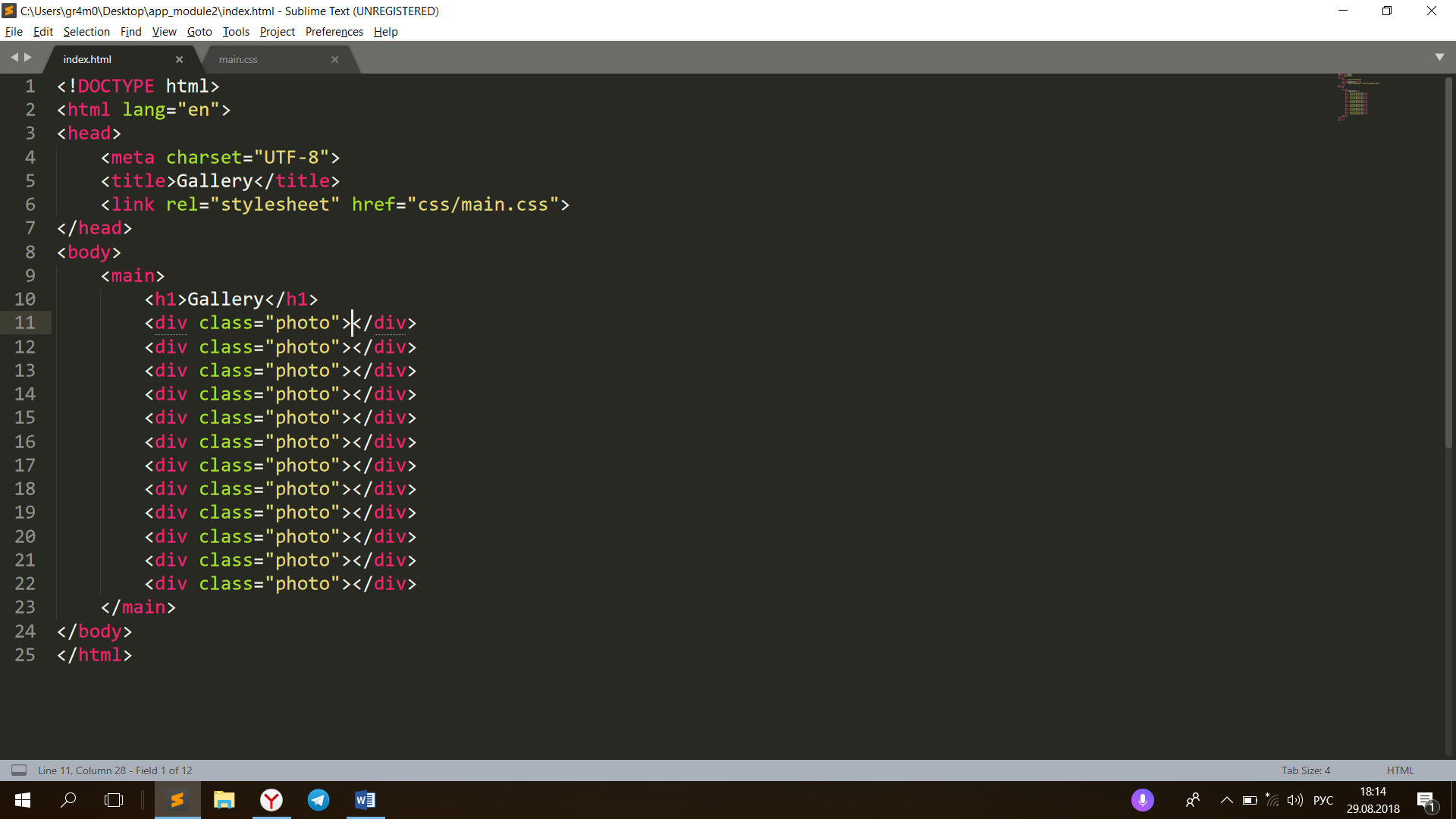
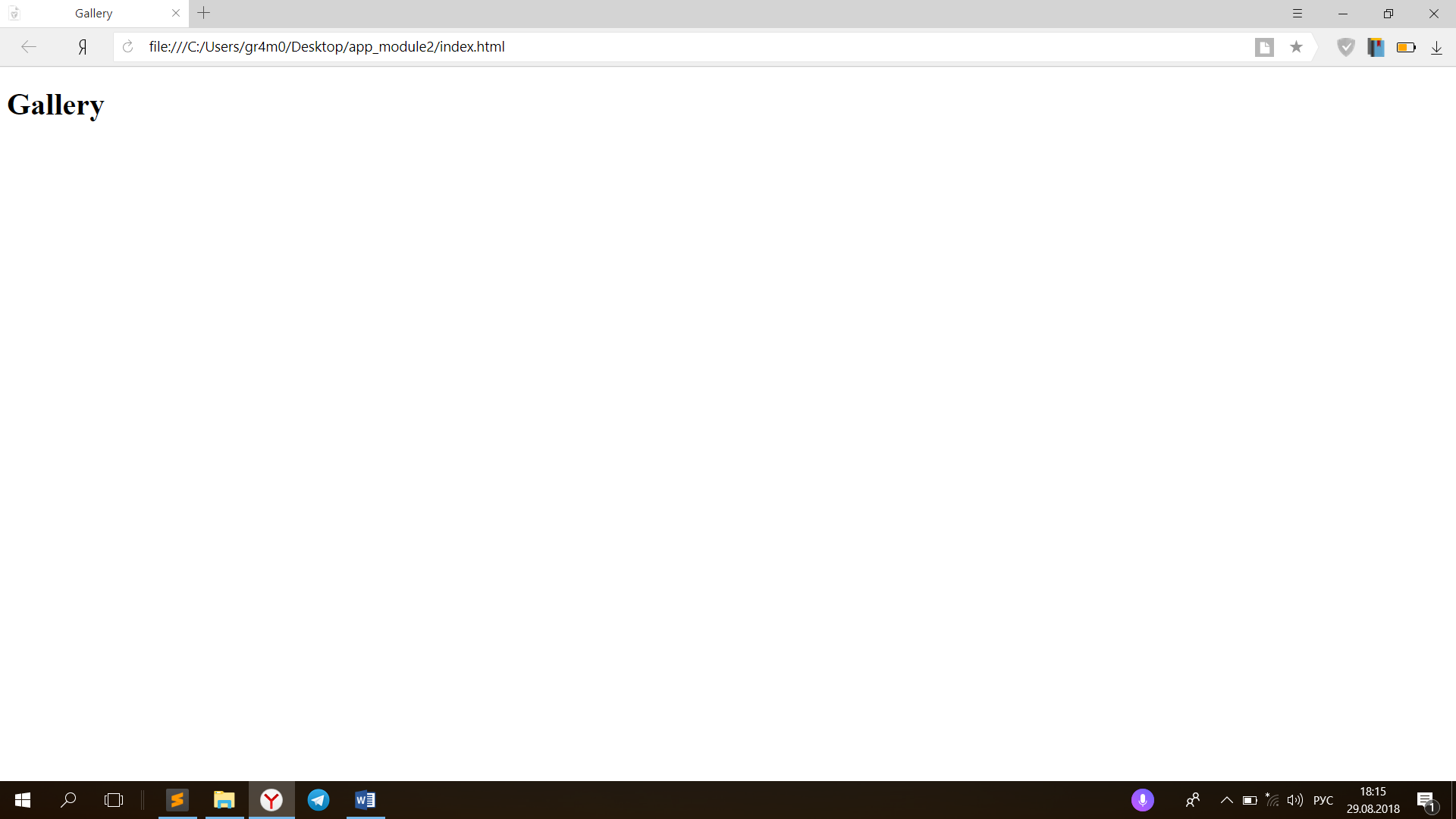
