

8강

# 코어객체( String )





## JavaScript

## ➤ 문자열 리터럴 (기본 타입, primitive)

- 문자열 리터럴은 원시(primitive) 타입으로 저장된다.
- 메모리 사용도 효율적이고, 대부분의 경우 이 방식만으로 충분하다.

```
let str1 = "hello";  
  
console.log(typeof str1); // "string"
```

## ➤ String 객체 (래퍼 객체, object)

- new String()을 쓰면 객체(object) 가 만들어진다.
- 이는 특별한 경우(예: 객체로써 속성을 붙이고 싶을 때) 아니면 거의 사용하지 않는다.

```
let str2 = new String("hello");  
  
console.log(typeof str2); // "object"
```



## JavaScript

## ➤ 문자열 리터럴 (기본 타입, primitive) 이 String 객체 메서드를 사용 할 수 있는 이유

- 자바스크립트에서 "hello".toUpperCase() 같은 코드가 동작하는 이유는,  
**원시 타입(primitive string)** 이 **잠깐 래퍼 객체(String 객체)**로 변환되기 때문이다.

## ➤ 동작 원리

- "hello" 는 원시 타입 string
- "hello".toUpperCase() 를 실행하면,
- 자바스크립트 엔진이 내부적으로 이렇게 처리한다:

```
let temp = new String("hello"); // 임시로 String 객체 생성  
let result = temp.toUpperCase(); // 메서드 실행  
temp = null; // 임시 객체 버림 (GC 처리)
```



## JavaScript

## ➤ String 란 ?

- String 객체는 원시 타입인 문자열을 다룰 때 유용한 프로퍼티와 메소드를 제공하는 레퍼(wrapper) 객체이다.
- 변수 또는 객체 프로퍼티가 문자열을 값으로 가지고 있다면 String 객체의 별도 생성없이 String 객체의 프로퍼티와 메소드를 사용할 수 있다.

## ➤ String Property

## String.length

- 문자열 내의 문자 갯수를 반환한다.
- String 객체는 length 프로퍼티를 소유하고 있으므로 유사 배열 객체이다.

```
const str1 = 'Hello';
console.log(str1.length); // 5
```



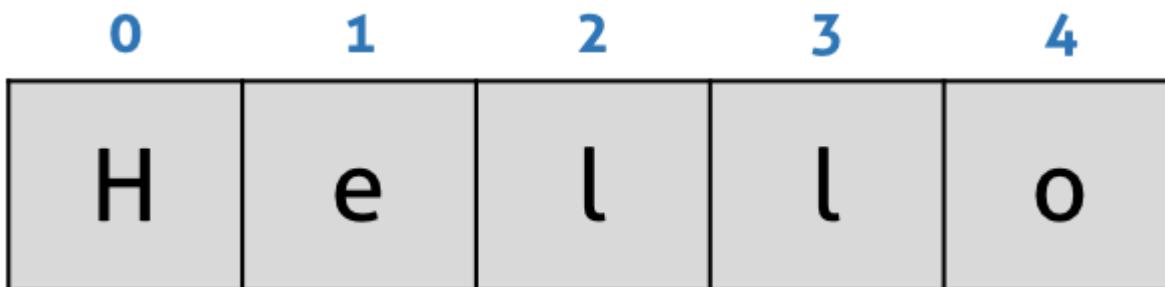
## JavaScript

## ➤ String Method

- String 객체의 모든 메소드는 언제나 새로운 문자열을 반환한다. 문자열은 **변경 불가능(immutable)**한 원시 값이기 때문이다.

## ➤ String.prototype.charAt(number) : string

- 인수로 전달한 index를 사용하여 **index에 해당하는 위치의 문자를 반환**한다.
- index는 0 ~ (문자열 길이 - 1) 사이의 정수이다. 지정한 index가 문자열의 범위(0 ~ (문자열 길이 - 1))를 벗어난 경우 빈 문자열을 반환한다.



