



## Chapter 05-1

# 객체-1 ( 내장 객체 )-3



# String 내장 객체

# String 내장 객체

## ■ String 객체

- 문자열에 대한 연산을 수행하고 문자열로부터 정보를 추출하는 객체
- String 객체의 속성
  - **length**
    - 글자의 길이를 지정하는 속성 (유일한 속성)
- String 객체의 메서드
  - 글자의 크기 모양, 효과를 적용하기 위한 메서드
    - `big()`, `small()`, `bold()`, `fixed()`, `italic()`, `strike()`, `sup()`, `sub()` 등
  - 문자열을 처리하기 위한 메서드
    - `CharAt()`, `IndexOf()`, `lastIndexOf()`, `match()`, `replace()`, `slice()`, `substring()`, `substr()`, 등

# String 내장 객체

## ■ 문자열 객체를 생성하는 방법

- new 연산자와 String() 객체를 사용하는 방법

```
변수 = new String("문자열")
```

- [예] `var str = new String("korea");`

- 문자 리터럴을 직접 지정하여 생성하는 방법

```
var 변수 = 문자열
```

- [예] `var str = "korea";`

# String 내장 객체 (속성)

## ■ length() 속성

- 문자열의 길이를 반환하는 속성
  - String 객체의 유일한 속성
  - 영문, 숫자, 한글 문자의 한 문자에 대해 1 값을 반환

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">

    var str1 = new String("korea");
    var str2 = "대한민국";

    document.write("str1 문자열의 길이 : ", str1.length, "<br>");
    document.write("str2 문자열의 길이 : ", str2.length, "<br>");

  </script>
</head>
<body> </body>
</html>
```

5-String-length.htm

```
str1 문자열의 길이 : 5
str2 문자열의 길이 : 4
```

## String 내장 객체 ( 메서드 )

### ■ String 내장 객체의 메서드 ( 문자 위치 관련 )

메서드	설명
charAt(index)	문자열 중에서 인자로 가지는 index에 해당하는 문자를 반환
indexOf("문자")	문자열을 처음부터 검색하여 인자로 지정된 문자의 처음 나타나는 위치를 반환
lastIndexOf("문자")	문자열을 마지막부터 검색하여 인자로 지정된 문자의 처음 나타나는 위치를 반환

# String 내장 객체 ( 메서드 )

## ■ charAt(index) 메서드

- 문자열 객체에서 인자로 지정한 index에 해당하는 위치의 문자를 추출

```
var 변수 = 문자열.charAt(index)
```

- 문자열 객체의 맨 왼쪽 문자(index:0)를 기준으로 검색함

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">

    var str = "i love korea";

    var data = str.charAt(7);
    document.write("index가 7인 위치의 문자 : ", data);

  </script>
</head>
<body> </body>
</html>
```

5-String-charAt.htm

index가 7인 위치의 문자 : k

# String 내장 객체 ( 메서드 )

## ■ indexOf()와 lastIndexOf() 메서드

### • indexOf() 메서드

```
var 변수 = 문자열.indexOf( "문자", [시작위치] )
```

- 문자열 객체의 왼쪽부터 검색하여 인자로 가지는 문자의 위치 값을 숫자로 반환
  - 문자열 객체의 가장 왼쪽 문자의 위치 값은 0임
- [시작 위치]를 지정한 경우 지정된 index의 문자부터 검색
  - 시작 위치만을 지정할 뿐 문자열 객체의 가장 왼쪽 문자의 위치 값은 0
  - 찾는 문자가 존재하지 않을 경우 -1을 반환

### • lastIndexOf() 메서드

```
var 변수 = 문자열.lastIndexOf( "문자", [시작위치] )
```

- 문자열의 오른쪽부터 검색하여 인자로 가지는 문자의 위치 값을 숫자로 반환
  - 검색은 오른쪽부터 수행하지만 위치 값의 기준은 문자열 객체의 왼쪽문자 (위치 값:0) 임
  - 찾는 문자가 존재하지 않을 경우 -1을 반환



