**JavaScript Style Guide**

Contents

1. 소개

1.1 용어 참고

1.2 가이드 노트

2. 소스 파일 기본 사항

2.1 파일 이름

2.2 파일 인코딩 : UTF – 8

2.3 특수 문자

3. 소스 파일 구조

3.1 라이선스 또는 저작권 정보 (있는 경우)

3.2 @fileoverview JSDoc (있는 경우)

3.3 goog.module 문

3.3.3 goog.module 내보내기

3.4 ES 모듈

3.5 goog.setTestOnly

3.6 goog.require 및 goog.requireType 문

3.7 파일의 구현

4. 서식

4.1 교정기

4.2 블록 들여 쓰기 : +2 공백

4.3 진술

4.4 열 제한 : 80

4.5 줄 바꿈

4.6 공백

4.7 그룹화 괄호 : 권장

4.8 댓글

5. 언어 기능

5.1 지역 변수 선언

5.2 배열 리터럴

5.3 객체 리터럴

5.4 클래스

5.5 기능

5.6 문자열 리터럴

5.7 숫자 리터럴

5.8 제어 구조

5.9 this

5.10 평등 검사

5.11 허용되지 않는 기능

6. 명명규칙

6.1 모든 식별자에 공통적인 규칙

6.2 식별자 유형별 규칙

6.3 카멜 케이스

7. JSDoc

1.1 용어 참고

1.2 가이드 노트

8. 정책

8.1 Google 스타일에서 지정하지 않은 문제 : 일관성 유지!

8.2 컴파일러 경고

8.3 지원 중단

8.4 Google 스타일에없는 코드

8.5 지역 스타일 규칙

8.6 생성 된 코드 : 대부분 면제

9. 부록

9.1 JSDoc 태그 참조

9.2 일반적으로 오해되는 스타일 규칙

9.3 스타일 관련 도구

9.4 레거시 플랫폼에 대한 예외

**1. 소개**

이 문서는 JavaScript 프로그래밍 언어의 소스 코드에 대한 코딩 표준을 정의합니다. 다른 프로그래밍 스타일 가이드와 마찬가지로 다루는 문제는 형식화의 미적 문제 뿐 아니라 다른 유형의 규칙이나 코딩 표준에도 적용됩니다. 그러나이 문서는 주로 우리가 보편적으로 따르는 엄격하고 빠른 규칙에 초점을 맞추고 명확하게 시행 할 수없는 조언 (인간 또는 도구)을 제공하지 않습니다.

1.1 용어 참고

이 문서에서 달리 명시되지 않는 한 :

1. 주석 이라는 용어 는 항상 구현 주석을 의미합니다 . Google은 문서 주석 이라는 문구를 사용하지 않고 대신 사람이 읽을 수있는 텍스트와 /\*\* … \*/.

2. 이 스타일 가이드는 must , must not , should , should not , may 라는 문구 를 사용할 때 RFC 2119 용어를 사용합니다 . 조건은 선호 하고 회피 대응 해야 하고 해야하지 , 각각. 명령문 및 선언문은 규범 적이며 must에 해당 합니다 .

문서 전체에 다른 용어 참고 사항 이 가끔 표시됩니다.

1.2 가이드 노트

이 문서의 예제 코드는 **표준**이 **아닙니다** .  즉, 예제는 단순히 스타일이지만 코드를 표현하는 유일한 세련된 방법을 설명하지 못할 수 있습니다.  예제에서 선택한 선택적 형식 지정은 규칙으로 적용되지 않아야합니다.

**2. 소스 파일 기본 사항**

2.1 파일 이름

파일 이름은 모두 소문자 여야하며 밑줄 ( \_) 또는 대시 ( -)를 포함 할 수 있지만 추가 구두점은 없습니다. 프로젝트에서 사용하는 규칙을 따르십시오. 파일 이름의 확장자는 .js

2.2 파일 인코딩 : UTF-8

소스 파일은 **UTF-8** 로 인코딩됩니다.

2.3 특수 문자

2.3.1 공백 문자

줄 종결 자 시퀀스를 제외하고 ASCII 가로 공백 문자 (0x20)는 소스 파일의 모든 위치에 나타나는 유일한 공백 문자입니다. 이것은

- 문자열 리터럴의 다른 모든 공백 문자는 이스케이프됩니다.

- 탭 문자는 들여 쓰기에 사용 되지 않습니다 .

2.3.2 특수 이스케이프 시퀀스

특별한 이스케이프 시퀀스가 있는 모든 문자 ( \', \", \\, \b, \f, \n, \r, \t, \v), 그 시퀀스 (예 : 해당 숫자 탈출보다는 사용 \x0a, \u000a또는 \u{a}). 레거시 8 진 이스케이프는 사용되지 않습니다.

2.3.3 비 ASCII 문자

나머지 비 ASCII 문자의 경우 실제 유니 코드 문자 (예 ∞:) 또는 동등한 16 진수 또는 유니 코드 이스케이프 (예 :) \u221e가 사용됩니다. 이는 코드 를 읽고 이해하기 쉽게 만드는 요소에 따라 다릅니다 .

팁 : 유니 코드 이스케이프 케이스 및 실제 유니 코드 문자가 사용되는 경우에도 설명 주석이 매우 유용 할 수 있습니다.

/ \* 최고 : 댓글이 없어도 완벽하게 선명합니다. \* / const 단위 = 'μs' ;

/ \* 허용 : 그러나 μ는 인쇄 가능한 문자이므로 불필요합니다. \* / const 단위 = '\ u03bcs' ; // 'μs'

/ \* 좋음 : 명확성을 위해 주석과 함께 인쇄 할 수없는 문자에는 이스케이프를 사용합니다. \* / return '\ ufeff' + content ; // 바이트 순서 표시를 앞에 추가합니다.

/ \* 나쁨 : 독자는 이것이 어떤 캐릭터인지 모릅니다. \* / const 단위 = '\ u03bcs' ;

팁 : 일부 프로그램이 비 ASCII 문자를 제대로 처리하지 못할 수 있다는 두려움 때문에 코드의 가독성을 낮추지 마십시오. 이런 일이 발생하면 해당 프로그램이 **중단** 되고 **수정** 해야합니다 .