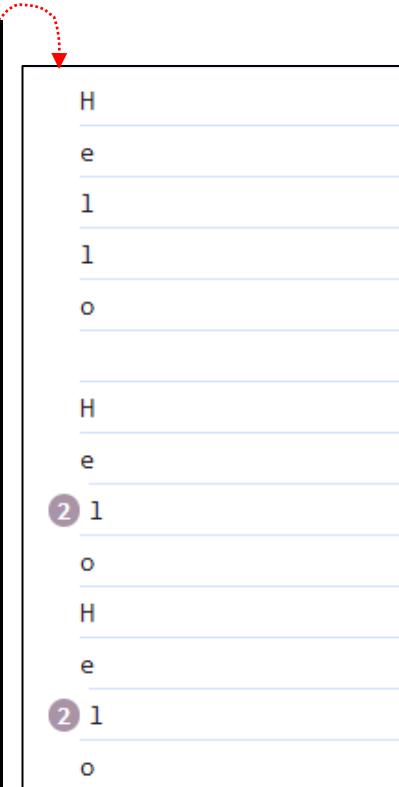


## JavaScript

```
const str = 'Hello';
console.log(str.charAt(0)); //H
console.log(str.charAt(1)); //e
console.log(str.charAt(2)); //l
console.log(str.charAt(3)); //l
console.log(str.charAt(4)); //o

//지정한 index 범위 (0~(문자열길이-1)) 벗어난 경우 빈 문자열 반환한다.
console.log(str.charAt(5)); // ''

//문자열 순회, 문자열은 length 프로퍼티를 갖는다.
for(let i=0; i<str.length; i++){
    console.log(str.charAt(i));
}
// String 객체는 유사 배열 객체이므로 배열과 유사하게 접근할 수 있다.
for(let i=0;i<str.length; i++){
    console.log(str[i]);
}
```





## JavaScript

## 프로토타입 메소드

## 설명

indexOf()	String 인스턴스에서 특정 <u>문자나 문자열</u> 이 처음으로 등장하는 위치의 인덱스를 반환함.
<u>lastIndexOf()</u>	String 인스턴스에서 특정 <u>문자나 문자열</u> 이 마지막으로 등장하는 위치의 인덱스를 반환함.
charAt()	String 인스턴스에서 전달받은 인덱스에 위치한 문자를 반환함.
charCodeAt()	String 인스턴스에서 전달받은 인덱스에 위치한 문자의 UTF-16 코드를 반환함. (0 ~ 65535)
charCodeAt()	String 인스턴스에서 전달받은 인덱스에 위치한 문자의 유니코드 코드 포인트(unicode code point)를 반환함.
slice()	String 인스턴스에서 전달받은 시작 인덱스부터 종료 인덱스 바로 앞까지의 문자열을 추출한 새 문자열을 반환함.
substring()	String 인스턴스에서 전달받은 시작 인덱스부터 종료 인덱스 바로 앞까지의 문자열을 추출한 새 문자열을 반환함.
substr()	String 인스턴스에서 전달받은 시작 인덱스부터 길이만큼의 문자열을 추출한 새로운 문자열을 반환함.
split()	String 인스턴스에서 구분자(separator)를 기준으로 나눈 후, 나뉜 문자열을 하나의 배열로 반환함.
concat()	String 인스턴스에 전달받은 문자열을 결합한 새로운 문자열을 반환함.



