

# ブラウザの座標関係

## マウスイベントの座標系

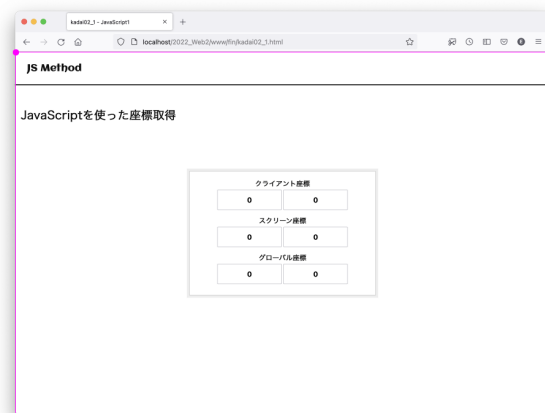
JavaScript を利用することで、マウスイベントのブラウザやモニター、HTML要素からの各種座標を取得することができます。

座標は基準からみての距離を数値（ピクセル単位）で取得するので、どの対象基準から見ての座標が欲しいかが重要になります。ここでは、媒体別にクライアント座標・スクリーン座標・グローバル座標の3つを紹介します。

## クライアント座標

ブラウザの表示可能領域の左上を基準点（x : 0 y : 0）とした座標のことを**クライアント座標**といいます。

JavaScript でマウスイベントのクライアント座標は、clientX プロパティとclientY プロパティに保存されています。



### *event.clientX*

clientXは、マウスイベントが発生したブラウザ上のX座標が格納されています。

clientXは読み取り専用プロパティとなります。


### *event.clientY*


clientYは、マウスイベントが発生したブラウザ上のY座標が格納されています。

clientYは読み取り専用プロパティとなります。

#### MouseEvent.clientX - Web API | MDN


clientX は MouseEvent の読み取り専用のプロパティで、このイベントが発生した時点のアプリケーションのビューポートに

 <https://developer.mozilla.org/ja/docs/Web/API/MouseEvent/clientX>

 mdn web docs

#### MouseEvent.clientY - Web API | MDN

clientY は MouseEvent の読み取り専用のプロパティで、このイベントが発生した時点のアプリケーションのビューポートに

 <https://developer.mozilla.org/ja/docs/Web/API/MouseEvent/clientY>

 mdn web docs

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <title>Web演習 2 - サンプル 5</title>
  <style>
    body{height:500vh;}
  </style>
</head>
<body>
  <h1>マウスイベントの座標</h1>
  <h2>クライアント座標</h2>
  <p id="client">x : y</p>

  <h2>スクリーン座標</h2>
  <p id="screen">x : y</p>

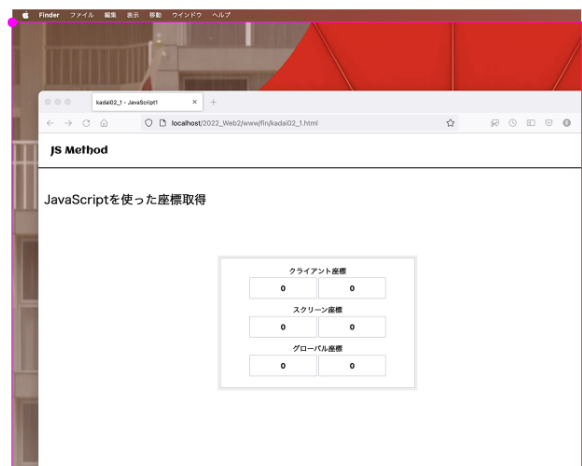
  <h2>グローバル座標</h2>
  <p id="global">x : y</p>

  <script src="sample05.js"></script>
</body>
</html>
```

```
// document mouseover event
document.addEventListener('mousemove', (event)=>{
  // client position
  document.querySelector('#client').innerText = `${ event.clientX } : ${ event.clientY }`;
});
```

## スクリーン座標

モニターの左上を基準点（x : 0 y : 0）とした座標のことをいいます。



### *event.screenX*


screenXは、マウスイベントが発生したモニター上のX座標が格納されています。  
screenXは読み取り専用プロパティとなります。


### *event.screenY*

screenYは、マウスイベントが発生したモニター上のY座標が格納されています。  
screenYは読み取り専用プロパティとなります。

#### MouseEvent.screenX - Web API | MDN


screenX は MouseEvent インターフェイスの読み取り専用プロパティで、グローバル（スクリーン）座標における、マウスポイ

 <https://developer.mozilla.org/ja/docs/Web/API/MouseEvent/screenX>

 mdn web docs

#### MouseEvent.screenY - Web API | MDN

screenY は MouseEvent インターフェイスの読み取り専用プロパティで、グローバル（スクリーン）座標における、マウスポイ

 <https://developer.mozilla.org/ja/docs/Web/API/MouseEvent/screenY>

 mdn web docs

