

Координаты на странице

события, продолжение



Клиентские координаты

Каждое событие происходит в конкретном месте области просмотра, при этом координаты указателя мыши сохраняются в свойствах `clientX` / `clientY` объекта `event`, которые поддерживаются во всех браузерах.

```
(clientX,clientY) = (0,0)
(pageX, pageY) = (0,0)
```

W3 Standards - W3C

www.w3.org/standards/

Views: desktop mobile print

STANDARDS PARTICIPATE MEMBERSHIP ABOUT W3C

Google™

W3C » Standards Skip

STANDARDS

W3C standards define an **Open Web Platform** for application development that has the unprecedented potential to enable developers to build rich interactive experiences, powered by vast data stores, that are available on any device. Although the boundaries of the platform continue to evolve, industry leaders speak

VIEW BY

Technology topic

W3C

- Standards
- Participate
- Membership
- About W3C

Пример:

допустим, нам нужно получить координаты щелчка при клике на кнопку:

```
var x = document.getElementById('but');  
  
x.addEventListener('click', showCoordinates);  
  
function showCoordinates(event) {  
    alert('you click in ' + event.clientX + 'x' + event.clientY);  
}
```

Страничные координаты

В то время как клиентские координаты указывают на место возникновения события в области просмотра, страничные координаты определяют это место на странице. Они представлены свойствами `pageX` / `pageY` и рассчитываются относительно левого и верхнего краев самой страницы, а не области просмотра.

Если страница не прокручивалась, страничные координаты будут совпадать с клиентскими.

Размер прокрутки

Помимо позиции элемента относительно области просмотра либо всей страницы, мы можем получить размер проскроленной части при помощи свойств `pageYOffset/pageXOffset`.

Пример:

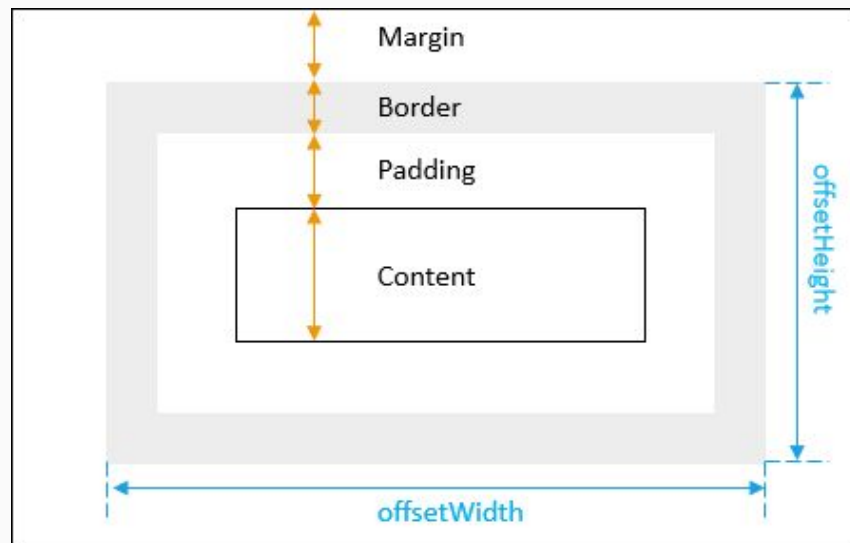
допустим, нам нужно рассчитать, на сколько мы прокрутили страницу в момент клика по кнопке.

Для этого можно поступить так:

```
var x = document.getElementById('but');  
  
x.addEventListener('click', showCoordinates);  
  
function showCoordinates(event) {  
  
    var scrollNow = window.pageYOffset;  
  
    alert('you click in ' + scrollNow);  
  
}
```

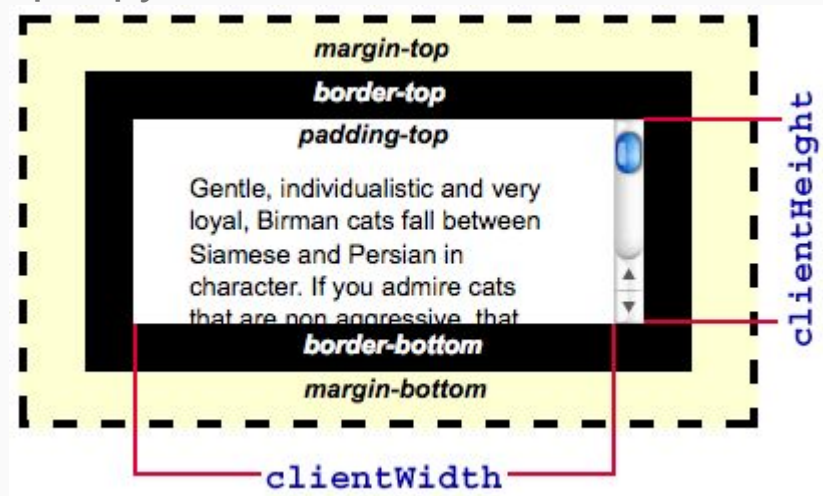
offsetWidth/Height

Они содержат «внешнюю» ширину/высоту элемента, то есть его полный размер, включая рамки.



