**Ch08 기본객체**

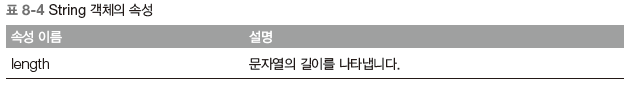
**w**3schools(https://www.w3schools.com/jsref) 자바스크립트의 기본 내장객체의 속성과 메소드

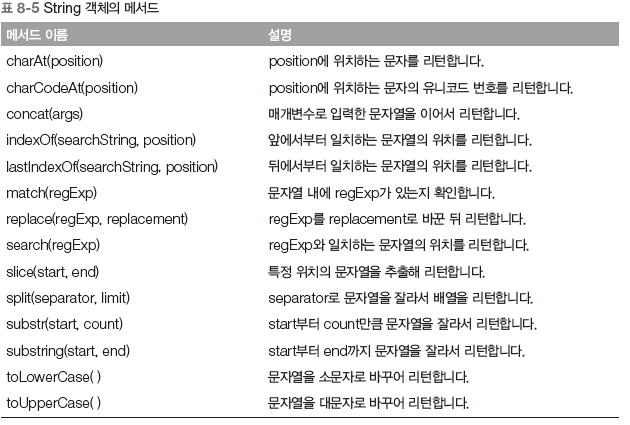
devdocs.io(http://devdocs.io/javascript ) 각각의 속성, 메소드에 대한 자세한 정보 제공

**String : 문자열을 다루기 위한 객체**

**Date : 날짜와 시간을 처리하기 위한 객체**

8.1 String 객체





* String 객체의 메서드는 자기 자신을 변화시키지 않고 리턴하므로 문자열을 변화시키고 싶으면 아래와 같이 사용한다.

<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

<script>

var str = '@abcd@ef';

var patternNum = /[0-9]/; //숫자 패턴

var patternEng = /[a-zA-Z]/; //문자 패턴

var patternSpe = /[~!@#$%^&\*()\_+|<>?:;{}]/; // 특수문자 패턴

document.write('글씨길이 : ' + str.length + '<br>');

document.write('스트링 더하기 : ' + str.concat(str) + '<br>');

document.write('부분 글씨 : ' + str.substr(3, 2) + '<br>');

document.write('앞에서부터 일치하는 글자 위치 : '

+ str.indexOf('\*') + '<br>');

document.write('뒤에서부터 일치하는 글자위치'

+ str.lastIndexOf('@') + '<br>');

document.write('1번째부터 2번째 앞까지 :' + str.slice(1, 2)+'<br>');

document.write('str을 다 대문자로 : '+str.toUpperCase()+'<br>');

document.write('str에 숫자가 포함되었는지 : '+patternNum.test(str) +'<br>');

document.write('str에 알파벳이 포함되었는지 : '+patternEng.test(str) +'<br>');

document.write('str에 숫자가 포함되었는지 : '+patternSpe.test(str) +'<br>');

document.write('str='+str)

