

# Javascript 내장기능 함수 정리

🕒 작성일시	@2022년 2월 26일 오후 7:11
📌 강의 번호	
📌 유형	
📎 자료	
☑ 복습	<input type="checkbox"/>
☰ 속성	

기능 이름	태그 이름	태그 예시	기능 설명
isNaN	isNaN	console.log(isNaN(파라미터));	숫자가 아니면 true, 숫자이거나 변환 가능한 형식은 false ex) NaN, undefined, {}, '문자열', '123ABC' ⇒ true true, null, 37, '37', ' '(빈문자열은 0으로 변환) ⇒ false
parseFloat	parseFloat	console.log(parseFloat(값));	주어진 값에서 변환한 실수(소수점)을 리턴 ex) 3.14, '3.14', '314e-2', '0.0314E+2' e-2는 왼쪽으로 소수점 2칸 이동, E+2는 오른쪽으로 소수점 2칸 이동
parseInt	parseInt	console.log(parseInt(value, int));	- 첫 번째 파라미터는 파라미터를 10진 점수 값으로 변환 - 두 번째 파라미터는 value가 어떤 진법인지 알려주는 값(기본값=10)
encodeURIComponent	encodeURIComponent	const 변수 = encodeURIComponent(상위변수);	- 주어진 문자열을 URL에 포함 시키기에 적절한 형태로 변환(=인코딩)하는 처리(공백은 %20으로 인코딩) - 인코딩하지 않는 문자 ⇒ A-Z a-z 0-9 ; , / ? : @ & = + \$ - _ . ! ~ * ' ( ) #
decodeURI	decodeURI	console.log(decodeURI(상위변수));	encodeURIComponent로 변환된 문자열을 원래 형태로 인코딩
encodeURIComponent	encodeURIComponent	const 변수 = encodeURIComponent(상위변수)	- 알파벳과 숫자 및 비예약 표식을 제외한 모든 글자를 URL에 포함시킬 수 있는 문자열로 인코딩한다. ⇒ URL에 사용해도 문제가 없는 특수문자를 제외하고 모든 글자를 반환
decodeURIComponent	decodeURIComponent	console.log(decodeURIComponent(상위변수));	decodeURIComponent로 변환된 문자열을 원래 형태로 인코딩
setTimeout	setTimeout	setTimeout() =>{ ..... } ,시간(1/1000초));	-지정된 함수를 전달된 시간 후에 실행되도록 예약(딜레이 가능) - setTimeout() 이후의 처리 로직들은 func의 실행 여부와 상관없이 즉시 실행된다.
setInterval	setInterval	setInterval() =>{ ..... } ,시간(1/1000초));	- 지정된 함수를 전달된 시간마다 한 번씩 호출(타이머기능) - 타이머를 종료시킬 수 있는 timerid를 반환한다.
string	charAt	const 변수이름 = 상위변수이름.charAt(위치값);	피라미터로 선정된 위치의 글자를 리턴 ⇒ 위치는 0부터 카운트
string	msg	console.log("몇 번째 글자 : " + msg[위치값]);	모든 문자열은 그 자체로 배열처럼 취급될 수 있다.
string	indexOf	const 변수이름 = 상위변수이름.indexOf("");	- 파라미터로 전달된 내용일 처음 나타내는 위치를 리턴 - 찾지 못할 경우 -1을 반환함 - 단어나 문장으로 검색할 경우 일치하는 내용이 시작되는 첫 글자의 위치를 반환 - indexOf가 두 개인 경우, 두 번째 숫자 값은 첫 번째의 파라미터의 글자를 찾기 시작하는 위치는 의미 ⇒ indexOf("찾을 글자", 첫 번째 변수 이름 +1)
string	lastIndexOf	const 변수이름 = 상위변수이름.lasstIndexOf("");	- 파라미터로 전달된 글자가 마지막으로 나타나는 위치를 리턴 - 위치를 문자열의 끝부터 세는 것이 아니라 문자열을 처음부터 센다. - 찾지 못할 경우 -1을 반환.
string	substring	const 변수이름 = 상위변수이름.substring(시작위치, 끝위치);	- 지정한 시작 위치와 끝 위치 자르기 - 지정된 끝 위치의 직전 글자까지만 잘라낸다. - 두 번째 파라미터가 없을 경우 지정된 위치부터 끝까지 자른다.
string	toUpperCase	const 변수이름 = 상위변수이름.toUpperCase();	- 모든 글자 대문자로 변환
string	toLowerCase	const 변수이름 = 상위변수이름.toLowerCase();	- 모든 글자 소문자로 변환
string	trim	const 변수이름 = 상위변수이름.trim();	- 문자열 앞, 뒤 공백 지우기

