

삼육대학교 SW융합교육원 김진호

자바스크립트 기초교육

11-6. String

11-6-1. String

11-6-2. String method

❖ 11-6. String

표준 빌트인 객체인 String은 원시 타입인 문자열을 다룰 때 유용한 프로퍼티와 메서드를 제공한다.

11-6-1. String

```
// new 연산자와 함께 호출하여 String 인스턴스를 생성
const obj = new String();
console.log(obj);           // 인수 전달하지 않으면 빈 문자열을 할당한 객체 생성

const obj2 = new String('홍길동');
console.log(obj2);          // 인수로 문자열 전달 시 전달 받은 문자열 할당

// length 프로퍼티(문자열의 문자 개수)
// 인덱스를 나타내는 숫자 형식의 문자열 - 프로퍼티 키, 각 문자 - 프로퍼티 값
console.log(obj2.length);
console.log(obj2[0]);

// 단, 문자열은 원시 값이므로 변경 불가
obj2[0] = '김';             // 에러는 발생하지 않음
console.log(obj2);

// 문자열이 아닌 값을 인수로 전달했을 경우 문자열로 강제 변환
const obj3 = new String(100);
const obj4 = new String(null);
console.log(obj3[0]);
console.log(obj4[0]);
```

❖ 11-6. String

11-6-2. String method

- 문자열은 변경 불가능한 원시 값이기 때문에 String 래퍼 객체도 읽기 전용 객체로 제공된다.
- String 객체의 모든 메서드는 String 래퍼 객체를 직접 변경할 수 없고, 언제나 새로운 문자열을 생성하여 반환한다.

```
const obj = new String('홍길동');  
console.log(Object.getOwnPropertyDescriptors(obj)); // writable : false
```

❖ 11-6. String

11-6-2. String method

String.prototype.indexOf

- 전달한 값과 일치하는 첫 번째 인덱스를 반환

```
const str = 'JavaScript';  
// 문자열에서 a 검색하여 첫번째 인덱스 반환  
console.log(str.indexOf('a')); // 1  
// 검색에 실패하면 -1 반환  
console.log(str.indexOf('b')); // -1  
// 검색 시작 인덱스 지정  
console.log(str.indexOf('a', 2)); // 3  
// 특정 문자열 존재 유무 확인에 사용  
if(str.indexOf('a') !== -1)  
    console.log('a가 있다');
```

❖ 11-6. String

11-6-2. String method

String.prototype.includes

- 하나의 문자열이 다른 문자열에 포함되어 있는지를 판별하고, 결과를 true 또는 false 로 반환

```
const str = 'JavaScript';  
// 문자열에서 a 검색하여 포함 여부 반환  
console.log(str.includes('a')); // true  
// 검색에 실패하면 false 반환  
console.log(str.includes('b')); // false  
// 검색 시작 인덱스 지정  
console.log(str.includes('a', 2)); // true  
// 특정 문자열 존재 유무 확인에 사용  
if(str.includes('a'))  
    console.log('a가 있다');
```

❖ 11-6. String

11-6-2. String method

String.prototype.search

- 인수로 전달 받은 정규 표현식과 매치하는 문자열을 검색하여 일치하는 문자열 인덱스를 반환

```
const str = 'JavaScript';  
console.log(str.search(/a/));    // 1  
console.log(str.search(/b/));    // -1 (검색 실패)
```

❖ 11-6. String

11-6-2. String method

String.prototype.startsWith

- 어떤 문자열이 특정 문자로 시작하는지 확인하여 결과를 true 혹은 false로 반환

String.prototype.endsWith

- 어떤 문자열이 특정 문자로 끝나는지 확인하여 결과를 true 혹은 false로 반환

```
const str = 'JavaScript';  
// 'Ja'로 시작하는지  
console.log(str.startsWith('Ja'));           // true  
// 검색 시작 인덱스 지정  
console.log(str.startsWith('va', 2));         // true  
// 'pt'로 끝나는지  
console.log(str.endsWith('pt'));              // true  
// 기준 인덱스 지정  
console.log(str.endsWith('va', 4));           // true
```