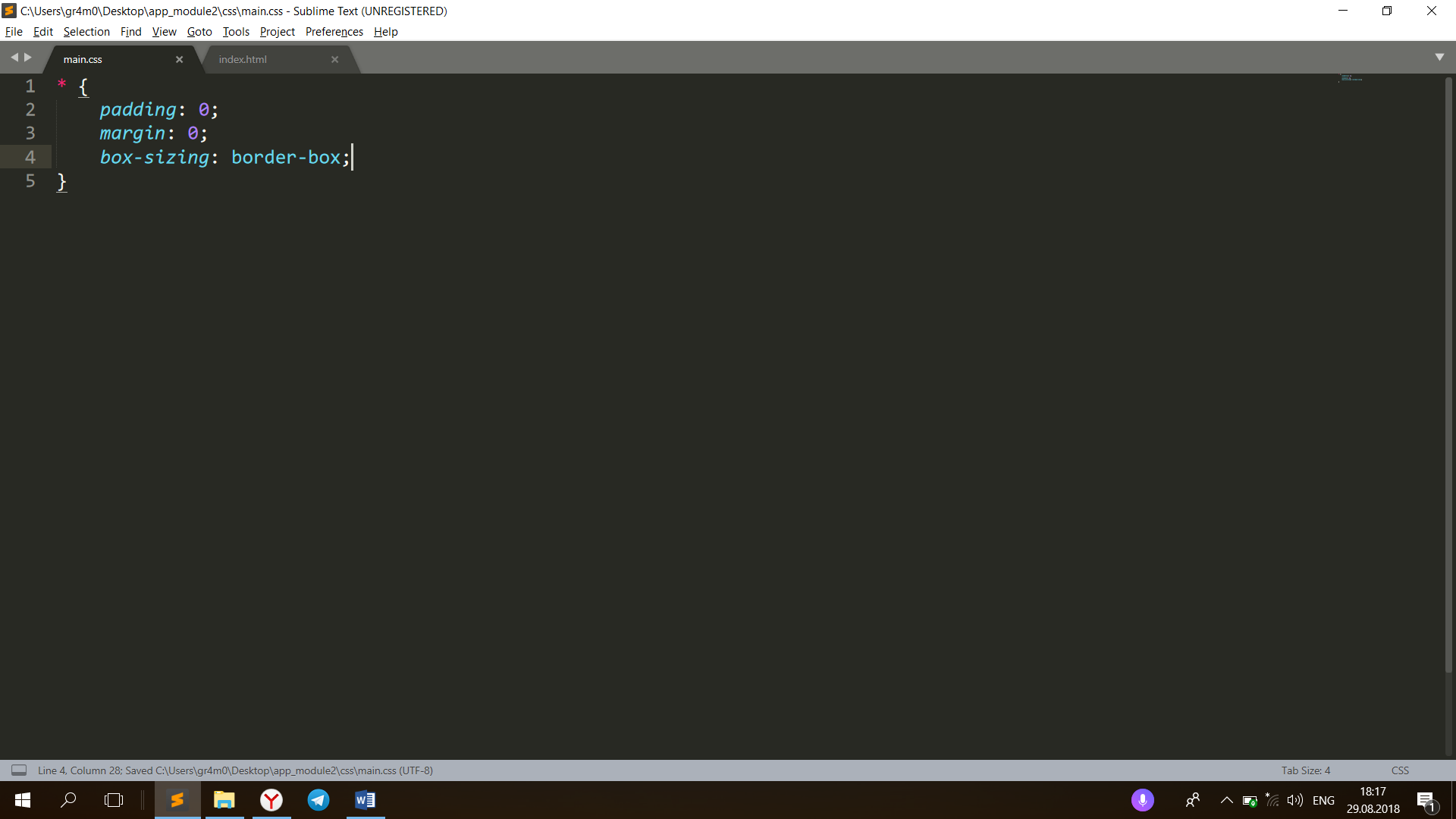
**Курс веб-разработке**