기능 이름	태그 이름	태그 예시	기능 설명
string	split	const 변수이름 = 상위변수이름.split("구분자");	- 문자열 안에 포함된 구분자를 기준으로 문자열을 잘라 배열로 변환 ex) const txt = "HTML,CSS,Javascript"; const arr = txt.split(",");
string	replace	const 변수이름 = 상위변수이름.replace(파라미터1, 파라미터2);	- 첫 번째 파라미터의 내용을 두 번째 파라미터로 변경한 결과 반환 - 첫 번째 파라미터와 일치하는 내용이 둘 이상 존재할 경우 첫 번째 항목만 변경한다.
string	replaceAll	console.log(변수이름.replaceAll(파라미터1, 파라미터2));	- 파라미터1를 파라미터2로 변경하여 반환
Math	Math.max Math.min	var 변수 = Math.max(값1, 값2) var 변수 = Math.min(값1, 값2)	- 주어진 파라미터 값 중 최대값, 최소값 구하기
Math	Math.round Math.ceil Math.floor	console.log("소수점 반올림:" + Math.round(변수이름)); console.log("소수점 올림:" + Math.ceil(변수이름)); console.log("소수점 내림:" + Math.floor(변수이름));	- 주어진 변수의 값을 소수점을 기준으로 반올림, 올림, 내림 - 미래의 시간, 날짜를 구할 때는 올림 사용 - 과거의 시간, 날짜를 구할 때는 내림 사용
Math	abs	console.log("절대값:" + Math.abs(변수이름));	- 주어진 값이 마이너스(-)일 경우 플러스(+)로 변환
Math	random	범위 난수 발생 console.log("난수:" + Math.random()); 두 수 사이의 난수 리턴 function random(값, 값2) { return parseInt(Math.random() *(값2-값1+1)) + 값1; }	- random을 실행할 때마다 출력 값이 달라짐 - 기본 난수의 값은 0~1 사이의 소수점 값임으로 그 이상의 값은 반복문을 통해 숫자의 범위를 지정 해줘야함
Date	getFullYear getMonth getDate getDay	const yy = 변수이름.getFullYear(); const mm = 변수이름.getMonth() +1; const dd = 변수이름.getDate(); const i = 변수이름.getDay();	- 년, 월, 일, 시간, 분, 초를 리턴받기 - 월은 0이 1월 11일이 12월을 의미한다. - 0=일요일 6=토요일의 값이 리턴된다.
Date	toString	console.log(date1.toString());	- 날짜 부분만 나타내는 문자열 반환 - 날짜 부부만 나타내는, 사람이 읽을 수 있는 문자열을 반환
Date	toISOString	console.log(date1.toISOString());	- Date를 나타내는 문자열을 ISO 8691 확장 형식에 맞춰 반환
Date	toLocaleString	console.log(변수이름.toLocaleString());	- 형식 문자열을 사용해서 Date를 나타내는 문자열을 생성 - 특정 날짜를 의미하는 객체 생성을 할 수 (시각이 없으면 자동으로 설정됨) ex) console.log(변수이름.toLocaleString('ko-KR'));
Date	toLocaleDateString	console.log(변수이름.toLocaleDateString());	- Date의 날짜 부분을 나타내는 문자열을 시스템에 설정된 현재 지역의 형식으로 반환
Date	toLocaleTimeString	console.log(변수이름.toLocaleTimeString());	- Date의 시간 부분을 나타내는 문자열을 시스템에 설정된 현재 지역의 형식으로 반환
Date	setYear setMonth setDate setHour setMinutes setSeconds	변수이름.setYear(); 변수이름.setMonth(); 변수이름.setDate(); 변수이름.setHour(); 변수이름.setMinutes(); 변수이름.setSeconds();	- 이미 생성된 날짜 객체의 성분을 변경
Number	Number	var 변수이름 = Number(값);	- Number() 함수를 통해 반환받는 값은 객체가 아닌 일반숫자 - Number() 함수는 parseFloat, parseInt와 비슷한 기능.
Number	Number.MAX_VALUE	console.log(Number.MAX_VALUE);	- 표현 가능한 가장 큰 함수.
Number	Number.MIN_VALUE	console.log(Number.MIN_VALUE);	- 표현 가능한 가장 작은 양수. - 0보다 크지만 0에 가장 가까운 양수.
Number	isInteger	console.log(Number.isInteger(값));	- 주어진 값이 정수 인지 확인
Regex	test(입력받은변수값)	const 변수명 = /^정규표현식*\$/ 변수명.test(입력받은변수값)	- 문자열이 특정 조건을 충족하는지 검사하거나 특정 패턴 문자열 검색
Array	push	변수이름.push(값);	- 배열의 맨 끝에 요소 추가(파라미터 수 제한 없음)
Array	pop	변수이름.pop(값);	- 배열의 마지막 요소를 리턴하고 제거
Array	shift	const 변수이름 = 변수이름.shift(값);	- 맨 앞 요소를 리턴하고 제거
Array	unshift	const 변수이름 = 변수이름.unshift(값);	- 맨 앞에 요소를 추가 ( 파라미터 수 제한 없음)