# String 내장 객체 ( 메서드 )

## 5-String-indexOf-lastIndexOf.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">

    str = "more than words";
    document.write("문자열 객체 : ", str, "<hr><br>");

    str1 = str.indexOf('o');
    document.write("왼쪽 처음부터 검색한 문자 o의 위치 값 : ", str1, "<br><br>");

    str2 = str.indexOf('o',5);
    document.write("인덱스가 5인 문자부터 검색한 문자 o의 위치 값 : ", str2, "<br><br>");

    str3 = str.lastIndexOf('o');
    document.write("뒤에서 검색한 문자 o의 위치 값 : ", str3);

  </script>
</head>
<body> </body>
</html>
```

문자열 객체 : more than words

왼쪽 처음부터 검색한 문자 o의 위치 값 : 1

인덱스가 5인 문자부터 검색한 문자 o의 위치 값 : 11

뒤에서 검색한 문자 o의 위치 값 : 11

# String 내장 객체 ( 메서드 )

## ■ String 내장 객체의 메서드 ( 문자의 검색과 추출 관련 )

메서드	설명
match(“문자”)	인자로 지정한 문자와 동일한 문자가 존재하는지 조사 - 존재할 경우 그 문자(단어)를 반환 - 존재하지 않을 경우 null을 반환
search(“문자”)	match()와 동일한 기능 수행 - 존재할 경우 위치를 반환 - 존재하지 않을 경우 null을 반환
replace(str1, str2)	문자열을 검색하여 str1이 존재하는 경우 이를 str2로 치환
substring( m, n )	문자열 객체 중 위치 값 m번부터 n번까지의 문자열을 추출해 반환
slice( m, n )	substring() 메소드와 기능은 동일
substr( m, n )	문자열 객체의 위치 값 m번부터 n개의 문자열을 추출해 반환

# String 내장 객체 ( 메서드 )

## ■ match() 메서드

```
var 변수 = 문자열.match("문자")
```

- 문자열에서 인자로 지정한 문자를 왼쪽부터 검색한 후 나타나는 첫번째 문자를 반환
  - 인자는 하나의 문자일 수 있고, 단어일 수도 있음
  - 발견되지 않을 경우 null을 반환

## ■ search() 메서드

```
var 변수 = 문자열.search("문자")
```

- 문자열에서 인자로 지정한 문자를 왼쪽부터 검색한 후 나타나는 첫번째 문자의 index를 반환
  - 발견되지 않을 경우 -1을 반환

# String 내장 객체 ( 메서드 )

5-String-match-search.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">

    var str = "i love you and you love me";
    document.write("문자열 객체 : ", str, "<br>");

    var data1 = str.match('you')
    document.write("문자열 검색 : ", data1, "<br>");

    var data2 = str.search('you')
    document.write("문자열의 위치 : ", data2);

  </script>
</head>
<body> </body>
</html>
```

문자열 객체 : i love you and you love me  
문자열 검색 : you  
문자열의 위치 : 7

## String 내장 객체 ( 메서드 )

### ■ replace() 메서드

```
var 변수 = 문자열. replace( str1, str2 )
```

- 문자열을 검색하여 str1이 존재하는 경우 str2로 치환하여 반환
  - str1이 존재하지 않을 경우 원래 문자열 전체를 반환

# String 내장 객체 ( 메서드 )

5-String-replace.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">

    var str = "i love you";
    document.write("원래 문자열 : ", str, "<br><hr>");

    var str1 = str.replace('love', 'like');
    document.write("love를 like로 : ", str1, "<br><br>");

    var str2 = str.replace('LOVE', 'like');
    document.write("LOVE를 love로 : ", str2, "<br><br>");

  </script>
</head>
<body> </body>
</html>
```

원래 문자열 : i love you

love를 like로 : i like you

LOVE를 love로 : i love you

# String 내장 객체 ( 메서드 )

## ■ substring() 메서드

```
var 변수 = 문자열.substring( m, n )
```

- 문자열 중 index 값이 m인 위치 문자부터 n인 위치 문자까지 추출해 새로운 문자열을 반환
  - m의 경우 첫번째 문자의 index는 0임
  - n의 경우 첫번째 문자의 index는 1임
  - n과 m은 음수 값 사용 불가능

## ■ slice()메서드