**Второй модуль**

**Второе занятие**

На предыдущем занятии мы создали следующую разметку: 

Наша веб-страница в данный выглядит так: xt 

Приступим к написанию css-стилей. Для этого в Sublime Text 3 открываем файл main.css и сбрасываем отступы для всех тегов на странице. 

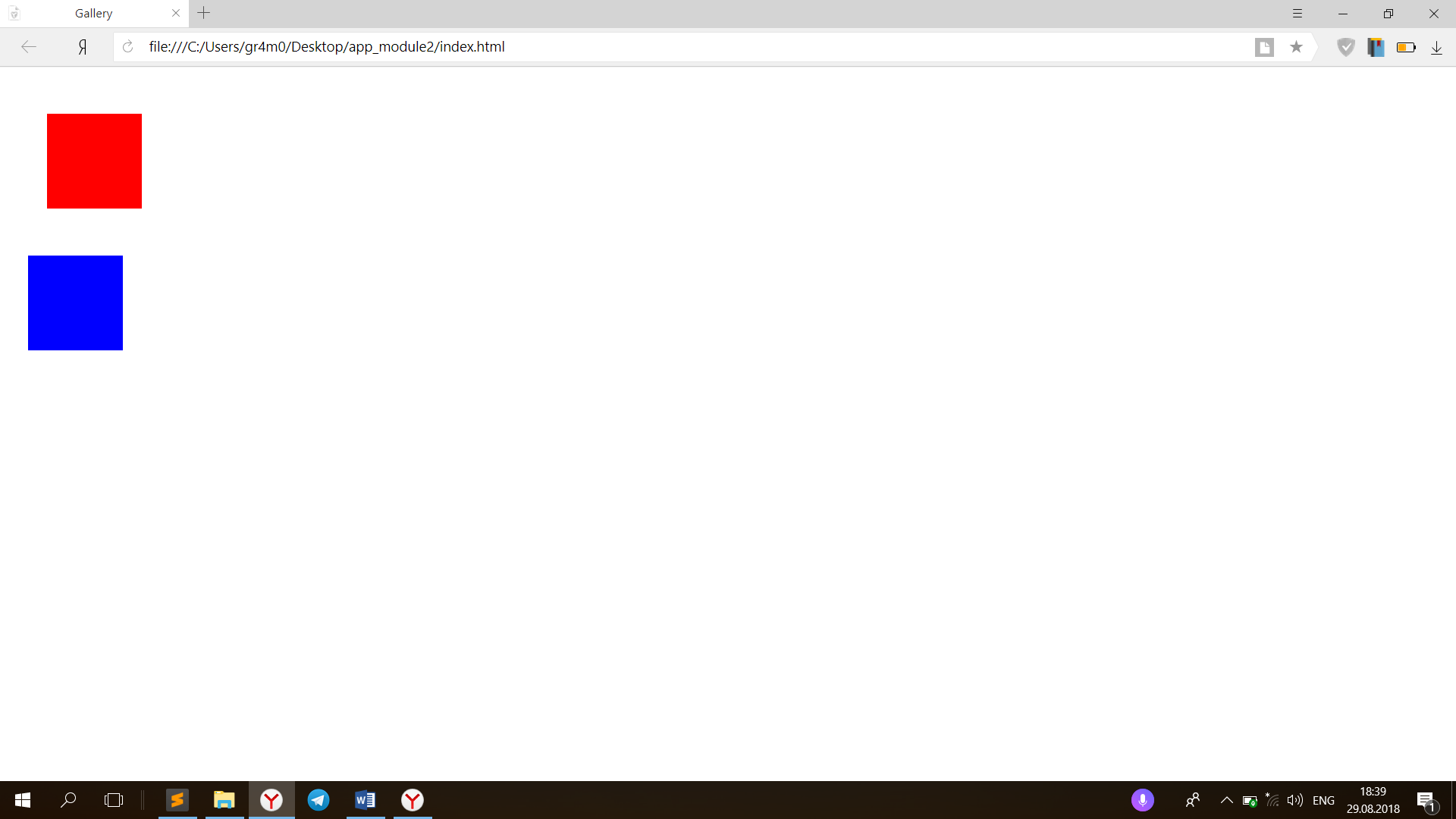
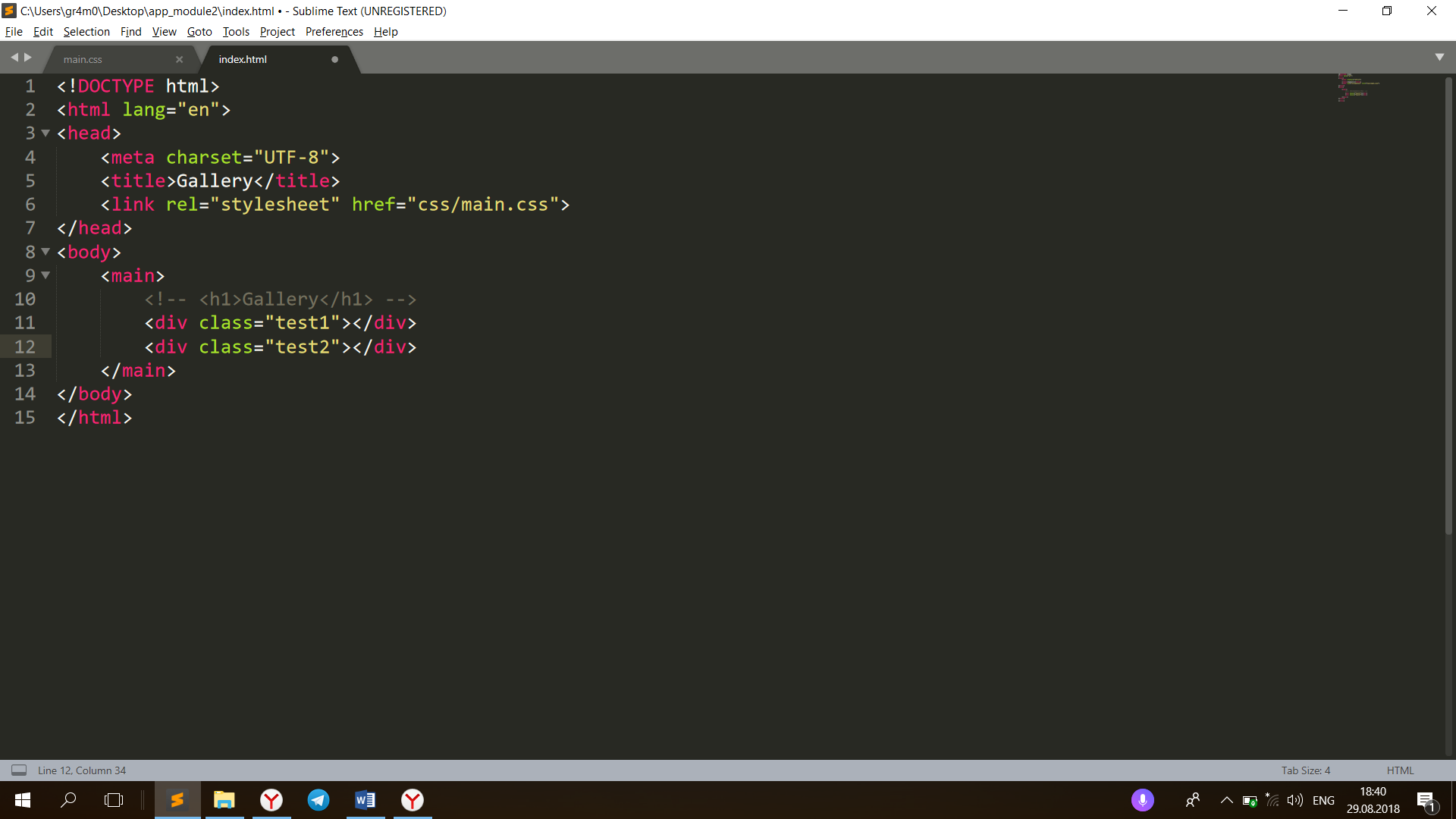
Это делается для того, чтобы все браузеры отображали отступы html-элементов одинаково, именно так, как мы хотим их видеть. Еще раз повторим значения приведенных выше css-свойств:

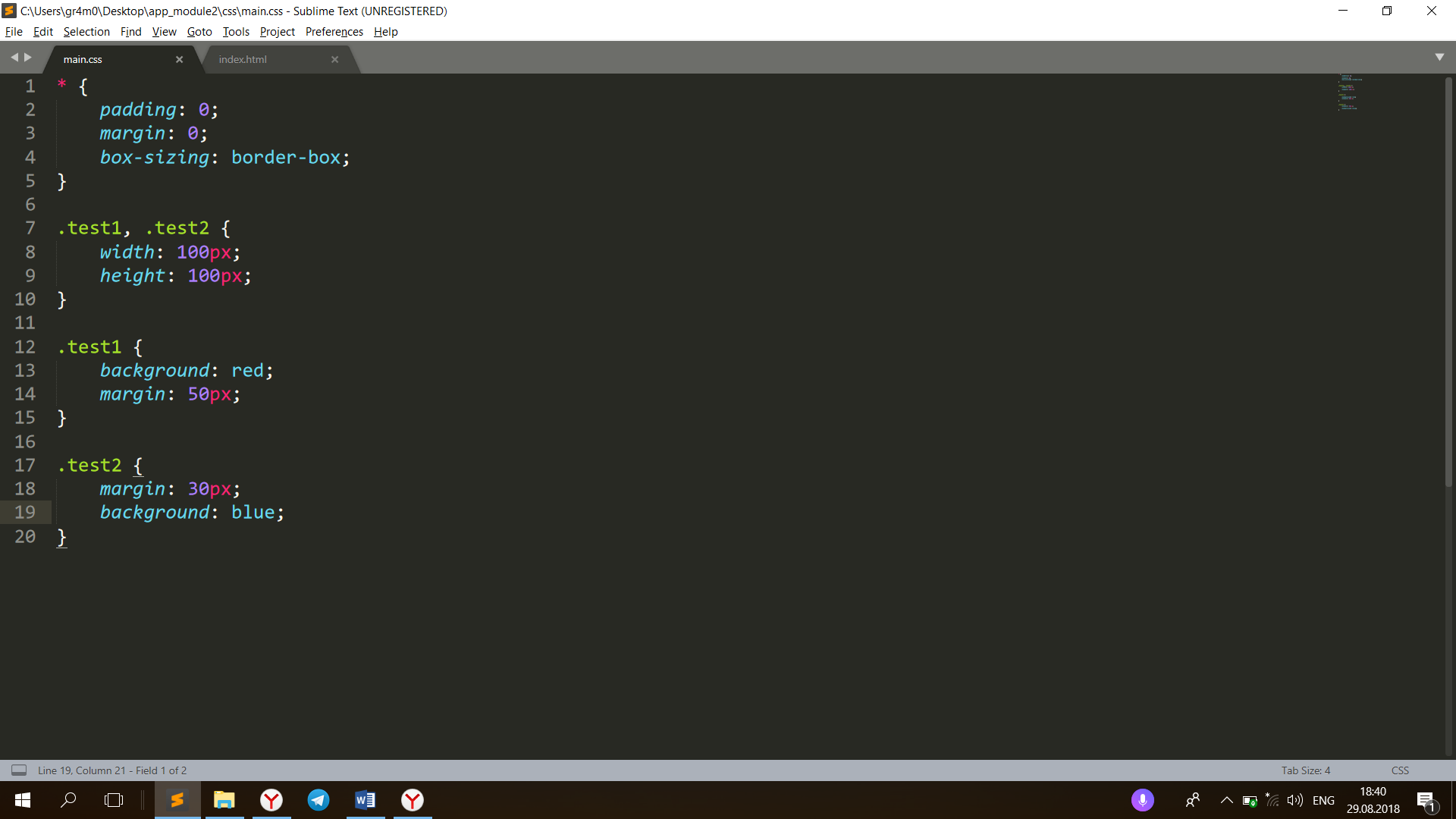
margin — внешний отступ

padding — внутренний отступ

box-sizing — box-модель элемента

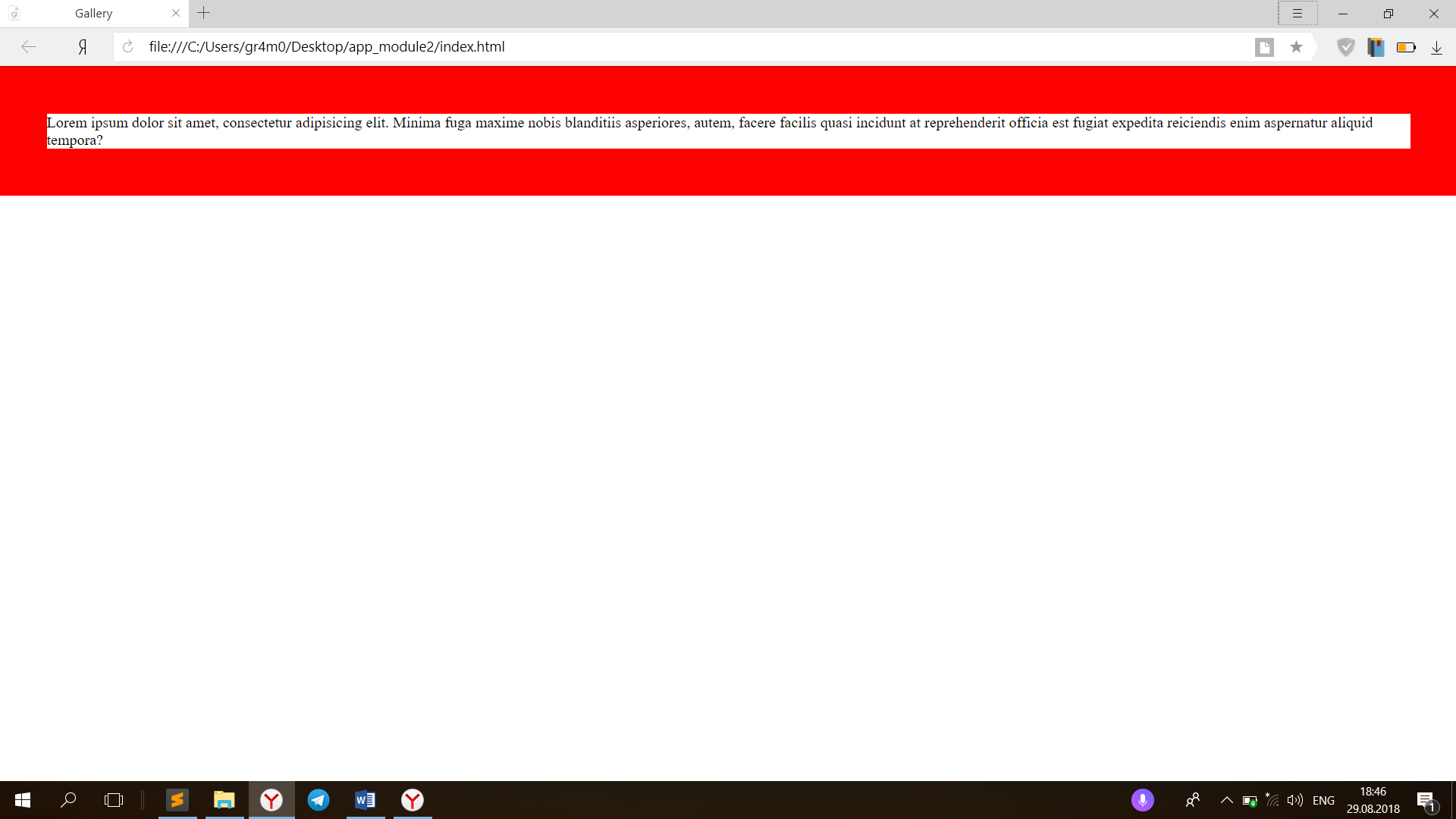
Здесь следует пояснить, чем внутренний отступ отличается от внешнего. Внешний отступ или margin определяет расстояние между двумя соседними элементами, например, у нас есть два тега div, и мы задаем margin: 20px. Таким образом, расстояние между нашими div’ами будет 20px, так как отступы у соседних элементов не суммируется, за отступ берется наибольшее значение свойства margin. Например, если у первого элемента margin равен 50px, а у второго — 30px, то отступ между ними будет 50px.