기능 이름	태그 이름	태그 예시	기능 설명
Array	splice	const 변수이름 = 변수이름.splice(자를 위치,갯수,추가할값);	- 자를 위치부터 자를 개수만큼 잘라서 반환하고 원본에서는 제거 - 추가할 값을 넣을 경우 자른 위치에 다른 요소가 배치 됨 (기존 원소는 뒤로 밀리고 제거된 수와 추가될 수가 다를 경우 배열이 축소, 확장됨)
Array	concat	const 변수이름 = a.concat(b,c);	- a에 b와 c를 결합한 새로운 배열을 반환
Array	indexOf(item, from)	변수이름.indexOf(item, from)	- index from부터 시작해 item(요소)를 탐색 - 요소를 발견하면 해당 요소의 인덱스를 반환한다.
Array	includes(item, from)	변수이름.includes(item, from)	- includes from부터 시작해 item이 있는지를 검색 - 요소가 배열 내 존재하는지 여부만 확인해줌 - includes는 NaN도 제대로 처리한다
Array	find	const 변수이름 = 상위변수이름.find(function(v){ ...조건판별을 위한 if 문... });	- 배열 검색 - 주어진 판별 함수를 만족하는 첫번째 값을 반환한다. - 못찾으면 undefinde를 반환한다.
Array	filter	const 변수이름 = 상위변수이름.filter(function(v, i, arr){ ...조건판별을 위한 if문... });	- 배열 전체 검색 - 주어진 판별 함수를 만족하는 값을 반환한다. - 배열의 모든 원소를 탐색하기 전까지는 종료되지 않는다
Array	sort	변수이름.sort(function(a,b){ ... 정렬을 위한 if 문... });	- 배열 정렬 - sort 함수도 정렬 조건을 콜백함수로 처리함
Array	reverse	변수이름.reverse();	- 지정된 변수 내에 있는 값들을 역순으로 배치