**Лабораторная работа №18**

**Тема работы:** Разработка программ с использованием принципа наследования.

**Цель работы:** закрепить навыки по использованию наследования в классах.

**Теоретические сведения**

**Размеры и прокрутка окна**

Как узнать ширину и высоту окна браузера? Как получить полную ширину и высоту документа, включая прокрученную часть? Как прокрутить страницу с помощью JavaScript?

Для большинства таких запросов мы можем использовать корневой элемент документа document.documentElement, который соответствует тегу <html>. Однако есть дополнительные методы и особенности, которые необходимо учитывать.

**[Ширина/высота окна](https://learn.javascript.ru/size-and-scroll-window" \l "shirina-vysota-okna)**

Чтобы получить ширину/высоту окна, можно взять свойства clientWidth/clientHeight из document.documentElement:

Например, эта кнопка показывает высоту вашего окна:

alert(document.documentElement.clientHeight)

**Не window.innerWidth/Height**

Браузеры также поддерживают свойства window.innerWidth/innerHeight. Вроде бы, похоже на то, что нам нужно. Почему же не использовать их?

Если есть полоса прокрутки, и она занимает какое-то место, то свойства clientWidth/clientHeight указывают на ширину/высоту документа без неё (за её вычетом). Иными словами, они возвращают высоту/ширину видимой части документа, доступной для содержимого.

А window.innerWidth/innerHeight включают в себя полосу прокрутки.

Если полоса прокрутки занимает некоторое место, то эти две строки выведут разные значения:

alert( window.innerWidth ); // полная ширина окна

alert( document.documentElement.clientWidth ); // ширина окна за вычетом полосы прокрутки

В большинстве случаев нам нужна доступная ширина окна: для рисования или позиционирования. Полоса прокрутки «отъедает» её часть. Поэтому следует использовать documentElement.clientHeight/Width.

**DOCTYPE – это важно**

Обратите внимание, что геометрические свойства верхнего уровня могут работать немного иначе, если в HTML нет <!DOCTYPE HTML>. Возможны странности.

В современном HTML мы всегда должны указывать DOCTYPE.

**[Ширина/высота документа](https://learn.javascript.ru/size-and-scroll-window" \l "shirina-vysota-dokumenta)**

Теоретически, т.к. корневым элементом документа является documentElement, и он включает в себя всё содержимое, мы можем получить полный размер документа как documentElement.scrollWidth/scrollHeight.

Но именно на этом элементе, для страницы в целом, эти свойства работают не так, как предполагается. В Chrome/Safari/Opera, если нет прокрутки, то documentElement.scrollHeight может быть даже меньше, чем documentElement.clientHeight! С точки зрения элемента это невозможная ситуация.

Чтобы надёжно получить полную высоту документа, нам следует взять максимальное из этих свойств:

let scrollHeight = Math.max(

document.body.scrollHeight, document.documentElement.scrollHeight,

document.body.offsetHeight, document.documentElement.offsetHeight,

document.body.clientHeight, document.documentElement.clientHeight

);

alert('Полная высота документа с прокручиваемой частью: ' + scrollHeight);

Почему? Лучше не спрашивайте. Эти несоответствия идут с древних времён. Глубокой логики здесь нет.

**[Получение текущей прокрутки](https://learn.javascript.ru/size-and-scroll-window" \l "page-scroll)**

Обычные элементы хранят текущее состояние прокрутки в elem.scrollLeft/scrollTop.

Что же со страницей? В большинстве браузеров мы можем обратиться к documentElement.scrollLeft/Top, за исключением основанных на старом WebKit (Safari), где есть ошибка ([5991](https://bugs.webkit.org/show_bug.cgi?id=5991)), и там нужно использовать document.body вместо document.documentElement.

К счастью, нам совсем не обязательно запоминать эти особенности, потому что текущую прокрутку можно прочитать из свойств window.pageXOffset/pageYOffset:

alert('Текущая прокрутка сверху: ' + window.pageYOffset);

alert('Текущая прокрутка слева: ' + window.pageXOffset);

Эти свойства доступны только для чтения.

**[Прокрутка: scrollTo, scrollBy, scrollIntoView](https://learn.javascript.ru/size-and-scroll-window" \l "window-scroll)**

**Важно:**

Для прокрутки страницы из JavaScript её DOM должен быть полностью построен.

Например, если мы попытаемся прокрутить страницу из скрипта в <head>, это не сработает.

Обычные элементы можно прокручивать, изменяя scrollTop/scrollLeft.

Мы можем сделать то же самое для страницы в целом, используя document.documentElement.scrollTop/Left (кроме основанных на старом WebKit (Safari), где, как сказано выше, document.body.scrollTop/Left).

Есть и другие способы, в которых подобных несовместимостей нет: специальные методы window.scrollBy(x,y) и window.scrollTo(pageX,pageY).

* Метод scrollBy(x,y) прокручивает страницу относительно её текущего положения. Например, scrollBy(0,10) прокручивает страницу на 10px вниз.

Кнопка ниже демонстрирует это:

window.scrollBy(0,10)

* 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).

window.scrollTo(0,0)

Эти методы одинаково работают для всех браузеров.

**[scrollIntoView](https://learn.javascript.ru/size-and-scroll-window" \l "scrollintoview)**

Для полноты картины давайте рассмотрим ещё один метод: [elem.scrollIntoView(top)](https://developer.mozilla.org/ru/docs/Web/API/Element/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.

Кнопка ниже прокрутит страницу так, что она сама окажется вверху:

this.scrollIntoView()

А следующая кнопка прокрутит страницу так, что она сама окажется внизу

this.scrollIntoView(false)

**[Запретить прокрутку](https://learn.javascript.ru/size-and-scroll-window" \l "zapretit-prokrutku)**

Иногда нам нужно сделать документ «непрокручиваемым». Например, при показе большого диалогового окна над документом – чтобы посетитель мог прокручивать это окно, но не документ.

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

Попробуйте сами:

document.body.style.overflow = 'hidden'

document.body.style.overflow = ''

Первая кнопка останавливает прокрутку, вторая возобновляет её.

Аналогичным образом мы можем «заморозить» прокрутку для других элементов, а не только для document.body.

Недостатком этого способа является то, что сама полоса прокрутки исчезает. Если она занимала некоторую ширину, то теперь эта ширина освободится, и содержимое страницы расширится, текст «прыгнет», заняв освободившееся место.

Это выглядит немного странно, но это можно обойти, если сравнить clientWidth до и после остановки, и если clientWidth увеличится (значит полоса прокрутки исчезла), то добавить padding в document.body вместо полосы прокрутки, чтобы оставить ширину содержимого прежней.

**Порядок выполнения работы**

Дана кнопка. По нажатию на эту кнопку выведите высоту окна браузера.

 Дана кнопка. По нажатию на эту кнопку прокрутите окно браузера до самого низа.

 Дана кнопка. По нажатию на эту кнопку узнайте, есть ли у окна браузера вертикальная прокрутка.