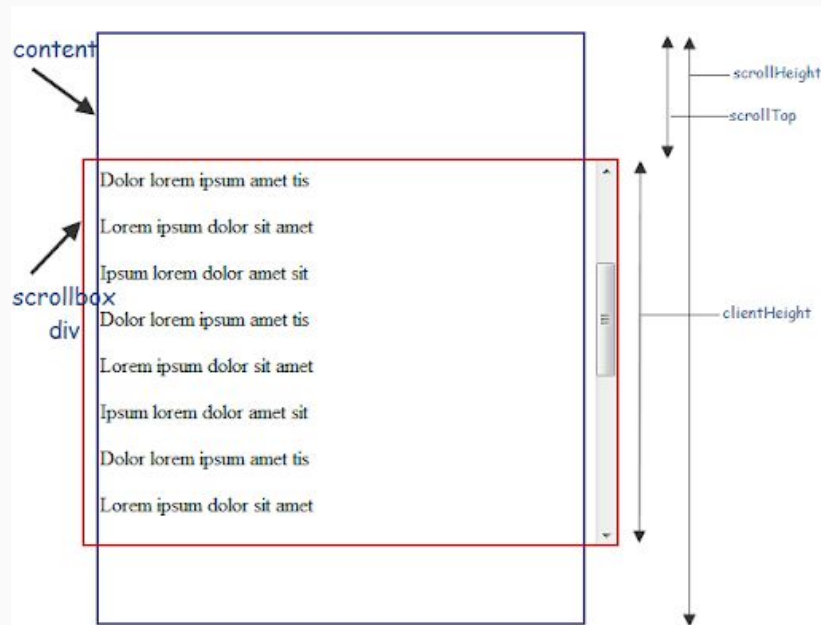
clientWidth/Height

Они включают в себя ширину области содержимого вместе с внутренними отступами padding, но без прокрутки



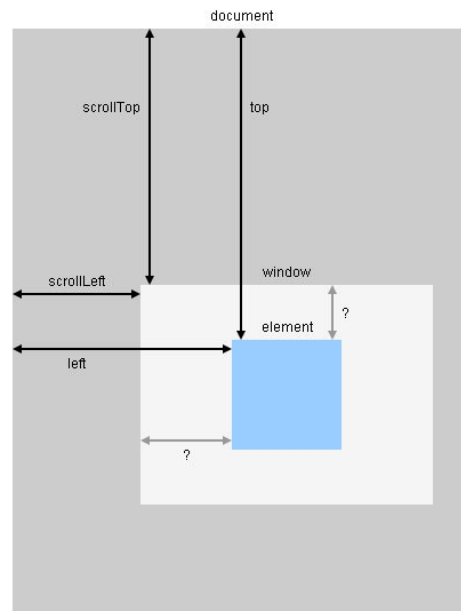
scrollWidth/Height

Эти свойства – как `clientWidth/clientHeight`, но также включают в себя прокрученную (которую не видно) часть элемента



scrollLeft/scrollTop

Свойства scrollLeft/scrollTop – ширина/высота невидимой, прокрученной в данный момент, части элемента слева и сверху.



Задание 1:

Выводить значения скрола с верху страницы на событие прокрутки страницы.

Задание 2:

При скроле, в момент активного 2-го экрана покажите фиксированную кнопку внизу страницы, которая до этого будет являться скрытой.

Прокрутка: `scrollTo`, `scrollBy`, `scrollIntoView`

- Метод `scrollBy(x,y)` прокручивает страницу относительно её текущего положения. Например, `scrollBy(0,10)` прокручивает страницу на 10px вниз.
- - Метод `scrollTo(pageX,pageY)` прокручивает страницу на абсолютные координаты (`pageX,pageY`). То есть, чтобы левый-верхний угол видимой части страницы имел данные координаты относительно левого верхнего угла документа. Это всё равно, что поставить `scrollLeft/scrollTop`. Для прокрутки в самое начало мы можем использовать `scrollTo(0,0)`.

scrollIntoView

Вызов `elem.scrollIntoView(top)` прокручивает страницу, чтобы `elem` оказался вверху. У него есть один аргумент:

- если `top=true` (по умолчанию), то страница будет прокручена, чтобы `elem` появился в верхней части окна. Верхний край элемента совмещён с верхней частью окна.
- если `top=false`, то страница будет прокручена, чтобы `elem` появился внизу. Нижний край элемента будет совмещён с нижним краем окна.

Чтобы запретить прокрутку страницы, достаточно установить `document.body.style.overflow = "hidden"`.

Задание 3:

Создать страницу с прокруткой. В конце страницы создать кнопку “Перейти в начало страницы”, которая скролит страницу в самое начало.

В середине страницы создать кнопку, по нажатию на которую, страница прокручивается так, чтобы эта кнопка была сверху видимой области.

Задание 4:

Реализовать TABS-блок.

При клике по вкладке, должен показываться соответствующий текст:



Задание 5*:

Реализовать подчёркивание пункта меню, соответствующего текущему экрану.

Например, если мы находимся на разделе “контакты”, вкладка контакты в фиксированном меню должна быть активной и т.д.