String 오브젝트

String 오브젝트

- "ABC"처럼 문자 처리를 위한 오브젝트
- 즉, String 오브젝트에
  - 문자 처리를 위한
  - 함수와 프로퍼티가 포함되어 있으며
  - 함수를 호출하여 문자 처리를 하게 됩니다.

문자열 연결 방법

- 한 줄에서 연결
  - var book = "12" + "AB" + "가나";
- 줄을 분리하여 연결
  - +로 문자열 연결 코드 1
  - 역슬래시(\)로 문자열 연결 코드 2

프로퍼티 리스트

이름	개요
new String()	인스턴스 생성
String 함수	
String()	문자열로 변환하여 반환
fromCharCode()	유니코드를 문자열로 변환하여 반환
String 프로퍼티	
length	문자열의 문자 수 반환
String.prototype	
constructor	생성자
valueOf()	프리미티브 값 반환
toString()	문자열로 변환
charAt()	인덱스 번째 문자 반환
indexOf()	일치하는 문자열 중에서 가장 작은 인덱스 반환
lastIndexOf()	일치하는 문자열 중에서 가장 큰 인덱스 반환

계속…

프로퍼티 리스트

이름	개요
String.prototype	
concat()	문자열 연결
toLowerCase()	영문 소문자로 변환
toUpperCase()	영문 대문자로 변환
trim()	문자열 앞뒤의 화이트 스페이스 삭제
substring()	시작에서 끝 직전까지 값 반환
substr()	시작 위치부터 지정한 문자 수 반환
slice()	시작에서 끝 직전까지 값 반환. substring()과 차이 있음
match()	매치 결과 반환
replace()	매치 결과를 지정한 값으로 대체
search()	검색된 첫 번째 인덱스 반환
split()	구분자로 분리하여 반환
charCodeAt()	인덱스 번째 문자를 유니코드로 반환
localeCompare()	값의 위치를 1, 0, -1로 반환

문자열로 변환

String()

구분	데이터(값)
파라미터	변환 대상opt
반환	변환한 값

- 파라미터 값을 String 타입으로 변환하여 반환
  - 값을 작성하지 않으면 빈문자열 반환 코드 1
- 가독성
  - ("" + 123)도 숫자가 String 타입이 되지만
  - String(123) 형태가 가독성이 높습니다.

new String()

구분	데이터(값)
파라미터	값opt
반환	생성한 String 인스턴스

- String 인스턴스를 생성하여 반환 코드 1
- 파라미터 값을 String 타입으로 변환
  - 파라미터 값이 프리미티브 값이 됩니다.

valueOf()

구분	데이터(값)
data	String 인스턴스, 문자
파라미터	사용하지 않음
반환	프리미티브 값

- String 인스턴스의 프리미티브 값 반환 코드 1

length 프로퍼티

length 프로퍼티

- 문자 수 반환 코드 1
- length 프로퍼티 활용 코드 2
- length 값이 반환되는 논리

화이트 스페이스 삭제

trim()

구분	데이터(값)
data	삭제 대상
파라미터	사용하지 않음
반환	삭제한 결과

- 문자열 앞뒤의 화이트 스페이스 삭제 코드 1
- 메소드 체인Method chain

함수 호출 구조

toString()

구분	데이터(값)
data	문자열, String 인스턴스
파라미터	사용하지 않음
반환	변환한 값

- data 위치의 값을 String 타입으로 변환 코드 1

toString()

- "123".toString();
  - String 타입을 String 타입으로 변환
  - 의미가 없다?
- toString() 함수가 필요한 이유
- __proto__:
  toString();
  __proto__
  toString();
- 그래서 대부분의 빌트인 오브젝트에 toString()과 valueOf()가 있습니다.

JS 함수 호출 구조

- 우선, 데이터 타입으로
  - 오브젝트를 결정하고
  - 오브젝트의 함수를 호출합니다. 코드 1
- toString(123)
  - 123을 파라미터에 작성 코드 2

인덱스로 문자열 처리

charAt()

구분	데이터(값)
data	반환 대상
파라미터	반환 기준 인덱스(Index)
반환	인덱스 번째 문자

- 인덱스의 문자를 반환 코드 1
- 문자열 길이보다 인덱스가 크면
  - 빈 문자열 반환 코드 2
- 일반적으로 존재하지 않으면 undefined를 반환 코드 3

indexOf()

구분	데이터(값)
data	검색 대상
파라미터	검색할 문자열
	검색 시작 위치, 디폴트: 0
반환	인덱스

- data 위치의 문자열에서
  파라미터의 문자와 같은
  첫 번째 인덱스를 반환
- 검색 기준
  - 왼쪽에서 오른쪽으로 검색 코드 1
  - 두 번째 파라미터를 작성하면
    작성한 인덱스부터 검색 코드 2
  - 같은 문자가 없으면 **-1** 반환 코드 3 코드 4

lastIndexOf()

구분	데이터(값)
data	검색 대상
파라미터	검색할 문자열
	검색 시작 위치, 디폴트: 0
반환	인덱스

- data 위치의 문자열에서
  파라미터의 문자와 같은 인덱스를 반환
  단, 뒤에서 앞으로 검색 코드 1
- 검색 기준
  - 두 번째 파라미터를 작성하면
    작성한 인덱스부터 검색 코드 2
  - 같은 문자가 없으면 **-1** 반환