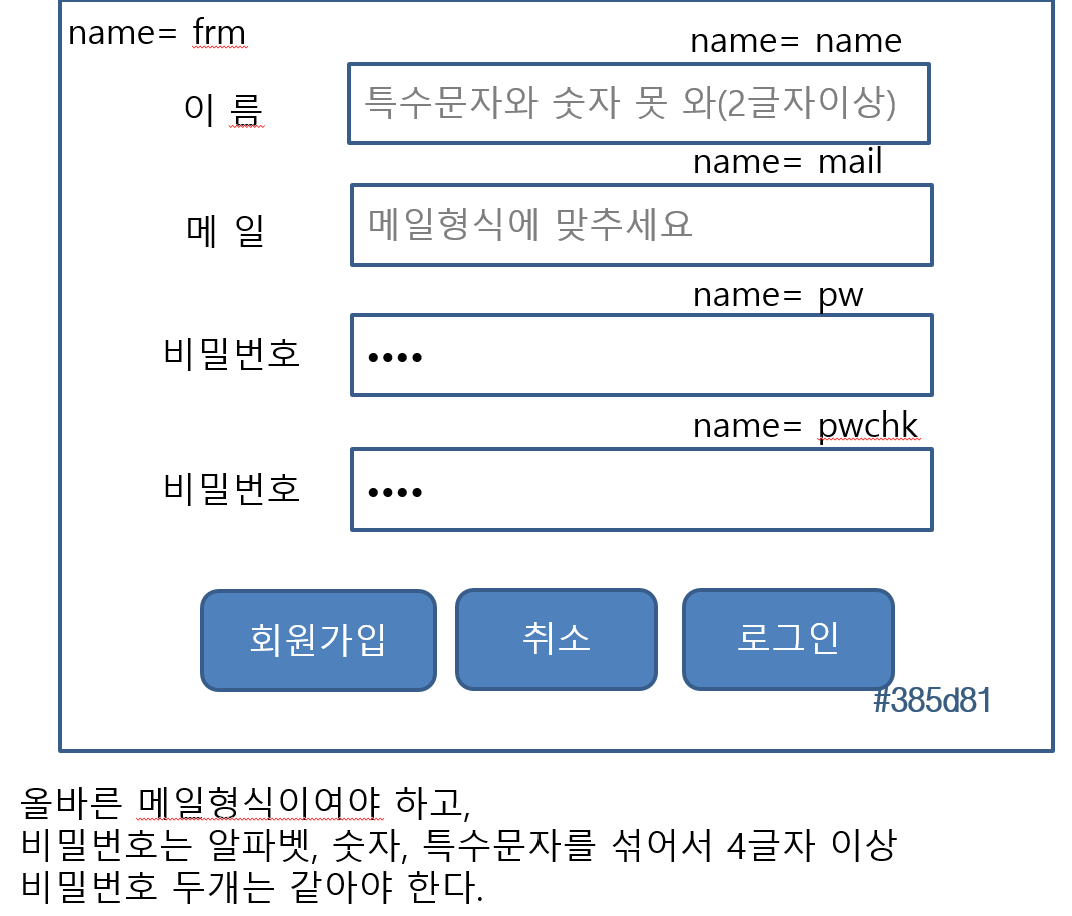
</script>

</head>

<body></body>

</html>

QUIZ : 다음 페이지를 자바스크립트로 구현하시오



hint : 특수문자 패턴

// 특수문자패턴(\-은- , \\은 \ , \/은 / , \[ 은 [ , \]은 ] , \|은 |)

var patternSpc = /[~`!@#$%^&\*()\_\-+=\|\\\[\]{}'";;?\/<>,\.]/;

<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

  <meta charset="utf-8" />

  <title></title>

  <style>

    table {width:350px; margin:20px auto; padding:20px 10px;}

    table, input[type='text'], input[type='password'] {

        border: 1px solid #385d81;

outline: 1px solid #385d81;

        padding: 2px;

    }

    input[type='submit'], input[type='reset'], input[type='button'], button{

        background-color: #385d81;

        border:0; outline:0;

        color:white;

        border-radius: 3px;

        padding:10px 20px;

        margin:5px;

    }

    td, th { padding:5px 10px; text-align:center;}

  </style>

  <script>

  </script>

</head>

<body>

  <form name="frm" onsubmit="return chk()">

    <table>

      <tr>

        <th>이름</th>

        <td>

          <input type="text" name="name" placeholder="특수문자와 숫자 못 와"