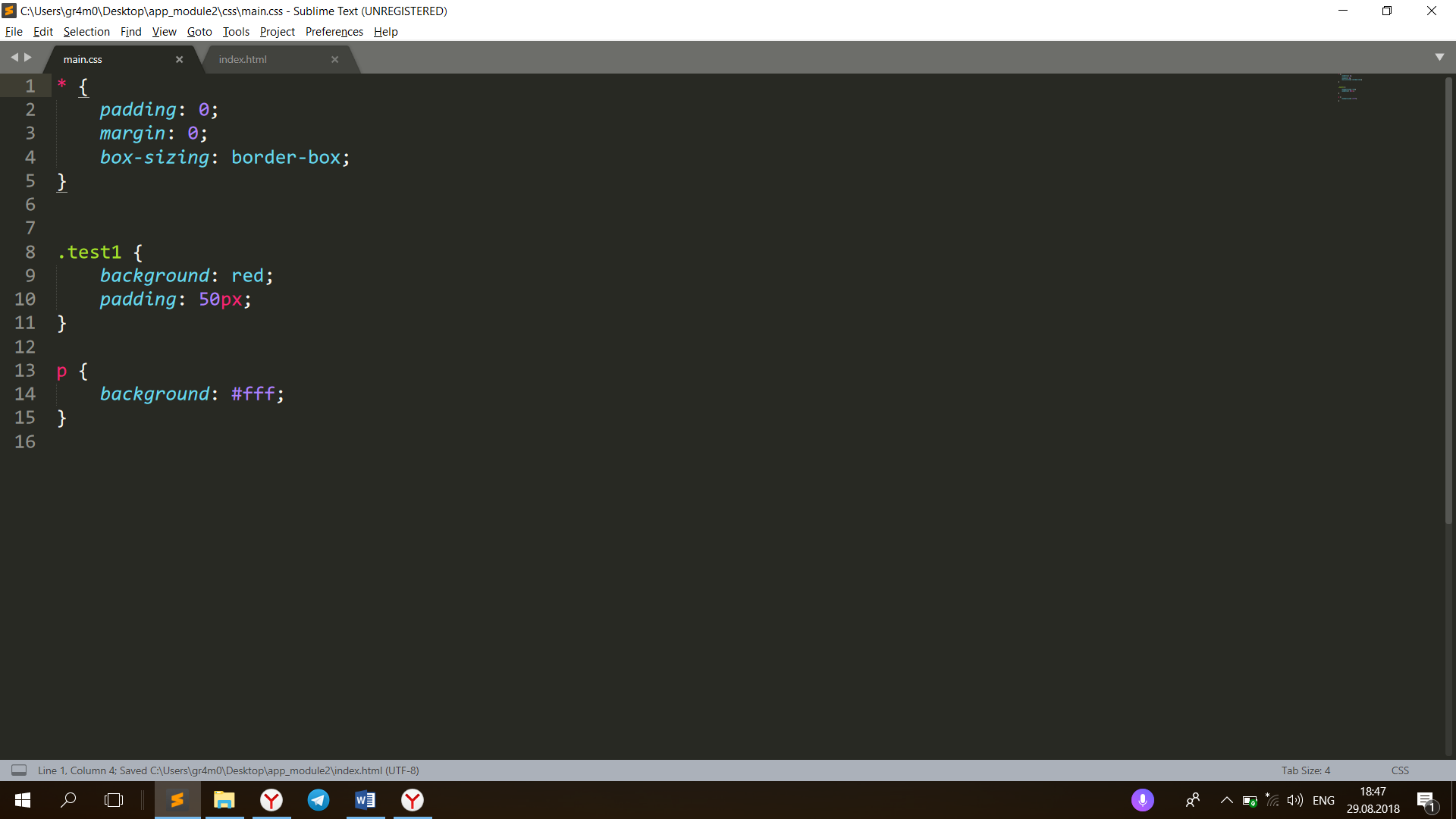
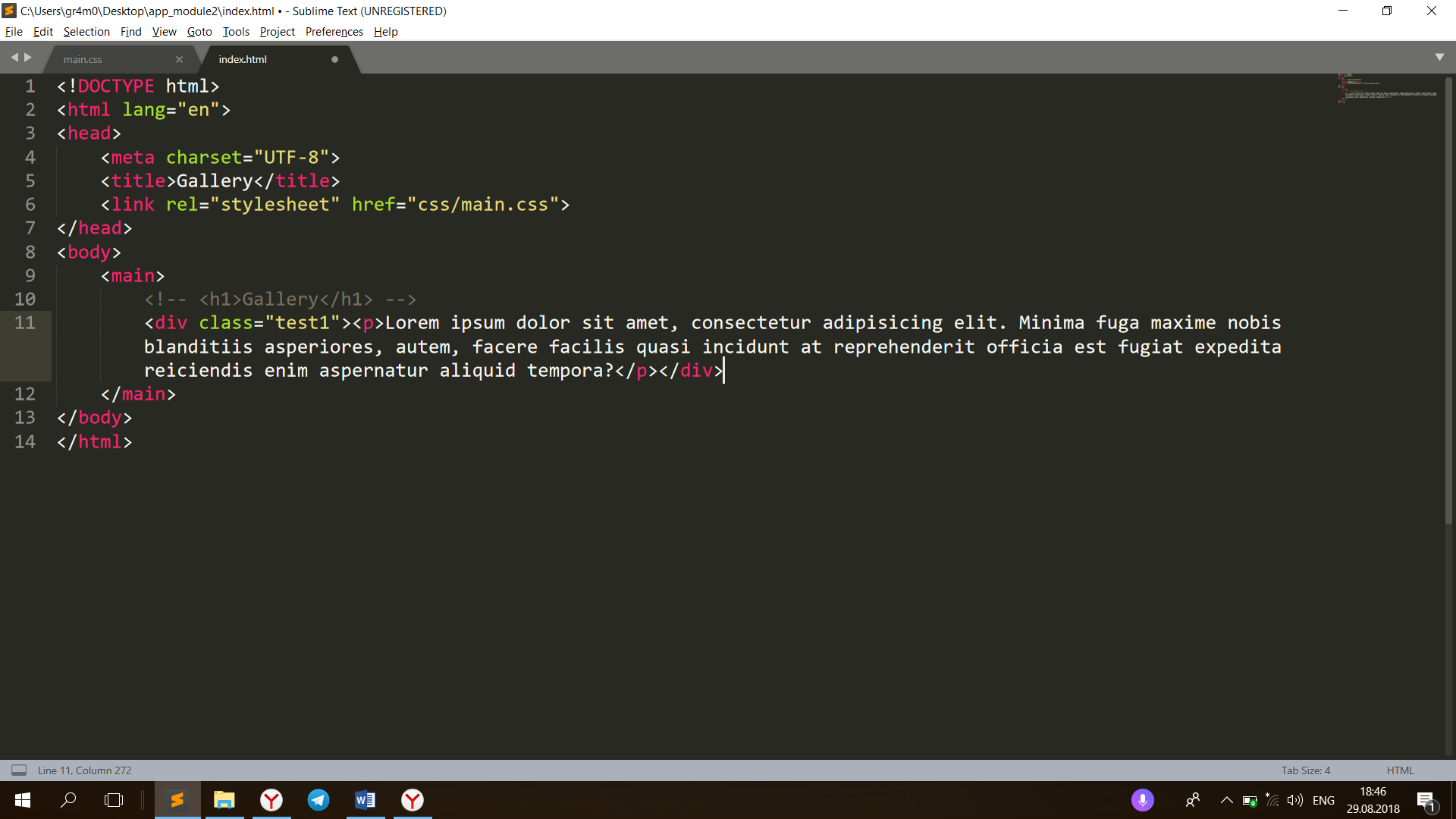


Также остановимся поподробнее на самом свойстве margin. У каждого элемента на странице есть четыре стороны: верх (top), право (right), низ (bottom) и лево (left). Обратите внимание, я указал названия сторон по часовой стрелке, это очень важно. Свойство margin можно задать следующими способами:

* margin: 50px; — все четыре стороны будут иметь отступ 50px;
* margin: 50px 20px; — top и bottom будут иметь отступ 50px, а right и left — 20px;
* margin: 50px 40px 30px; — top будет иметь отступ 50px, left и right по 40px, bottom — 30px;
* margin: 50px 40px 30px 20px; — top будет иметь отступ 50px, right — 40px, bottom — 30px, left — 20px;
* margin: auto; — выравнивает элемент по горизонтальной оси.