[코딩 시간]

- 요구 사항
  - indexOf()와 lastIndexOf()를 통합하여
    발생 가능한 케이스를 기술하고
    이에 맞는 코드를 작성하세요.
- 목적
  - 코드 작성의 사고
- 작성 예
  - 두 번째 파라미터에 음수를 작성하면 -1을 반환한다.
  - var value = "1234512345"
    console.log(value.lastIndexOf(3, -2));

문자열 연결, 대소문자 변환

concat()

구분	데이터(값)
data	연결 시작 값, String 인스턴스
파라미터	연결 대상opt, 다수 작성 가능
반환	연결한 결과

- data 위치의 값에
  - 파라미터 값을 작성 순서로
  - 연결하여 문자열로 반환 코드 1
- String 인스턴스를 작성하면
  - 프리미티브 값을 연결

toLowerCase()

구분	데이터(값)
data	변환 대상
파라미터	사용하지 않음
반환	변환 결과

- 영문 대문자를 소문자로 변환

toUpperCase()

구분	데이터(값)
data	변환 대상
파라미터	사용하지 않음
반환	변환 결과

- 영문 소문자를 대문자로 변환 코드 1

문자열 추출

substring()

구분	데이터(값)
data	반환 대상
파라미터	시작 인덱스
	끝 인덱스
반환	결과

- 파라미터의 시작 인덱스부터
  끝 인덱스 직전까지 반환 코드 1
- 두 번째 파라미터를 작성하지 않으면
  반환 대상의 끝까지 반환 코드 2
- 다양한 추출 조건 작성 코드 3 코드 4

substr()

구분	데이터(값)
data	반환 대상
파라미터	시작 인덱스
	반환할 문자 수
반환	결과

- 파라미터의 시작 인덱스부터
  지정한 문자 수를 반환 코드 1
- 첫 번째 파라미터 코드 2
  - 값이 음수이면 length에서
    파라미터 값을 더해 시작 인덱스로 사용
- 두 번째 파라미터를 작성하지 않으면
  양수 무한대로 간주 코드 3

slice()

구분	데이터(값)
data	반환 대상
파라미터	시작 인덱스
	끝 인덱스
반환	결과

- 파라미터의 시작 인덱스부터
  끝 인덱스 직전까지 반환 코드 1
- 첫 번째 파라미터 코드 2
  - 값을 작성하지 않거나 NaN이면 0으로 간주
- 두 번째 파라미터 코드 3
  - 작성하지 않으면 length 사용
  - 값이 음수이면 length에 더해 사용 코드 4

정규 표현식 사용 함수

match()

구분	데이터(값)
data	매치 대상
파라미터	정규표현식, 문자열
반환	[매치 결과]

- 매치 결과를 배열로 반환
  - 매치 대상에 정규 표현식의 패턴을 적용하여 매치하고 매치 결과를 반환 코드 1
  - 문자열 작성 가능,
    엔진이 정규 표현식으로 변환하여 매치
- 정규 표현식
  - 문자열을 패턴으로 매치
  - 패턴(pattern) 형태: ^, $, *, + 등

replace()

구분	데이터(값)
data	치환 대상
파라미터	정규 표현식, 문자열
	대체할 값, 함수
반환	치환 결과

- 매치 결과를
  파라미터에 작성한 값으로 대체하여 반환
- 두 번째 파라미터에 함수를 작성하면
  먼저 함수를 실행하고
  함수에서 반환한 값으로 대체 코드 1

search()

구분	데이터(값)
data	검색 대상
파라미터	정규 표현식, 문자열
반환	매치된 인덱스

- 검색된 첫 번째 인덱스 반환
  - 매치되지 않으면 **-1** 반환 코드 1

split()

구분	데이터(값)
data	분리 대상
파라미터	분리자: 정규 표현식, 문자열
	반환 수
반환	결과

- 분리 대상을 분리자로 분리하여 배열로 반환 코드 1
- 분리자를 작성하지 않은 경우 코드 2
- 두 번째 파라미터에 반환 수를 작성 코드 3

Unicode 관련 함수

charCodeAt()

구분	데이터(값)
data	반환 대상
파라미터	반환 기준 인덱스(Index)
반환	인덱스 번째 문자

- 인덱스 번째의 문자를
  유니코드의 코드 포인트 값을 반환
- 인덱스가 문자열 길이보다 크면
  NaN 반환 코드 1

fromCharCode()

구분	데이터(값)
data	String 오브젝트
파라미터	유니코드, 다수 작성 가능
반환	변환한 문자

- 파라미터의 유니코드를
  문자열로 변환하고 연결하여 반환
  - 작성하지 않으면 빈 문자열 반환
- 작성 방법
  - data 위치에 String 오브젝트 작성
    변환 대상 값을 작성하지 않음
  - String.fromCharCode() 형태 코드 1

localeCompare()

구분	데이터(값)
data	비교 대상
파라미터	비교할 값
반환	1(앞), 0(같음), -1(뒤)

- 값을 비교하여
  위치를 나타내는 값으로 반환
- 위치 값: 1(앞), 0(같음), -1(뒤)
- Unicode 사전 순으로 비교 코드 1