```
var 변수 = 문자열.slice( m, n )
```

- substring() 메소드와 기능은 동일하나, n과 m은 음수 값 사용 가능
  - 음수일 경우 오른쪽부터 검색
  - m이 음수일 경우, 오른쪽 문자의 위치 값은 1부터 시작
  - n이 음수일 경우, 오른쪽 문자의 위치 값은 0부터 시작

# String 내장 객체 ( 메서드 )

5-String-substring-slice.htm

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<script type="text/javascript">
```

```
str = "more than words";
```

```
document.write("문자열 객체 : ", str, "<hr><br>");
```

```
str1 = str.substring(5,9);
```

```
document.write("위치값 5번 문자부터 9번 문자까지 출력(subtring) : ", str1, "<br><br>");
```

```
str2 = str.slice(5,9);
```

```
document.write("위치값 5번 문자부터 9번 문자까지 출력(slice) : ", str2, "<br><br>");
```

```
str3 = str.slice(5,-2);
```

```
document.write("위치값 5번 문자부터 오른쪽에서 3번째 문자까지 출력(slice) : ", str3);
```

```
</script>
```

```
</head>
```

```
<body> </body>
```

```
</html>
```

문자열 객체 : more than words

위치값 5번 문자부터 9번 문자까지 출력(subtring) : than

위치값 5번 문자부터 9번 문자까지 출력(slice) : than

위치값 5번 문자부터 오른쪽에서 3번째 문자까지 출력(slice) : than wor



# String 내장 객체 ( 메서드 )

## ■ substr() 메서드

```
var 변수 = 문자열.substr( m, n )
```

- 문자열의 index 값이 m인 위치부터 n개의 문자열을 추출해 반환

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <script type="text/javascript">
```

```
    str = "more than words";
```

```
    document.write("문자열 객체 : ", str, "<hr><br>");
```

```
    str1 = str.substr(5,4);
```

```
    document.write("오른쪽 위치값 5번 문자부터 4개의 문자 출력 : ", str1);
```

```
  </script>
```

```
</head>
```

```
<body> </body>
```

```
</html>
```

5-String-substr.htm

문자열 객체 : more than words

오른쪽 위치값 5번 문자부터 4개의 문자 출력 : than

# String 내장 객체 ( 메서드 )

## ■ String 내장 객체의 메서드 ( 문자의 크기와 모양 관련 )

메서드	설명
big( )/ small( )	문자열의 크기를 크게/작게 설정
fontsize(숫자)	문자열의 크기를 지정. 1(작게)~7(크게)까지 지정 가능
fontcolor("색상명")	문자열의 색 지정
bold( )	문자열을 굵게 표현
italics( )	문자열을 탸릭체 표현
strike( )	문자열에 취소선을 표현
sup( )/ sub()	문자열을 윗첨자/ 아랫첨자로 표현
fixed()	문자열을 타자기체로 표현
anchor()	문자열을 <a> 태그로 감싸 반환
link(URL)	문자열을 <a> 태그로 감싸고 href 속성에 URL을 지정해 반환

# String 내장 객체 ( 메서드 )

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">

    str = "i love korea";

    document.write(str.big(), "<hr>");
    document.write(str.small(), "<hr>");
    document.write(str.fontSize(7), "<hr>");
    document.write(str.fontcolor('#0000ff'), "<hr>");
    document.write(str.fixed(), "<hr>");
    document.write(str.bold(), "<hr>");
    document.write(str.italics(), "<hr>");
    document.write(str.strike(), "<hr>");
    document.write(str + "Fighting".sup(), "<hr>");
    document.write(str + "Fighting".sub());

    alert(str.anchor());
    alert(str.link('www.ut.ac.kr'));

  </script>
</head>
<body> </body>
</html>
```

localhost:8080 내용:  
<a name="undefined">i love korea</a>

확인

## 5-String-font.htm

i love korea

i love korea

i love korea

i love korea

i love korea

**i love korea**

*i love korea*

~~i love korea~~

i love korea<sup>Fighting</sup>

i love korea<sub>Fighting</sub>

localhost:8080 내용:  
<a href="www.ut.ac.kr">i love korea</a>

확인

# String 내장 객체 ( 메서드 )

## ■ String 내장 객체의 메서드 ( 기타 문자열 관련 )

메서드	설명
split("구분자")	문자열 객체의 각 단어를 '구분자'로 분할하여 배열로 반환
concat("문자")	기존 문자열에 인자로 지정한 문자열을 결합하여 새로운 문자열 생성
toUpperCase()	문자열을 모두 대문자로 변환
toLowerCase()	문자열을 모두 소문자로 변환
charCodeAt(index)	문자열 중 index에 해당하는 위치 문자의 ASCII 코드를 반환
fromCharCode(ASCII코드)	인자로 가지는 ASCII 코드에 해당하는 문자를 반환
trim()	문자열의 좌우에 존재하는 공백문자를 제거함

