



JavaScript Strings

```
let x = "John";
```

```
let y = new String("John");
```

문자열은 키워드를 사용하여 객체로 정의 할 수도 있습니다.

String Length

```
let text = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
```

```
let length = text.length;
```

문자열의 일부를 추출하는 방법에는 3가지가 있습니다.

`slice(start, end)`

`substring(start, end)`

`substr(start, length)`

`slice()` 메서드는 문자열의 일부를 추출하고 새 문자열에서 파트를 추출했습니다.

이 방법은 시작 위치와 끝 위치 (끝은 포함되지 않음)의 2 가지 매개 변수를 사용합니다.
첫 번째 위치는 0입니다.

```
let text = "Apple, Banana, Kiwi";
```

```
let part = text.slice(7, 13);
```

결과 : Banana

두 번째 매개 변수를 생략하면 메서드는 문자열의 나머지 부분을 조각냅니다.

```
let text = "Apple, Banana, Kiwi";
```

```
let part = text.slice(7);
```

결과 : Banana, Kiwi

매개 변수가 음수이면 위치는 문자열의 끝에서 계산됩니다.

```
let text = "Apple, Banana, Kiwi";
```

```
let part = text.slice(-12);
```

결과 : Banana, Kiwi

```
let text = "Apple, Banana, Kiwi";
```

```
let part = text.slice(-12, -6);
```

결과 : Banana

JavaScript String substring()

`substring()`는 `slice()` 와 비슷하지만 0보다 작은 시작 및 끝값은 0으로 처리됩니다.

두 번째 매개 변수를 생략하면 나머지 매개 변수를 잘라냅니다.

```
let str = "Apple, Banana, Kiwi";
```

```
let part = str.substring(7, 13);
```

JavaScript String substr()

substr() 메서드는 slice() 와 비슷하지만 두 번째 매개변수는 추출 된 문자의 길이를 지정합니다.

두 번째 매개 변수를 생략하면 나머지 매개 변수를 잘라냅니다.

```
let str = "Apple, Banana, Kiwi";
```

```
let part = str.substr(7, 6);
```

결과 : Banana

```
let str = "Apple, Banana, Kiwi";
```

```
let part = str.substr(7);
```

결과 : Banana, Kiwi

매개 변수가 음수이면 위치는 문자열의 끝에서 계산됩니다.

```
let str = "Apple, Banana, Kiwi";
```

```
let part = str.substr(-4);
```

결과 : Kiwi

Replacing String Content

Replace() 메서드는 지정된 값을 다른 값으로 바꿉니다

```
let text = "Please visit Microsoft!";
```

```
let newText = text.replace("Microsoft", "W3Schools");
```

메서드는 첫 번째 일치 항목만 대체합니다.

```
let text = "Please visit Microsoft and Microsoft!";
```

```
let newText = text.replace("Microsoft", "W3Schools");
```

기본적으로 메서드는 대/소문자를 구분합니다.

```
let text = "Please visit Microsoft!";
```

```
let newText = text.replace(/MICROSOFT/i, "W3Schools");
```

모든 항목을 바꾸려면 정규식을 사용 /g

```
let text = "Please visit Microsoft and Microsoft!";
```

```
let newText = text.replace(/Microsoft/g, "W3Schools");
```

JavaScript String ReplaceAll()

```
let text = "I love cats. Cats are very easy to love. Cats are very popular."
```

```
text = text.replaceAll("Cats", "Dogs");
```

```
text = text.replaceAll("cats", "dogs");
```

toUpperCase() 메서드는 문자열을 대문자로 변환됩니다.

toLowerCase() 메서드는 문자열을 소문자로 변환됩니다.

```
let text1 = "Hello";
```

```
let text2 = "World";
```

```
let text3 = text1.concat(" ", text2);
```



`trim()` 메서드는 문자열의 양쪽에서 공백을 제거합니다.

```
let text1 = "    Hello World!    ";  
let text2 = text1.trim();
```

`charAt()` 메서드는 문자열에서 지정된 인덱스(위치)에 있는 문자를 반환합니다.

첫 번째 문자의 인덱스는 0, 두 번째 문자는 1, ...

```
let text = "HELLO WORLD";  
let char = text.charAt(0);
```

`charCodeAt()` : 문자의 유니코드를 반환합니다.

UTF-16 코드(0에서 65535 사이의 정수)를 반환합니다.

```
let text = "HELLO WORLD";  
let char = text.charCodeAt(0);
```

문자열에 대한 속성 액세스 []를 허용합니다.

```
let text = "HELLO WORLD";  
let char = text[0];
```

`split()` 메서드는 문자열을 하위 문자열의 배열로 분할합니다.

`split()` 메서드는 새 배열을 반환합니다.

`split()` 메서드는 원래 문자열을 변경하지 않습니다.

(" ")가 구분 기호로 사용되면 문자열이 단어 사이에 분할됩니다.

```
let text = "a,b,c,d,e,f";  
const myArray = text.split(",");
```

```
let text = "Hello";
```

```
const myArr = text.split("");
```

`search()` 메서드는 문자열을 정규 표현식과 일치시킵니다.

`search()` 메서드는 첫 번째 일치 항목의 인덱스(위치)를 반환합니다.

`search()` 메서드는 일치하는 항목이 없으면 -1을 반환합니다.

`search()` 메서드는 대소문자를 구분합니다.

```
let text = "search"  
let position = text.search("a");
```

`replace()` 메서드는 문자열에서 값 또는 정규식을 검색합니다.

`replace()` 메서드는 값이 교체된 새 문자열을 반환합니다.

`replace()` 메서드는 원래 문자열을 변경하지 않습니다.

```
let text = "Visit Microsoft!";
```

```
let result = text.replace("Microsoft", "W3Schools");
```

match() 메서드는 문자열을 정규식과 일치시킵니다.
match() 메서드는 일치 항목이 있는 배열을 반환합니다.
match() 메서드는 일치하는 항목이 없으면 null을 반환합니다.

```
let text = "Visit Microsoft!";  
let result = text.match("Microsoft");
```

JavaScript Events

event	Description
onchange	HTML 요소가 변경되었습니다.
onclick	사용자가 HTML 요소를 클릭합니다.
onmouseover	사용자가 HTML 요소 위로 마우스를 이동합니다.
onmouseout	사용자가 HTML 요소에서 멀리 마우스를 이동합니다.
onkeydown	사용자가 키보드 키를 누릅니다
onload	브라우저가 페이지 로드를 완료했습니다.

이벤트 핸들러는 사용자 입력, 사용자 작업 및 브라우저 작업 등을 처리하고 확인하는 데 사용할 수 있습니다.

페이지가 로드 될 때마다해야 할 일
페이지를 닫을 때 수행해야 할 작업
사용자가 단추를 클릭할 때 수행해야 하는 작업
사용자가 데이터를 입력할 때 확인해야 하는 콘텐츠

JavaScript가 이벤트와 함께 작동하도록 하기 위해 다양한 방법을 사용할 수 있습니다.

HTML 이벤트 속성은 자바 스크립트 코드를 직접 실행할 수 있습니다.
HTML 이벤트 속성은 자바 스크립트 함수를 호출 할 수 있습니다.
사용자 고유의 이벤트 처리기 함수를 HTML 요소에 할당할 수 있습니다.
이벤트가 전송되거나 처리되지 않도록 할 수 있습니다.