## JavaScript

## 프로토타입 메소드

## 설명

toLowerCase()	String 인스턴스의 모든 문자를 소문자로 변환한 새로운 문자열을 반환함.
trim()	String 인스턴스의 양 끝에 존재하는 공백과 모든 줄 바꿈 문자(LF, CR 등)를 제거한 새로운 문자열을 반환함.
search()	인수로 전달받은 정규 표현식에 맞는 문자나 문자열이 처음으로 등장하는 위치의 인덱스를 반환함.
replace()	인수로 전달받은 패턴에 맞는 문자열을 대체 문자열로 변환한 새 문자열을 반환함.
match()	인수로 전달받은 정규 표현식에 맞는 문자열을 찾아서 하나의 배열로 반환함.
includes()	인수로 전달받은 문자나 문자열이 포함되어 있는지를 검사한 후 그 결과를 불리언 값으로 반환함.
startsWith()	인수로 전달받은 문자나 문자열로 시작되는지를 검사한 후 그 결과를 불리언 값으로 반환함.
endsWith()	인수로 전달받은 문자나 문자열로 끝나는지를 검사한 후 그 결과를 불리언 값으로 반환함.
toLocaleUpperCase()	영문자뿐만 아니라 모든 언어의 문자를 대문자로 변환한 새로운 문자열을 반환함.
toLocaleLowerCase()	영문자뿐만 아니라 모든 언어의 문자를 소문자로 변환한 새로운 문자열을 반환함.
localeCompare()	인수로 전달받은 문자열과 정렬 순서로 비교하여 그 결과를 정수 값으로 반환함.
normalize()	해당 문자열의 유니코드 표준화 양식(Unicode Normalization Form)을 반환함.
repeat()	해당 문자열을 인수로 전달받은 횟수만큼 반복하여 결합한 새로운 문자열을 반환함.
toString()	String 인스턴스의 값을 문자열로 반환함.
valueOf()	String 인스턴스의 값을 문자열로 반환함.



## JavaScript

```
let a = new String('Boys and Girls');
let b = '!!';
document.write('a : ' + a + '<br>');
document.write('b : ' + b + '<br><hr>');
document.write(a.charAt(0) + '<br>');
document.write(a.concat(b, '입니다') + '<br>');
document.write(a.indexOf('s') + '<br>');
document.write(a.indexOf('And') + '<br>');
document.write(a.slice(5, 8) + '<br>');
document.write(a.substr(5, 3) + '<br>');
document.write(a.toUpperCase() + '<br>');
document.write(a.replace('and', 'or') + '<br>');
document.write(' kitae '.trim() + '<br><hr>');

let sub = a.split(' ');
document.write('a를 빈칸으로 분리<br>');
for (let i = 0; i < sub.length; i++)
  document.write('sub' + i + '=' + sub[i] + '<br>');
document.write('<hr>String 메소드를 실행 후 a와 b 변함 없음<br>');
document.write('a : ' + a + '<br>');
document.write('b : ' + b + '<br>');
```

a : Boys and Girls

b : !!

B

Boys and Girls!!입니다

3

-1

and

and

BOYS AND GIRLS

Boys or Girls

kitae

a를 빈칸으로 분리

sub0=Boys

sub1=and

sub2=Girls

String 메소드를 실행 후 a와 b 변함 없음

a : Boys and Girls

b : !!

# 코어객체( String )

## 기초 연습 문제





조건] 아래 주어진 1번 ~ 10번 까지 아래와 같이 p태그에 출력하시오.

(단, 아래 문제에 주어진 문자는 모두 변수에 지정하여 사용, new String( )객체로 지정하지 말 것)

```
<div class="question">
  <h2>1. 문자열 길이 구하기</h2>
  <p id="q1"></p>
</div>
```

### 1. 문자 개수 세기

문자열 "Hello JavaScript"의 길이를 출력하세요. (length 사용)

### 2. 대소문자 변환

문자열 "javascript"을 모두 대문자로 바꾸고, "HELLO"를 모두 소문자로 바꾸기.

### 3. 특정 문자 가져오기

문자열 "Hello"에서

첫 번째 문자, 세번째 문자를 출력. (charAt() 또는 인덱스 [] 사용)

### 4. 부분 문자열 추출

문자열 "JavaScript"에서 "Script"만 잘라서 출력하세요. (substring() 또는 slice())

### 5. 문자 위치 찾기

문자열 "I like JavaScript"에서 "Java"라는 단어가 시작되는 인덱스를 출력하세요. (indexOf())



## 6. 문자 포함 여부 확인

문자열 "Programming is fun" 안에 "fun"이 포함되어 있는지 확인하세요. (includes())

## 7. 문자열 교체하기

문자열 "I like Java"에서 "Java"를 "JavaScript"로 바꿔보세요. (replace())