❖ 11-6. String

11-6-2. String method

String.prototype.charAt

- 문자열에서 특정 인덱스에 위치하는 유니코드 단일문자를 반환

```
const str = 'JavaScript';  
for(let i = 0; i < str.length; i++)  
    console.log(str.charAt(i));
```

❖ 11-6. String

11-6-2. String method

String.prototype.substring

- string 객체의 시작 인덱스로부터 종료 인덱스 전까지 문자열의 부분 문자열을 반환

```
const str = 'JavaScript';  
// 두번째 인수의 인덱스 위치 바로 이전 문자까지  
console.log(str.substring(1,4));           // "ava"  
// 두번째 인수 생략 시 문자열 끝까지  
console.log(str.substring(1));             // "avaScript"  
// 인수 교환하여 기능  
console.log(str.substring(4,1));           // "ava"  
// 음수는 0으로 취급  
console.log(str.substring(-1));            // "JavaScript"  
// length보다 크면 length로 취급  
console.log(str.substring(1,20));         // "avaScript"
```

❖ 11-6. String

11-6-2. String method

String.prototype.slice

- 문자열의 일부를 추출하면서 새로운 문자열을 반환
- substring과 동일하게 동작하지만 음수인 인수를 전달하면 가장 뒤에서부터 시작하여 잘라내 반환

```
const str = 'JavaScript';  
// substring 동일  
console.log(str.slice(1,4));           // "ava"  
// substring 동일  
console.log(str.slice(1));             // "avaScript"  
// 인수 교환하여 기능하지 않음  
console.log(str.slice(4,1));           // ""  
// 음수는 뒤에서부터  
console.log(str.slice(-1));            // "t"  
// substring 동일  
console.log(str.slice(1,20));
```

❖ 11-6. String

11-6-2. String method

String.prototype.toUpperCase

- 문자열을 대문자로 변환해 반환

String.prototype.toLowerCase

- 문자열을 소문자로 변환해 반환

```
const str = 'JavaScript';  
console.log(str.toUpperCase()); // "JAVASCRIPT"  
console.log(str.toLowerCase()); // "javascript"
```

String.prototype.trim

- 문자열 앞뒤 공백 문자 제거 후 반환

```
const str2 = '   JavaScript   ';  
console.log(str2.trim()); // "JavaScript"
```

❖ 11-6. String

11-6-2. String method

String.prototype.repeat

- 전달받은 정수만큼 반복해 연결한 새로운 문자열을 반환

```
const str = 'JavaScript';
console.log(str.repeat());           // ""
console.log(str.repeat(0));          // ""
console.log(str.repeat(1));          // "JavaScript"
console.log(str.repeat(2));          // "JavaScriptJavaScript"
// 실수는 정수로 처리
console.log(str.repeat(2.5));         // "JavaScriptJavaScript"
// 음수는 예러
// console.log(str.repeat(-1));      // RangeError: Invalid count value
```

❖ 11-6. String

11-6-2. String method

String.prototype.replace

- 첫번째 인수로 전달 받은 문자열 또는 정규표현식을 검색하여 두번째 인수로 전달한 문자열로 치환한 문자열 반환

```
const str = 'JavaScript';  
console.log(str.replace('Java', 'Type')); // TypeScript  
// 검색 된 문자열이 여러 존재할 경우 첫번째로 검색된 문자열만 치환  
console.log(str.replace('a', 'b')); // JbvaScript  
// 첫번째 인수로 정규표현식 전달  
console.log(str.replace(/j/i, 'Z')); // ZavaScript
```

❖ 11-6. String

11-6-2. String method

String.prototype.split

- 첫번째 인수로 전달한 문자열 또는 정규식을 검색하여 문자열을 구분한 후 분리 된 각 문자열로 이루어진 배열 반환

```
const str3 = 'Hello, Everyone! Nice to see you again.';
console.log(str3.split(' '));      // 공백을 구분하여 배열로 반환
console.log(str3.split(''));       // 인수로 빈 문자열을 전달하면
    각 문자를 모두 분리
console.log(str3.split());         // 인수를 생략하면 문자열 전체를
    단일 요소로 하는 배열 반환
console.log(str3.split(' ', 5));   // 두번째 인수로 배열의 길이 지정
```