

## JavaScript Math Object

JavaScript Math 개체를 사용하면 숫자에 대한 수학적 작업을 수행할 수 있습니다.  
다른 객체와 달리 Math 객체에는 생성자가 없습니다.  
Math 객체를 먼저 생성하지 않고도 모든 메서드와 속성을 사용할 수 있습니다.

### Math Properties

Math 속성의 구문은 Math.property입니다.

JavaScript는 Math 속성으로 액세스할 수 있는 8개의 수학 상수를 제공합니다.

ex)

```
Math.E      // 오일러 수를 반환합니다.
Math.PI     // PI 반환
Math.SQRT2   // 2의 제곱근을 반환합니다
Math.SQRT1_2 // 1/2의 제곱근을 반환합니다.
Math.LN2     // 2의 자연 로그를 반환합니다.
Math.LN10    // 10의 자연 로그를 반환합니다.
Math.LOG2E   // E의 밑이 2인 로그를 반환합니다.
Math.LOG10E  // E의 밑이 10인 로그를 반환합니다.
```

### Math Methods

Math 모든 메서드의 구문은 Math.method(숫자)입니다.  
숫자를 정수로 반올림하는 4가지 일반적인 방법이 있습니다.

Math.round(x)	가장 가까운 정수를 반환합니다.
Math.ceil(x)	가장 가까운 정수로 반올림한 x 값을 반환합니다.
Math.floor(x)	Math.floor(x)는 가장 가까운 정수로 내림한 x 값을 반환합니다.
Math.trunc(x)	x의 정수 부분을 반환합니다.

### Math.sign()

x가 음수, null 또는 양수이면 반환합니다.

ex)

```
Math.sign(-4);
Math.sign(0);
Math.sign(4);
```

### Math.pow()

Math.pow(x, y)는 x의 값을 y의 거듭제곱으로 반환합니다.

ex)

```
Math.pow(8, 2);
```

### Math.sqrt()

Math.sqrt(x)는 x의 제곱근을 반환합니다.

ex)

```
Math.sqrt(64);
```

### Math.abs()

Math.abs(x)는 x의 절대(양수) 값을 반환합니다.

ex)

```
Math.abs(-4.7);
```

### Math.min() and Math.max()

Math.min() 및 Math.max()는 인수 목록에서 가장 낮거나 가장 높은 값을 찾는 데 사용할 수 있습니다.

ex)

```
Math.min(0, 150, 30, 20, -8, -200);
```

```
Math.max(0, 150, 30, 20, -8, -200);
```

### Math.random()

Math.random()은 0(포함)과 1(제외) 사이의 난수를 반환합니다.

ex)

```
Math.random();
```

## onscroll Event

onscroll 이벤트는 요소의 스크롤 막대가 스크롤될 때 발생합니다.

### Syntax

In HTML:

```
<element onscroll="myScript">
```

In JavaScript:

```
object.onscroll = function(){myScript};
```

scrollTop 속성은 요소의 콘텐츠가 세로로 스크롤되는 픽셀 수를 설정하거나 반환합니다.

scrollLeft 속성은 요소의 콘텐츠가 가로로 스크롤되는 픽셀 수를 설정하거나 반환합니다.

### Syntax

element.scrollTop

element.scrollLeft

ex)

```
<div id="myDIV" onscroll="myfnc()">
  <div id="content">Scroll me!</div>
</div>
<p id="demo"></p>
```

```
<script>
function myfnc() {
  const element = document.getElementById("myDIV");
  let x = element.scrollLeft;
  let y = element.scrollTop;
  document.getElementById ("demo").innerHTML = "Horizontally: " + x.toFixed() +
  "<br>Vertically: " + y.toFixed();
}
</script>
```

ex)

```
<script>
    window.onscroll = function () {
        let ht = document.documentElement.scrollTop;
        console.log(ht);
        if (document.documentElement.scrollTop > 20) {
            document.getElementById("navbar").style.top = "0";
        } else {
            document.getElementById("navbar").style.top = "-50px";
        }
    };
</script>
```

## onmousemove Event

onmousemove 이벤트는 포인터가 요소 위에 있는 동안 포인터가 움직일 때 발생합니다.

In HTML:

```
<element onmousemove="myScript">
```

Try it Yourself »

In JavaScript:

```
object.onmousemove = function(){myScript};
```

**clientX, clientY** 속성은 브라우저가 기준인 좌표입니다. 현재 보이는 브라우저 화면 상에서 어느 지점에 위치하는 지를 의미하기 때문에 스크롤 해도 값은 변하지 않습니다.

**clientX** : 브라우저 페이지에서의 X좌표 위치를 반환합니다.

**clientY** : 브라우저 페이지에서의 Y좌표 위치를 반환합니다.

**pageX, pageY** 속성은 문서가 기준입니다. client와 비슷하지만 이 메서드는 문서 전체 크기가 기준이라 스크롤 시 값이 바뀝니다. (스크롤을 포함해서 측정합니다)

**pageX** : 브라우저 페이지에서의 x좌표 위치를 반환합니다.

**pageY** : 브라우저 페이지에서의 Y좌표 위치를 반환합니다.

**screenX, screenY** 속성은 화면 출력 크기(자신의 모니터)가 기준인 절대 좌표입니다. 브라우저를 움직여도 값은 같습니다.

**screenX** : 전체 모니터 스크린에서의 x좌표 위치를 반환합니다.

**screenY** : 전체 모니터 스크린에서의 y좌표 위치를 반환합니다.

## preventDefault() Event Method

preventDefault() 메서드는 이벤트가 취소 가능한 경우 이벤트를 취소합니다.

즉, 이벤트에 속한 기본 작업이 발생하지 않습니다.

"제출" 버튼을 클릭하면 양식이 제출되지 않습니다.

링크를 클릭하면 링크가 URL을 따라가지 않도록 방지합니다.

## The onload and onunload Events

onload 및 onunload 이벤트는 사용자가 페이지에 들어가거나 나갈 때 트리거됩니다.

onload 이벤트는 방문자의 브라우저 유형 및 브라우저 버전을 확인하고 정보를 기반으로 웹 페이지의 적절한 버전을 로드하는 데 사용할 수 있습니다.

ex)

```
<script>
function mymessage() {
    alert("This message was triggered from the onload event");
}
</script>
</head>

<body onload="mymessage()">
</body>
```

## The onchange Event

onchange 이벤트는 종종 입력 필드의 유효성 검사와 함께 사용됩니다.

다음은 사용자가 입력 필드의 내용을 변경할 때 upperCase() 함수가 호출됩니다.

ex)

```
<input type="text" id="fname" onchange="upperCase()">
<script>
function upperCase() {
    const x = document.getElementById("fname");
    x.value = x.value.toUpperCase();
}
</script>
```

## Window Size

두 가지 속성을 사용하여 브라우저 창의 크기를 결정할 수 있습니다.

두 속성 모두 크기를 픽셀 단위로 반환합니다.

window.innerHeight - 브라우저 창의 내부 높이(픽셀 단위)

window.innerWidth - 브라우저 창의 내부 너비(픽셀 단위)

ex)

```
<p id="demo"> </p>
<script>
document.getElementById("demo").innerHTML =
"Browser inner window width: " + window.innerWidth + "px<br>" +
"Browser inner window height: " + window.innerHeight + "px";
</script>
```

## Set Text Content

```
<p id="demo" onclick="myfnc()">demo</p>
<script>
function myfnc() {
    document.getElementById("demo").textContent = "I have changed!";
}
</script>
```