## 8. 앞뒤 공백 제거

문자열 "Hello World"에서 앞뒤의 공백을 제거하고 출력하세요. (trim())

## 9. 문자열 나누기

문자열 "apple,banana,grape"를 쉼표(,) 기준으로 잘라 배열로 만드세요. (split())

## 10. 문자열 이어붙이기

문자열 "Hello"와 "World"를 합쳐 "Hello World"로 출력하세요. (concat() 또는 + 연산자 사용)

8강

# 코어액체( Math )





## JavaScript

## ➤ Math 객체란?

- Math 객체는 수학에서 자주 사용하는 상수와 함수들을 미리 구현해 놓은 자바스크립트 표준 내장 객체입니다. Math 객체는 다른 전역 객체와는 달리 생성자(constructor)가 존재하지 않는다. 고로 new Math()로 객체를 생성하지 않음을 주의하자.

## ➤ Math 객체 종류

## 1. Math.ceil() : 올림 / Math.floor() : 내림

ceil()은 올림, floor()은 내림 기능을 한다. 소수점 자릿수에 상관없이 소수 자리의 값이 있다면 올리거나 내린다.

```
let num1 = 5.1;
let num2 = 5.7;

Math.ceil(num1); //6
Math.ceil(num2); //6

Math.floor(num1); //5
Math.floor(num2); //5
```



## JavaScript

## ➤ Math 객체 종류

## 2. Math.round() : 반올림 / toFixed()

```
Math.round(num1); //5  
Math.round(num2); //6
```

round()는 소수점 자리의 값에 따라 반올림을 한다.

만약 소수점 둘째 자리까지 표현할 것이며, 소수점 셋째 자리에서 반올림을 원할 경우 toFixed() 메소드를 사용할 수 있다.

```
let userRate = 30.1234;  
Number(userRate.toFixed(2));
```

괄호 안에는 표현할 소수점 자릿수를 말한다. 0이면 없다는 것으로 정수만 출력되고 2이면 소수점 둘째짜리까지 함께 출력된다. 다만 주의할 점은 문자열을 반환하기 때문에 숫자형으로 변환해야 한다.



## JavaScript

## ➤ Math 객체 종류

## 3. Math.random() : 랜덤

```
Math.random()
```

0에서 1 사이의 무작위 숫자를 생성한다. 계속 실행할 때마다 새로운 값이 생성되는 것을 볼 수 있다. 말 그대로 랜덤으로 생성된다.

```
Math.floor(Math.random() * 100) + 1
```

1부터 100 사이의 임의의 숫자를 출력 할 때 Math.floor( )와 함께 사용한다.

1. `Math.random()` = 0.6789
2. `Math.random() * 100` = 67.89
3. `Math.floor(Math.random() * 100)` = 67
4. `Math.floor(Math.random() * 100) + 1` = 68



## JavaScript

## ➤ Math 객체 종류

## 4. Math.max() : 최대값 / Math.min() : 최소값

```
Math.max(1, 4, 10, -1, 2.24) // 10  
Math.min(1, 4, 10, -1, 2.24) // -1
```

## 5. Math.abs() : 절대값

```
Math.abs(-1) // 1
```

## 6. Math.pow(n, m) : 제곱

```
Math.pow(2, 10); // 1024
```

## 7. Math.sqrt() : 제곱근

```
Math.sqrt(16); // 4  
Math.sqrt(2); // 1.414213562373095  
Math.sqrt(-1); // NaN
```



## JavaScript

## ➤ break : 종료하고 탈출

- break 문법은 현재의 반복문이나 조건문 내부 블록을 실행하다가 중지하고 빠져나오기 위한 문법이다.

