

Global 오브젝트

Global 오브젝트 개요

- 모든 `<script>`를 통해 하나만 존재
 - `new` 연산자로 인스턴스 생성 불가
 - 모든 코드에서 공유
- 이름(Global)은 있지만
 - 오브젝트 실체가 없음
 - 오브젝트를 작성(사용)할 수 없음

Global 오브젝트 함수, 변수

- Global 오브젝트의 함수, 변수를
 - Global 함수, Global 변수라고 부름
- 함수 안에 작성한 것
 - 지역 함수, 로컬 Local 함수라고 부름
 - 지역 변수, 로컬 변수라고 부름
- 전역 객체라고 부르기도 하지만
 - Global은 오브젝트 이름
 - 강좌에서는 Global(글로벌) 오브젝트로 표기

Global 프로퍼티

프로퍼티 리스트

이름	개요
값	
NaN	Not-a-Number
Infinity	무한대 값
undefined	undefined
함수	
isNaN()	NaN 여부. NaN이면 true, 아니면 false 반환
isFinite()	유한대 여부. 유한이면 true, 아니면 false
parseInt()	정수로 변환하여 반환
parseFloat()	실수로 변환하여 반환
eval()	문자열을 JS 코드로 간주하여 실행
encodeURIComponent()	URI 인코딩
encodeURIComponent()	URI 확장 인코딩
decodeURI()	encodeURIComponent 함수의 인코딩 값을 디코딩
decodeURIComponent()	encodeURIComponent 함수의 인코딩 값을 디코딩

Global 프로퍼티

- Global 프로퍼티 종류
 - NaN: Not-a-Number
 - Infinity: 무한대
 - undefined: undefined
- 상수 개념으로 사용
 - 외부에서 프로퍼티 값 변경 불가 [코드 1](#)
- `delete` 연산자로 삭제 불가

Global과 Window 관계

Global과 window 관계

- 글로벌과 window 오브젝트 주체
 - 글로벌 오브젝트는 JS가 주체
 - window 오브젝트는 window가 주체
- 주체는 다르지만,
글로벌 오브젝트의 프로퍼티와 함수가
window 오브젝트에 설정됨
- Global과 window 오브젝트 관계
- Host 오브젝트 개념 활용

정수, 실수 변환

parseInt()

구분	데이터터(값)
파라미터	값
	진수, 디폴트: 10진수
반환	변환한 값

- 값을 정수로 변환하여 반환
 - 값이 123.56이면 123 반환 [코드 1](#)
- 값이 "123px"이면 123 반환
 - 이 용도로도 많이 사용 [코드 2](#)
- 0 또는 빈 문자열을 제외시킴 [코드 3](#)
- 진수를 적용하여 값을 변환 [코드 4](#)

parseFloat()

구분	데이터(값)
파라미터	값
반환	변환한 값

- 값을 실수로 변환하여 반환 [코드 1](#)
 - JS는 기본적으로 실수로 처리하므로
 - 실수로 변환하는 것이 의미가 없지만
 - 문자열의 실수 변환은 의미가 있음
- 지수, 공백 변환 [코드 2](#)

NaN, 유한대 체크 함수

isNaN()

구분	데이터(값)
파라미터	값
반환	true, false

- 값의 NaN 여부 반환
- 숫자 값이 아니면 **true** 반환 [코드 1](#)
- 숫자 값이면 **false** 반환
 - 값이 숫자로 변환되면 숫자로 인식 [코드 2](#)
- **NaN === NaN** 결과는 **false**
 - 설계 실수
 - ES6의 **Object.is()** 사용 [코드 3](#)

isFinite()

구분	데이터터(값)
파라미터	값
반환	true, false

- 값이 Infinity, NaN이면
 - false 반환
 - 아니면 즉, finite(유한)이면 true 반환 [코드 1](#)
- 값이 숫자로 변환되면 숫자로 인식 [코드 2](#)

인코딩, 디코딩

encodeURIComponent()

구분	데이터(값)
파라미터	URI
반환	인코딩 결과

- URI를 인코딩Encoding하여 반환
 - Uniform Resource Identifier
 - 인코딩 제외 문자를 제외하고
"%16진수%16진수" 형태로 변환 [코드 1](#)
- 인코딩 제외 문자
 - 영문자, 숫자
 - # ; / ? : @ & = + \$, - _ . ! ~ * () 따옴표

encodeURIComponent()

구분	데이터(값)
파라미터	URI
반환	인코딩 결과

- URI를 인코딩하여 반환
 - "; / ? : @ & = + \$, "를 인코딩하는 것이 encodeURIComponent()와 다름
 - 인코딩 제외 문자를 제외하고 "%16진수 16진수" 형태로 변환
- 인코딩 제외 문자
 - 영문자, 숫자
 - # - _ . ! ~ * () 따옴표

decodeURI()

구분	데이터(값)
파라미터	인코딩 문자열
반환	디코딩 결과

- 인코딩을 디코딩^{Decoding}하여 반환
- 파라미터에 **encodeURIComponent()**로 인코딩한 문자열 작성 [코드 1](#)

decodeURIComponent()

구분	데이터(값)
파라미터	인코딩 문자열
반환	디코딩 결과

- 인코딩을 디코딩하여 반환
- 파라미터에 `encodeURIComponent()`로 인코딩한 문자열 작성

eval() 함수

eval()

구분	데이터(값)
파라미터	JS 코드
반환	JS 코드를 실행하여 반환한 것

- 파라미터의 문자열을
 - JS 코드로 간주하여 실행 [코드 1](#)
 - 실행 결과를 반환 값으로 사용
값을 반환하지 않으면 `undefined` 반환
- `eval()` 함수 실행 방법
- 보안에 문제가 있다고 알려져 있음
 - 사용 비권장