* 소스파일
* 소스파일의 이름은 알파벳 소문자, 하이픈, 밑줄으로만 작성한다
* 소스파일의 인코딩은 UTF-8으로 작성한다
* 형식
* 중괄호는 하나의 구문만을 포함하고 있더라도 모든 제어문에 사용되어야한다.

\*If문의 경우, 하나의 구문만 포함할 경우, 같은 라인에 작성하고 중괄호를 생략할 수 있다.  
  
\*중괄호의 사용은 **Kernighan and Ritchie Style**에 따른다.

여는 중괄호 전에는 줄을 바꾸지 않는다.  
여는 중괄호 이후에 줄을 바꾼다.  
닫는 중괄호 전에 줄을 바꾼다.  
닫는 중괄호 이후에 줄을 바꾼다. 단, 닫는 중괄호 다음에 else, catch, while, 점, 세미콜론, 닫는 소괄호가 따라올 경우에는 줄을 바꾸지 않는다

\*빈 블록의 경우, **여는 중괄호( { )** 바로 다음에 닫는 중괄호를 사용한다.  
단, **if-else 구조** 혹은 **try-catch-finally 구조**에서의 빈 블록에서는 줄을 바꾼다

\*들여쓰기

* + 새로운 블록이 나타날 때 마다 **두 칸의 공백**(스페이스바)를 통해 들여쓴다.
  + 배열, 객체는 블록과 동일하게 들여쓴다.
  + 클래스

클래스 리터럴은 블록과 동일하게 들여쓴다.  
클래스의 각 메소드와 닫는 중괄호 다음에는 세미콜론을 사용하지 않는다.  
클래스의 상속 키워드로 @extends가 아닌 extends를 사용한다

* 변수
  + - 한 줄에 하나의 변수를 선언한다
    - 지역 변수는 그 변수를 포함하는 블록 시작에서 선언하지 않고, 사용 범위를 최소화하기 위해 사용되는 지점과 가장 가까운 곳에서 선언한다.
    - 변수를 선언할 때는 **const**를 사용한다.단, 변수의 값이 바뀌는 경우 **let**을 사용한다.
    - const 선언문을 먼저 그룹화한 다음에 let 선언문을 그룹화한다
* 객체
* **예약어**를 키로 사용하지 않는다. 대신, 동의어를 사용한다.
* 키는 **큰 따옴표( " )**를 씌운다.  
  단, 큰 따옴표를 씌운 키와 씌우지 않은 키를 한 오브젝트에 동시에 사용하지 않는다.
* 오브젝트를 선언할 때는 Object 생성자가 아닌 리터럴 구문을 사용한다.
* 클래스
* 생성자는 선택적으로 작성한다. 하지만 하위 클래스는 필드를 설정하기 전에 반드시 **super()**를 호출해야한다. 그렇지 않으면 **this**에 접근할 것이다.  
  인터페이스의 경우 메소드가 아닌 속성을 생성자에서 반드시 선언해야 한다.
* 정적 메소드의 사용 보다는 모듈 함수를 더 지향한다.
  + 정적 메소드는 클래스 내에서만 호출되어야 gksek.
  + 동적 인스턴스를 포함하는 변수나 하위 클래스의 생성자 내부에서 호출되어서는 안된다.
  + 하위 클래스 내에 선언되지 않은 정적 메소드는 호출되어서는 안된다.
* 함수
* 함수 내에 또다른 함수를 선언할 수 있다. 함수에 이름이 필요한 경우 **const**를 사용한다.
* 함수식 보다는 함수 선언을 사용한다.
* 함수 이외의 블록(i.e. if, while) 안에서 함수를 선언하지 않는다.
* 함수의 마라미터에 **arguments**를 사용하지 않는다. 이는 오브젝트의 참조를 덮어쓰는 것을 야기한다.
* **arguments**를 사용하는 것 대신에 **Rest 신택스( ... )**를 사용한다.
* 함수의 파라미터를 조작하는 것보다는 **파라미터 기본값**을 사용한다.
* 사이드 이펙트가 발생할 파라미터 기본값의 사용을 지양한다.
* 파라미터 기본값은 가장 뒤쪽에 둔다.
* 함수식을 사용해야만한다면, 화살표 함수(Arrow Function)을 사용하라.
* 화살표 함수에서는 수 바디가 하나의 식으로 구성된 경우, 중괄호와 return문을 생략할 수 있다. 중괄호를 생략하지 않을 경우, **return**문을 포함시켜야 한다.
* 문자열
* 문자열을 선언할 때는 **작은 따옴표( ' )**를 사용한다.  
  문자열 내에 작은 따옴표가 포함될 경우 **템플릿 리터럴( ` ` )**을 사용한다.
* 80글자 이상의 긴 문자열을 여러 줄에 걸쳐 쓰기 위해서는 **템플릿 리터럴** 혹은 **문자열 연결( + )**을 사용한다.
* 동적으로 문자열을 생성할 경우에는 문자열 연결이 아닌 **템플릿 리터럴**을 사용한다.
* 반복문
* 일반적인 for문 보다는 **for-of**를 사용한다.  
  하지만, 가능하다면 **map()**, **reduce()**와 같은 고급 함수를 사용하라.
* 스위치문
  + 다음 case 구문이 실행되어야 한다면 주석으로 이를 남긴다.
  + default문은 항상 마지막에 위치한다
* 등가식
* **==**, **!=** 보다는 **===**, **!==**를 사용하라.
* 조건식은 **ToBoolean 메소드**에 의한 강제 형변환으로 평가되어 아래의 규칙을 따른다.
* 오브젝트는 true로 평가
* undefined는 false로 평가
* null 은 false 로 평가
* 값은 true로 평가. 단, +0, -0, or NaN 의 경우는 false로 평가
* 문자열은 true로 평가. 단, 빈문자( '' )의 경우는 false로 평가

[참조 URL]

* https://bit.ly/38sFwdC