                        autofocus="autofocus"/>

        </td>

      </tr>

      <tr>

        <th>메일</th>

        <td><input type="text" name="mail" placeholder="메일 형식에 맞추세요" /></td>

      </tr>

      <tr>

        <th>비밀번호</th>

        <td><input type="password" name="pw" placeholder="특수문자,알파벳,숫자 포함" /></td>

      </tr>

      <tr>

        <th>비밀번호</th>

        <td><input type="password" name="pwchk" placeholder="4글자 이상" /></td>

      </tr>

      <tr>

        <td colspan="2">

          <input type="submit" value="회원가입" />

          <input type="reset" value="취소" />

          <input type="button" value="로그인" />

        </td>

      </tr>

    </table>

  </form>

</body>

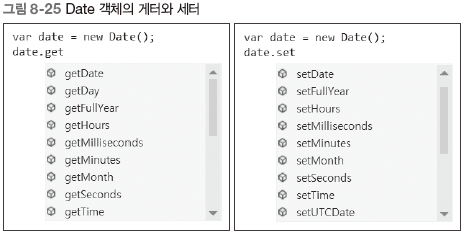
</html>

8.2 Date 객체

8.2.1 생성

- 특정한 날짜를 생성하기 위해서는, Date의 매개변수에 숫자를 연, 월-1, 일, 시, 분, 초, 밀리초 순서로 입력해 객체를 생성한다.

8.6.2 메서드

연도를 가져올 경우 getFullYear() 추천

<script>

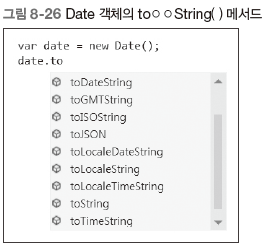
var date = new Date();

// 현재에서 7일을 더합니다.

date.setDate(date.getDate() + 7);

alert(date);

</script>



<script>

var date = new Date(2020, 0, 1, 13, 5 , 30);

var output = '';

output += '★toDateString: ' + date.toDateString() + '\n';

output += '★toLocaleDateString: ' + date.toLocaleDateString() + '\n';

output += '★toLocaleString: ' + date.toLocaleString() + '\n';

output += '★toLocaleTimeString: ' + date.toLocaleTimeString() + '\n';

output += '★toTimeString: ' + date.toTimeString() + '\n';

output += '★toString: ' + date.toString() + '\n';

output += '★toUTCString: ' + date.toUTCString() + '\n';

</script>

8.2.2 날짜 간격 계산

- Date 객체는 D-Day를 구할 때도 활용할 수 있다. 날짜 간격을 구할 때는 getTime() 함수를 사용한다. getTime()함수는 1970년 1월1일 자정부터 지난 밀리초를 구한다. 이를 사용해 두 시간 사이의 초간격을 구하고 다시 날짜로 바꿀 수 있다.

<script>

var now = new Date();

var before = new Date(2020,0,5); // 1월5일 0시 0분

var interval = now.getTime() - before.getTime();

interval = Math.floor(interval / (1000 \* 60 \* 60 \* 24));

alert('Interval: ' + interval + '일');

</script>

- 아래와 같이 Date 객체의 프로토타입에 메서드를 넣으면 쉽고 편리하게 날짜 간격을 구할 수도 있다.

<script>

// Date생성자 함수의 프로토타입에 메서드를 추가합니다.

Date.prototype.getInterval = function (otherDate) {

// 변수를 선언합니다.

var interval;

// 양수로 날짜 간격을 구하려고 조건문을 사용합니다.

if (this > otherDate) {

interval = this.getTime() - otherDate.getTime();

} else {

interval = otherDate.getTime() - this.getTime();

}

// 리턴합니다.

return Math.floor(interval / (1000 \* 60 \* 60 \* 24));

};

// 변수를 선언합니다.

var now = new Date();

var before = new Date(2016,0,27);

// 출력합니다.

alert('Interval: ' + now.getInterval(before) + '일');

</script>

- 아래와 같이 Date 객체의 프로토타입에 메서드를 interval.js로 빼고 간편하게 날짜 간격을 구할 수도 있다.

// interval.js

Date.prototype.getInterval = function (otherDate) {

// 변수 선언합니다.

var interval;

if (this > otherDate) {

interval = this.getTime() - otherDate.getTime();

} else {

interval = otherDate.getTime() - this.getTime();

}

return Math.floor(interval / (1000 \* 60 \* 60 \* 24));

};

<script src="interval.js"></script>

<script>

var now = new Date();

var before = new Date(2020, 0, 5);

alert('interval : ' + now.getInterval(before));

</script>