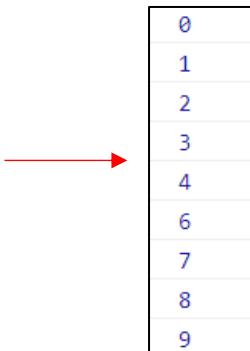
```
for(let i=0;i<10; i++){
    if(i<5){
        break;
    }
    console.log(i);
}
```



## ➤ continue : 다음 진행으로 넘어감

- continue 문법을 만나면 다음으로 진행하지 않고 조건이나 반복의 시작지점에서 이어서 진행하는 것을 의미한다.

```
for(let i=0;i<10; i++){
    if(i==5){
        continue;
    }
    console.log(i);
}
```



# 연습문제





예제1] 아래 주어진 조건에 만족하도록 html 문서를 작성하시오.

조건

① 랜덤 이름 뽑기

```
names = ["철수", "영희", "민수", "지영", "수현"];
```

- ② Math.random()을 이용해서 무작위로 한 명을 선택해 출력하시오.
- ③ <button>태그를 클릭하면 <p>태그에 출력되도록 작성 하시오.
- ④ 저장할 파일명은 Ex01.html로 작성 하시오

<출력 결과물>

1. 랜덤 이름 뽑기

정답 보기

뽑힌 이름: 민수



예제2] 아래 주어진 조건에 만족하도록 html 문서를 작성하시오.

조건

**① 랜덤 문자열 추출**

문자열 "JavaScript"에서 Math.random()을 이용해 무작위로 한 글자를 출력

- ② Math.random()을 이용해서 무작위로 한 글자를 선택해 출력하시오.
- ③ <button>태그를 클릭하면 <p>태그에 출력되도록 작성 하시오.
- ④ 저장할 파일명은 Ex02.html로 작성 하시오

<출력 결과물>

**2. 랜덤 문자열 추출**

정답 보기

뽑힌 글자: p



예제3] 아래 주어진 조건에 만족하도록 html 문서를 작성하시오.

조건

① 단어 길이 비교하기

단어 "apple"과 "watermelon" 중에서 더 긴 단어와 그 길이를 출력

- ② Length, Math.max()을 이용해서 출력하시오.
- ③ <button>태그를 클릭하면 <p>태그에 출력되도록 작성 하시오.
- ④ 저장할 파일명은 Ex03.html로 작성 하시오

<출력 결과물>

3. 단어 길이 비교하기

정답 보기

더 긴 단어: watermelon (길이: 10)



예제4] 아래 주어진 조건에 만족하도록 html 문서를 작성하시오.

조건

① 랜덤 비밀번호 생성기 (간단)

문자열 "ABCDEFGHIJKLMNOPQRSTUVWXYZ"에서 무작위로 5글자를 뽑아 **랜덤 비밀번호**를 만들어 출력

- ② Math.random(), Math.floor()을 이용해서 출력하시오.
- ③ <button>태그를 클릭하면 <p>태그에 출력되도록 작성 하시오.
- ④ 저장할 파일명은 Ex04.html로 작성 하시오

<출력 결과물>

4. 랜덤 비밀번호 생성기 (5글자)

정답 보기

생성된 비밀번호 : KQICP



예제5] 아래 주어진 조건에 만족하도록 html 문서를 작성하시오.

조건

① 문자열 공백 제거 후 길이 출력

문자열 " Hello JavaScript "에서 공백을 제거한 후(trim()), 문자열의 길이를 출력

- ② trim(), length을 이용해서 출력하시오.
- ③ <button>태그를 클릭하면 <p>태그에 출력되도록 작성 하시오.
- ④ 저장할 파일명은 Ex05.html로 작성 하시오

<출력 결과물>

5. 문자열 공백 제거 후 길이 출력

정답 보기

공백 제거 후: [Hello JavaScript], 길이: 16

## 연습문제 6



예제6] 아래 주어진 조건에 만족하도록 html 문서를 작성하시오.

### 조건

- ① Math.random()과 Math.floor() 객체를 이용하여 작성하시오.
- ② Function updown() 함수를 이용하여 작성하시오.
- ③ If문을 이용하여 작성하시오.
- ④ innerHTML을 이용하여 작성하시오.
- ⑤ 저장할 파일명은 Ex\_undown.html로 작성 하시오