```
// document mouseover event
document.addEventListener('mousemove', (event)=>{
  // client position
  document.querySelector('#client').innerText = `${ event.clientX } : ${ event.clientY }`;
  // screen position
  document.querySelector('#screen').innerText = `${ event.screenX } : ${ event.screenY }`;
});
```

## グローバル座標

Webページ全体の左上を基準点（x : 0 y : 0）とした座標のことをいいます。

Webページ全体の座標（グローバル座標）を取得するには、「ブラウザのスクロール量」と「クライアント座標」を加算することで、Webページ全体から見た座標を算出することができます。

JavaScriptでブラウザのスクロール量の数値を取得するには、**document.documentElement** の **scrollTop** プロパティ（垂直方向）と **scrollLeft** プロパティ（水平方向）があります。


### *document.documentElement*

documentElementは、documentのルート要素（HTMLであれば、html要素）であるElementを返します。

#### Document.documentElement - Web API | MDN

Document.documentElement は、その document のルート要素 (例えば、HTML 文書の場合は <html> 要素) である Element を返します。

 <https://developer.mozilla.org/ja/docs/Web/API/Document/documentElement>

 mdn web docs

### *Element.scrollTop*


要素の内容が垂直方向にスクロールするピクセル数を取得または設定します。

### *Element.scrollLeft*

要素の内容が左端からスクロールするピクセル数を取得または設定します。

#### Element.scrollTop - Web API | MDN


Element.scrollTop プロパティは、要素の内容が垂直にスクロールするピクセル数を取得または設定します。

 <https://developer.mozilla.org/ja/docs/Web/API/Element/scrollTop>

 mdn web docs

#### Element.scrollLeft - Web API | MDN

Element.scrollLeft プロパティは、要素の内容が左端からスクロールするピクセル数を取得または設定します。

 <https://developer.mozilla.org/ja/docs/Web/API/Element/scrollLeft>

 mdn web docs

```
// document mouseover event
document.addEventListener('mousemove', (event)=>{
  // client position
  document.querySelector('#client').innerText = `${ event.clientX } : ${ event.clientY }`;
  // screen position
  document.querySelector('#screen').innerText = `${ event.screenX } : ${ event.screenY }`;
  // global position
  document.querySelector('#global').innerText = `${ document.documentElement.scrollLeft + event.clientX } : ${ document.documentElement.scrollLeft }`;
});
```

## 要素の座標

直近の親要素左上を基準とした要素の距離（座標）を用いることで、Webページ上の座標を取得することができます。

### *Element.offsetTop*


親要素の上端に対して現在の要素の距離を返します。


### *Element.offsetLeft*

現在の要素の左上隅が、親要素内で左へオフセットされるピクセル数を返します。

#### HTMLElement.offsetTop - Web API | MDN


HTMLElement.offsetTop は読み取り専用プロパティで、現在の要素の外枠から、最も近い位置にある祖先要素である offsetParent の上枠の内


 <https://developer.mozilla.org/ja/docs/Web/API/HTMLElement/offsetTop>

 mdn web docs

#### HTMLElement.offsetTop - Web API | MDN

HTMLElement.offsetTop は読み取り専用プロパティで、現在の要素の外枠から、最も近い位置にある祖先要素である offsetParent の上枠の内

 <https://developer.mozilla.org/ja/docs/Web/API/HTMLElement/offsetTop>

 mdn web docs



対象要素がpositionプロパティのfixedに設定されている場合は、親要素からみた距離による座標は正しく取得することができません。

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Web演習 2 - サンプル 6</title>
<style>
  body{border:2px solid orange;}
  .line{border-top:1px solid tomato;border-left:1px solid tomato;}
</style>
</head>
<body>
<h1>タグの座標</h1>
<section id="sec01" class="line">
  <h2>セクション 1</h2>
  <p>セクション 1 の座標</p>
  <p class="pos">x : y</p>
</section>

  <section id="sec02" class="line">
    <h2>セクション 2</h2>
    <p>セクション 2 の座標</p>
    <p class="pos">x : y</p>
  </section>

  <script src="sample06.js"></script>
</body>
</html>

```

```

{
  // #sec01 click event
  document.querySelector('#sec01').addEventListener('click', (event)=>{
    document.querySelector('#sec01 > .pos').innerText = `${ event.currentTarget.offsetLeft } : ${ event.currentTarget.offsetTop }`;
  });

  // #sec02 click event
  document.querySelector('#sec02').addEventListener('click', (event)=>{
    document.querySelector('#sec02 > .pos').innerText = `${ event.currentTarget.offsetLeft } : ${ event.currentTarget.offsetTop }`;
  });
}

```