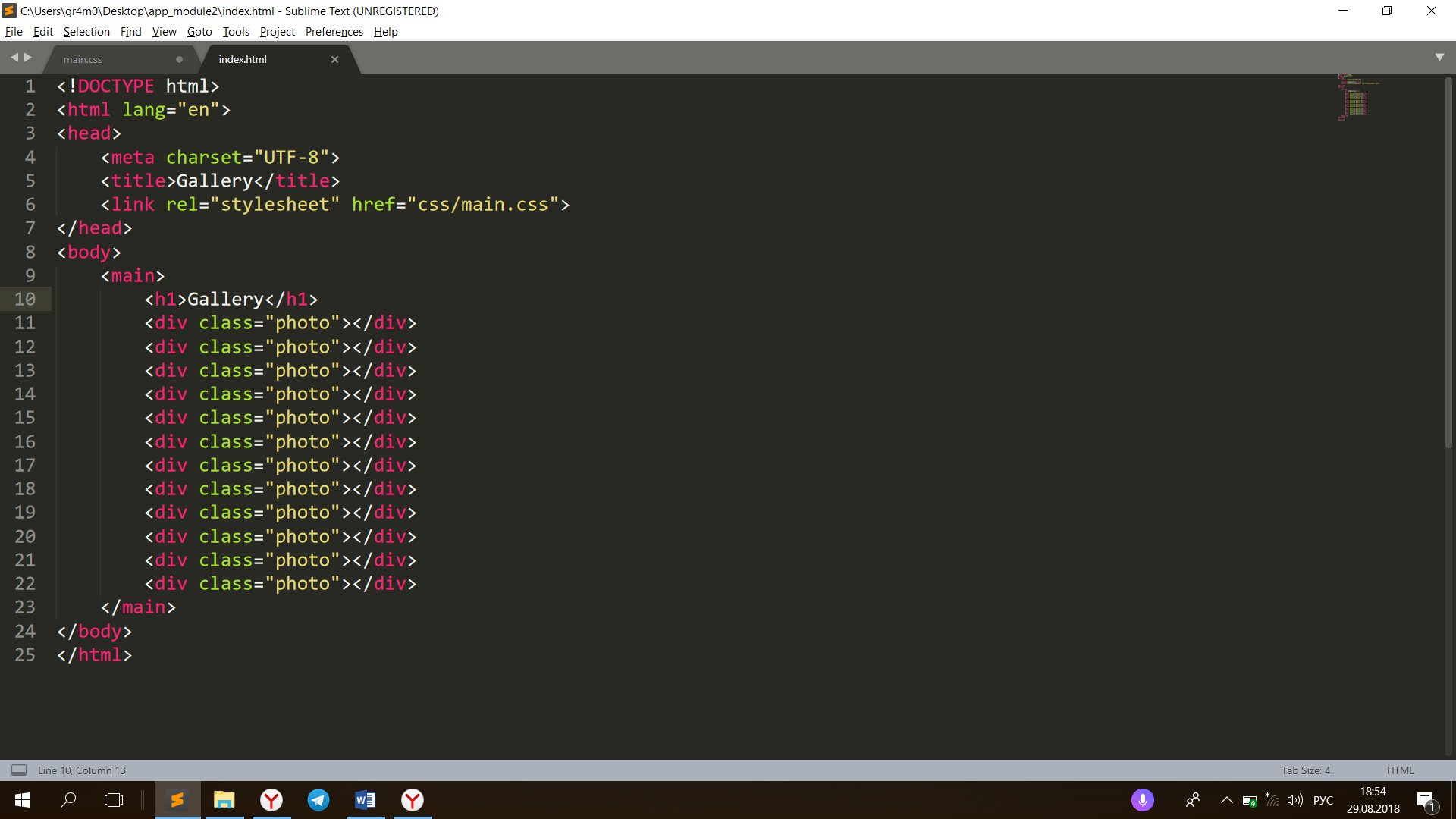
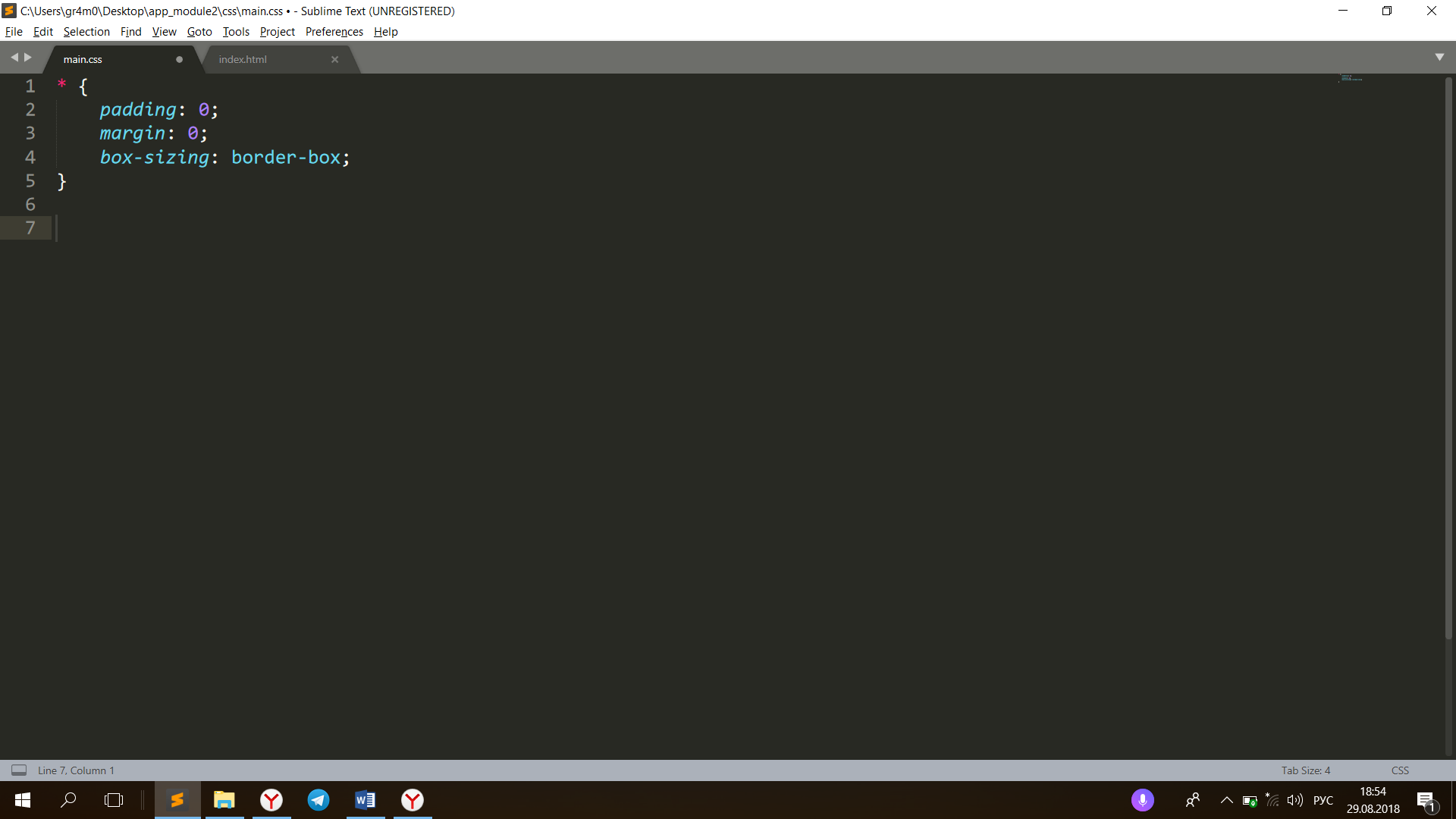
Переходим к свойству padding. Padding считается от исходного размера элемента. Например, если у элемента div высота и ширина по 100 пикселей, то если задать padding: 50px; размер div’a не изменится, а содержимое будет сдвинуто к центру невидимой рамкой размером в 50 пикселей.





