category Zs. Javascript chấp nhận các kí tự tiếp nối như kết thúc 1 dòng: line feed (\u000A), carriage (\u00D), line separotor (\u2028), paragrahp separotor (\u2029). Một carriage return , line feed được xử lí như 1 single line terminator.

Unicode format control character như RIGHT-TO-LEFT MARK (\u200F), LEFT-TO-RIGHT MARK (\u200E),

...... còn tiếp để dịch sau vì hơi khó dịch

**2.2 Comments**

Javascript hỗ trợ 2 kiểu comment :

// viết comment ở đây và /\* viết comment ở đây \*/, những đoạn comment này có thể viết thành nhiều dòng nhưng ko được lồng nhau. Các dòng sau của đoạn code đều là các đoạn comment hợp lệ:

// This is a single-line comment.  
/\* This is also a comment \*/ // and here is another comment.  
/\*  
 \* This is yet another comment.  
 \* It has multiple lines.  
 \*/  
var *a*;

**2.3 Literals(chuỗi ký tự)**

Một chuỗi ký tự là giá trị data xuất hiện trực tiếp ở trong chương trình. Ví dụ:

12 // The number twelve  
1.2 // The number one point two  
"hello world" // A string of text  
'Hi' // Another string  
true // A Boolean value  
false // The other Boolean value  
/javascript/gi // A "regular expression" literal (for pattern matching)  
null // Absence of an object

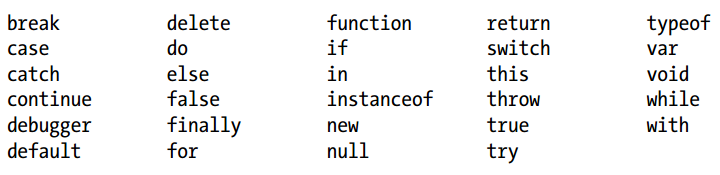
**2.4 Indentifier và Reserved Words (Định danh và từ riêng)**

Identifier đơn giản là 1 tên gọi. Trong Javascript, identifier dùng để gọi tên biến, hàm và cung cấp các hiệu cho các vòng lặp nhất định trong Javascript code. Một JavaScript indentifier phải bắt đầu bằng 1 kí tự, dấu gạch chân, $. Theo sau kí tự có thể là kí tự đơn, kí tự số , dấu gạch dưới, kí tự $, kí tự số học không được phép ở kí đầu tiên, điều này sẽ dễ dàng phân biệt đâu là indentifier và number:

i  
my\_variable\_name  
v13  
\_dummy  
$str

2.4.1 Reserved Words (Từ thay thế)

JavaScript thay thế 1 vài kiểu định dạng như những từ khóa có sẵn trong ngôn ngữ. Bạn không thể sử dụng những từ khóa này như những định danh trong chương trình của mình:



JavaScript cũng thay thế hoàn toàn những từ khóa không được sử dụng ở thời điểm hiện tại, nhưng có thể sẽ được sử dụng ở những phiên bản tiếp theo

Chapter 3. Types, Values, Variables

**3.3 Boolean Values**

Value boolean thường là giá trị của kết quả so sánh.

Boolean value thông thường được sử dụng trong cấu trúc điều khiển javascript, ví dụ như if/else thực hiện 1 hành động nếu value = true và thực hiện hành động khác nếu value = false.