# String 내장 객체 ( 메서드 )

## ■ split() 메서드

```
var 변수 = 문자열.split("문자")
```

- 문자열을 인자로 지정한 문자로 분할하여 배열로 반환

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">

    str1 = "more than words";
    document.write("원래 문자열 : " + str1 + "<hr>");

    var array = str1.split(' ');
    for(var i in array) {
      document.write("array["+i+"] : ", array[i], "<br>");
    }

  </script>
</head>
<body> </body>
</html>
```

5-String-slit.htm

원래 문자열 : more than words

array[0] : more  
array[1] : than  
array[2] : words

# String 내장 객체 ( 메서드 )

## ■ concat() 메서드

문자열 변수 = 문자열1.concat("문자열2")

- 문자열1에 문자열2를 결합해 새로운 문자열을 반환

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">

    var str1 = "안녕하세요.";
    var str2 = "반갑습니다.";
    document.write(str1, "<br>");
    document.write(str2, "<hr>");

    var str3 = str1.concat(str2);
    document.write(str3);

  </script>
</head>
<body> </body>
</html>
```

5-String-concat.htm

안녕하세요.  
반갑습니다.

안녕하세요.반갑습니다.

## String 내장 객체 ( 메서드 )

### ■ toLowerCase() 메서드

```
문자열변수 = 문자열.toLowerCase( )
```

- 문자열을 소문자로 변환하여 반환

### ■ toUpperCase() 메서드

```
문자열변수 = 문자열.toUpperCase( )
```

- 문자열을 대문자로 변환하여 반환

# String 내장 객체 ( 메서드 )

5-String-lower-uppercase.htm ( p.109 )

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">

    var userName = prompt("당신의 영문 이름은?", "");

    var upperName = userName.toUpperCase();
    document.write(upperName, "<br>");

    var userNum = prompt("당신의 연락처는?", "");
    var result = userNum.substring(0, userNum.length - 4) + "****";
    document.write(result, "<br>");

  </script>
</head>
<body> </body>
</html>
```

JOHN SMITH  
010-1234-\*\*\*\*

localhost:8080 내용:

당신의 영문 이름은?

john smith

확인

취소

localhost:8080 내용:

당신의 연락처는?

010-1234-5678

확인

취소



## String 내장 객체 ( 메서드 )

### ■ charCodeAt() 메서드

```
숫자형변수 = 문자열.charCodeAt( index )
```

- 문자열 중 인자로 지정한 index의 위치 문자를 ASCII 코드 값으로 변환

### ■ fromCharCode() 메서드

```
문자열변수 = String.fromCharCode( ASCII )
```

- 인자로 지정된 ASCII 값을 문자로 변환

# String 내장 객체 ( 메서드 )

5-String-charcode.htm

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">

    var str = "ABC";

    var ascii_num1 = str.charCodeAt(1);
    document.write("index가 1인 문자의 ASCII 문자 : ", ascii_num1, "<br>");

    var ascii_num2 = String.fromCharCode(65);
    document.write(" ASCII 문자가 65인 문자 : ", ascii_num2, "<br>");

  </script>
</head>
<body> </body>
</html>
```

index가 1인 문자의 ASCII 문자 : 66  
ASCII 문자가 65인 문자 : A

# String 내장 객체 ( 메서드 )

## ■ trim() 메서드

문자열변수 = 문자열1.trim()

- 문자열1의 좌우 공백을 제거한 다음 문자열2에 저장

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <script type="text/javascript">

    var str1 = " ABCD ";
    document.write("trim 이전의 문자길이 : ", str1.length, "<br>");

    str2 = str1.trim();
    document.write("trim 이후의 문자길이 : ", str2.length, "<br>");

  </script>
</head>
<body> </body>
</html>
```

5-String-trim.htm

trim 이전의 문자길이 : 10  
trim 이후의 문자길이 : 4

**수고하셨습니다**