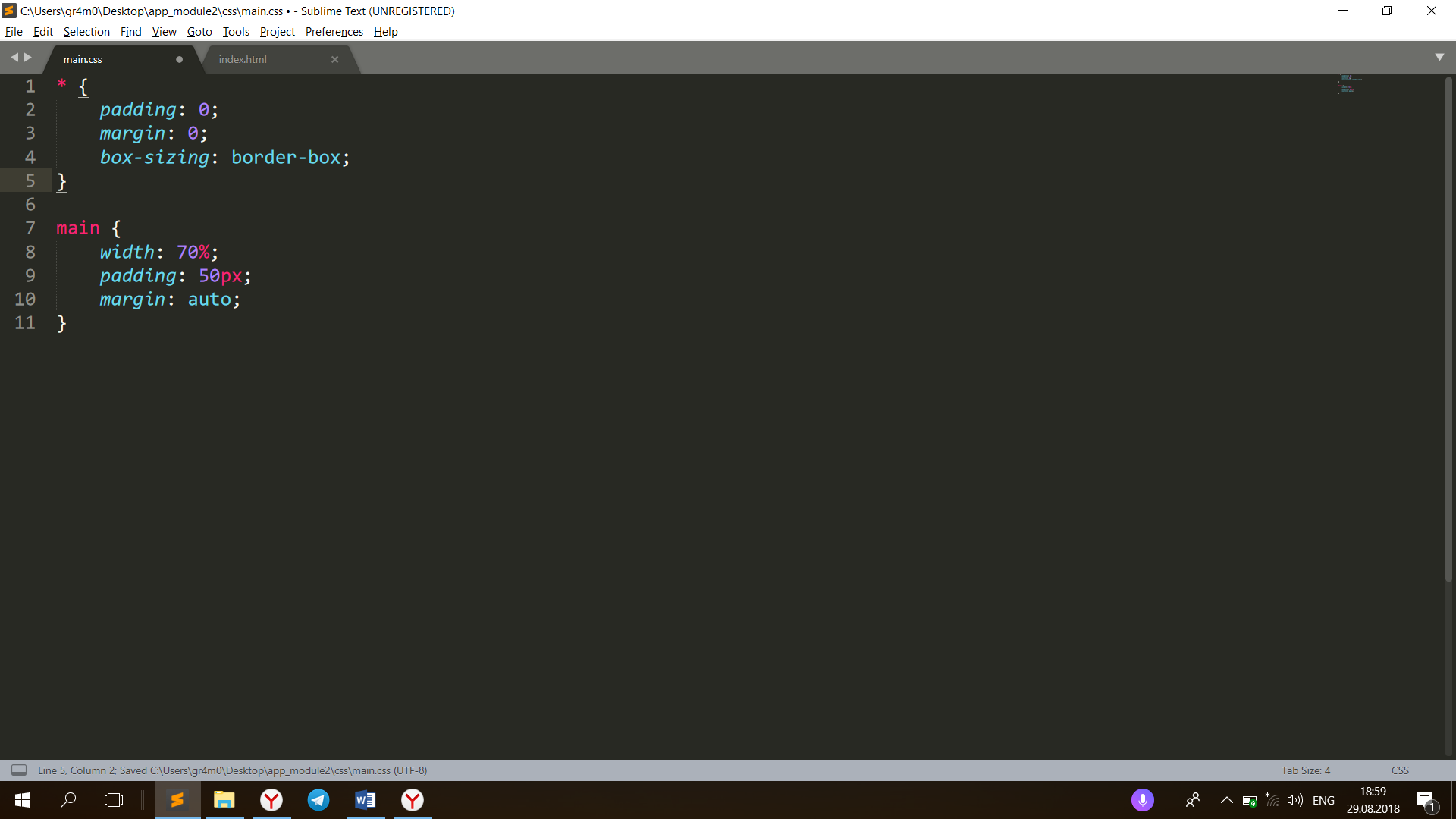
Для padding справедливы те же способы задания значения, что и для margin. Свойство box-sizing со значением border-box определяет правильное отображение box-модели элемента, например, padding в таком случае будет именно внутренним отступом, а не внешним, и не будет увеличивать размеры элемента.

Возвращаемся к нашей галерее.

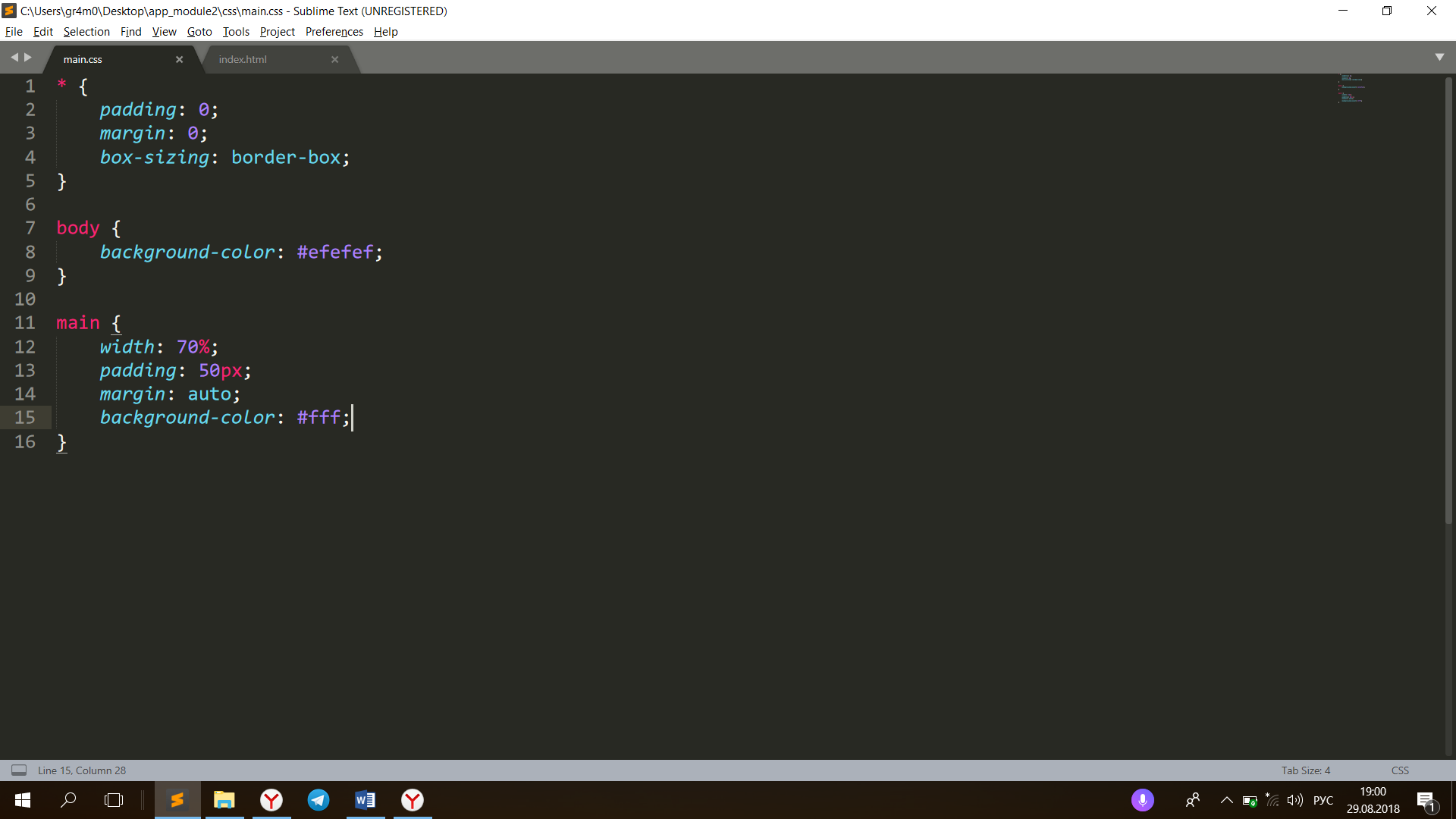
На данный момент мы имеем следующее:  

В файле main.css начинаем стилизовать main. Прописываем ему ширину 70%. Здесь следует сказать, что ширина считается от родительского элемента, в данном случае родителем выступает тег body. По умолчанию body имеет ширину 100% от ширины окна браузера, поэтому справедливо говорить, что main имеет 70% ширины окна.

