**[Найти размер прокрутки снизу](https://learn.javascript.ru/size-and-scroll" \l "nayti-razmer-prokrutki-snizu)**

важность: 5

Свойство elem.scrollTop содержит размер прокрученной области при отсчёте сверху. А как подсчитать размер прокрутки снизу (назовём его scrollBottom)?

Напишите соответствующее выражение для произвольного элемента elem.

P.S. Проверьте: если прокрутки нет вообще или элемент полностью прокручен – оно должно давать 0.

Ответ: let scrollBottom = elem.scrollHeight - elem.scrollTop - elem.clientHeight;

**[Узнать ширину полосы прокрутки](https://learn.javascript.ru/size-and-scroll" \l "uznat-shirinu-polosy-prokrutki)**

важность: 3

Напишите код, который возвращает ширину стандартной полосы прокрутки.

Для Windows она обычно колеблется от 12px до 20px. Если браузер не выделяет место под полосу прокрутки (так тоже бывает, она может быть прозрачной над текстом), тогда значение может быть 0px.

P.S. Ваш код должен работать в любом HTML-документе, независимо от его содержимого.

let div = document.createElement('div');

        div.style.overflowY = 'scroll';

        div.style.width = '50px';

        div.style.height = '50px';

        document.body.append(div);

        let scrollWidth = div.offsetWidth - div.clientWidth;

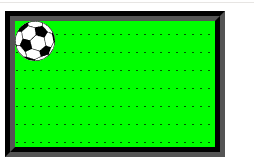
        div.remove();

        alert(scrollWidth);

### [Поместите мяч в центр поля](https://learn.javascript.ru/size-and-scroll" \l "pomestite-myach-v-tsentr-polya)

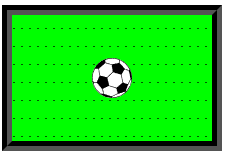
важность: 5

Исходный документ выглядит так:



Каковы координаты центра поля?

Вычислите их и используйте, чтобы поместить мяч в центр поля:



* Элемент должен позиционироваться за счёт JavaScript, а не CSS.
* Код должен работать с любым размером мяча (10, 20, 30 пикселей) и любым размером поля без привязки к исходным значениям.

P.S. Да, центрирование можно сделать при помощи чистого CSS, но задача именно на JavaScript. Далее будут другие темы и более сложные ситуации, когда JavaScript будет уже точно необходим, это – своего рода «разминка».

let ball\_posY = (field.clientHeight - ball.clientHeight) / 2;

        let ball\_posX = (field.clientWidth - ball.clientWidth) / 2;

        ball.style.top = `${ball\_posY}px`;

        ball.style.left = `${ball\_posX}px`;

**[В чём отличие CSS-свойств width и clientWidth](https://learn.javascript.ru/size-and-scroll" \l "v-chyom-otlichie-css-svoystv-width-i-clientwidth)**

важность: 5

В чём отличие между getComputedStyle(elem).width и elem.clientWidth?

Укажите хотя бы 3 отличия, лучше – больше.

1. getComputedStyle именно эту ширину мы только можем менять, а свойство clientWidth только для чтения.
2. clientWidth соответствует внутренней области элемента, включая внутренние отступы padding, а CSS-ширина (при стандартном значении box-sizing) соответствует внутренней области без внутренних отступов padding
3. getComputedStyle не всегда даст ширину, он может вернуть, к примеру, "auto" для строчного элемента