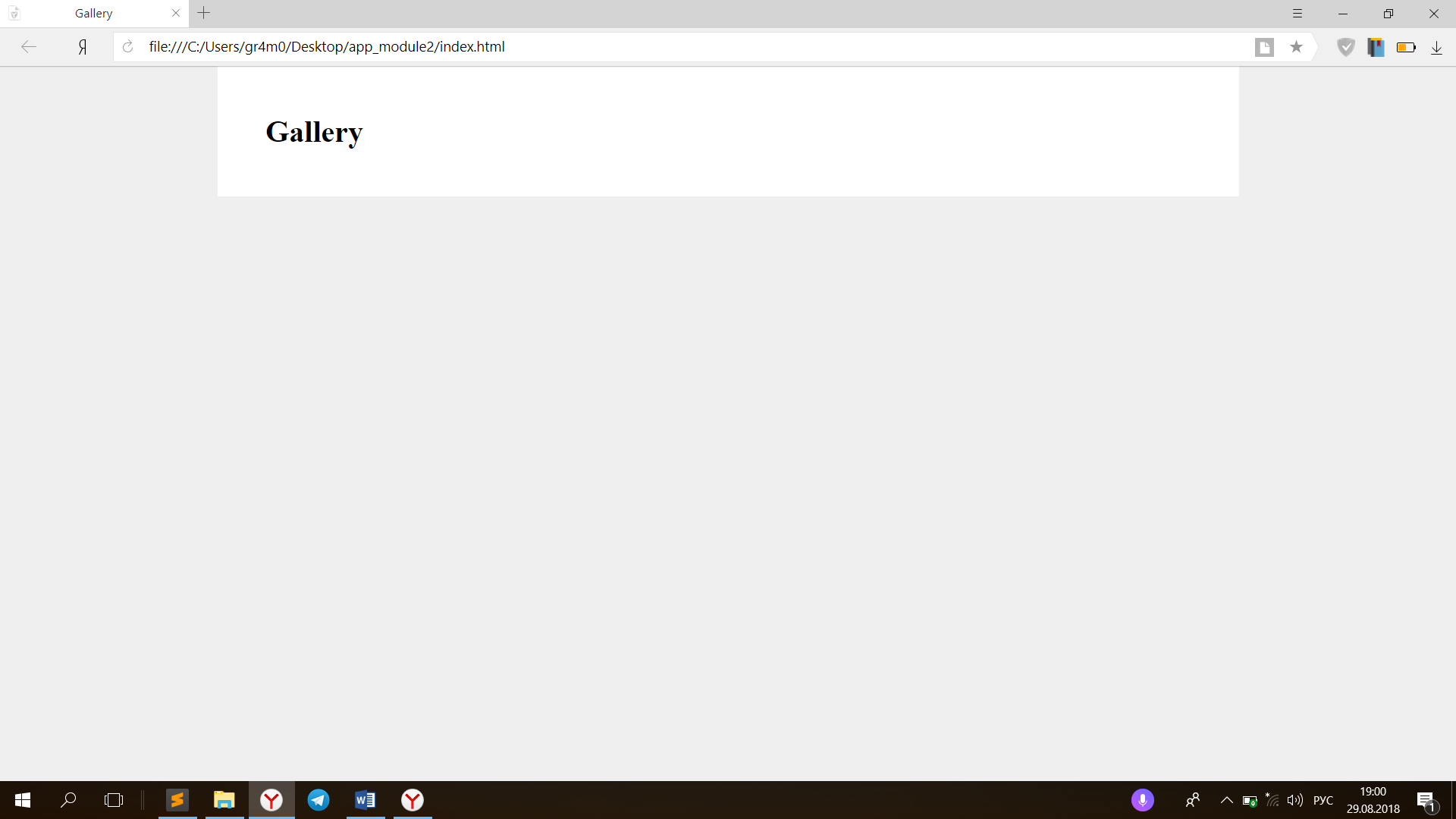
Также добавляем внутренний отступ в 50 пикселей. И центрируем его по горизонтали.



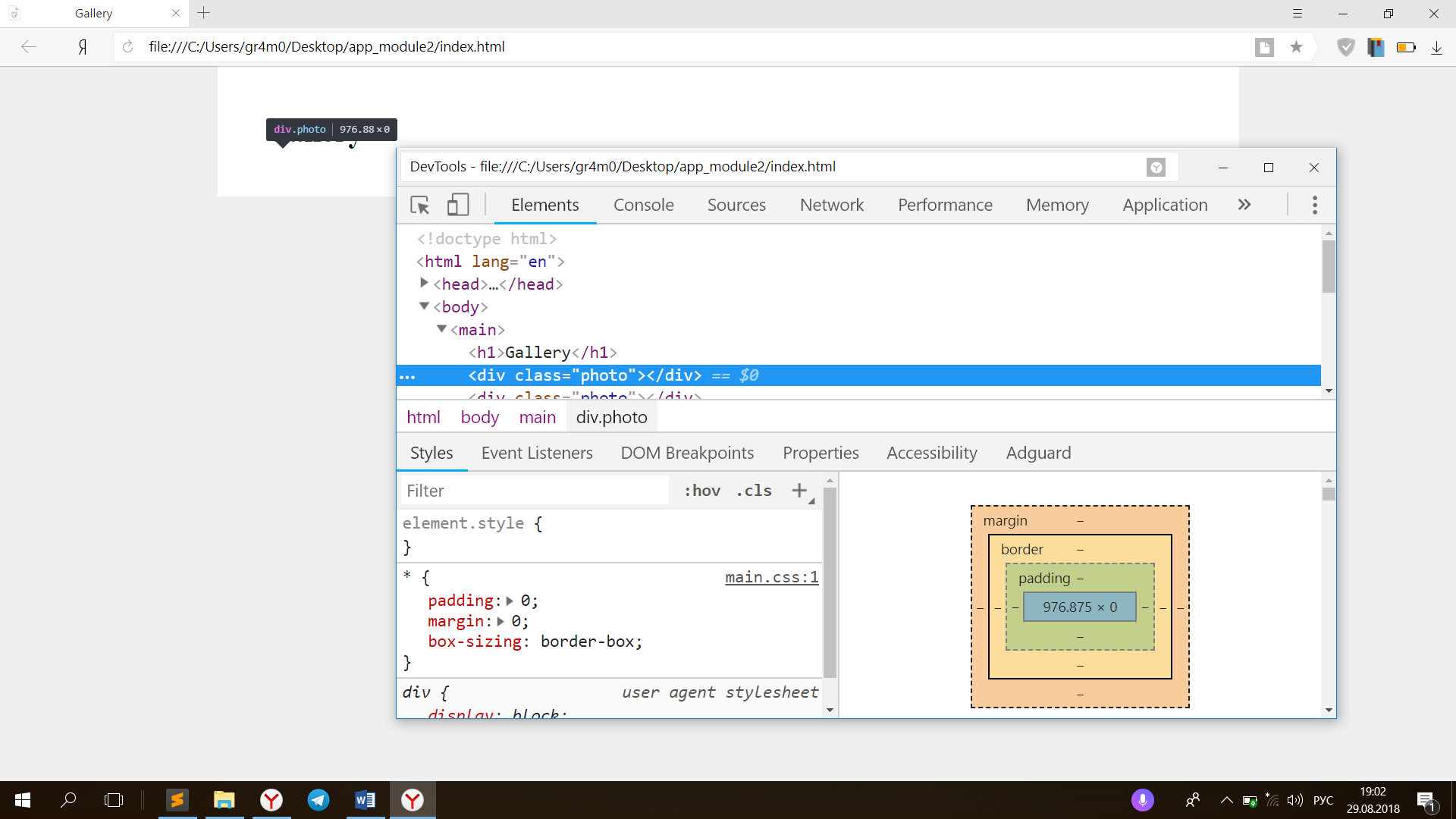
Устанавливаем для тега body светло-серый фон, а для main — белый.

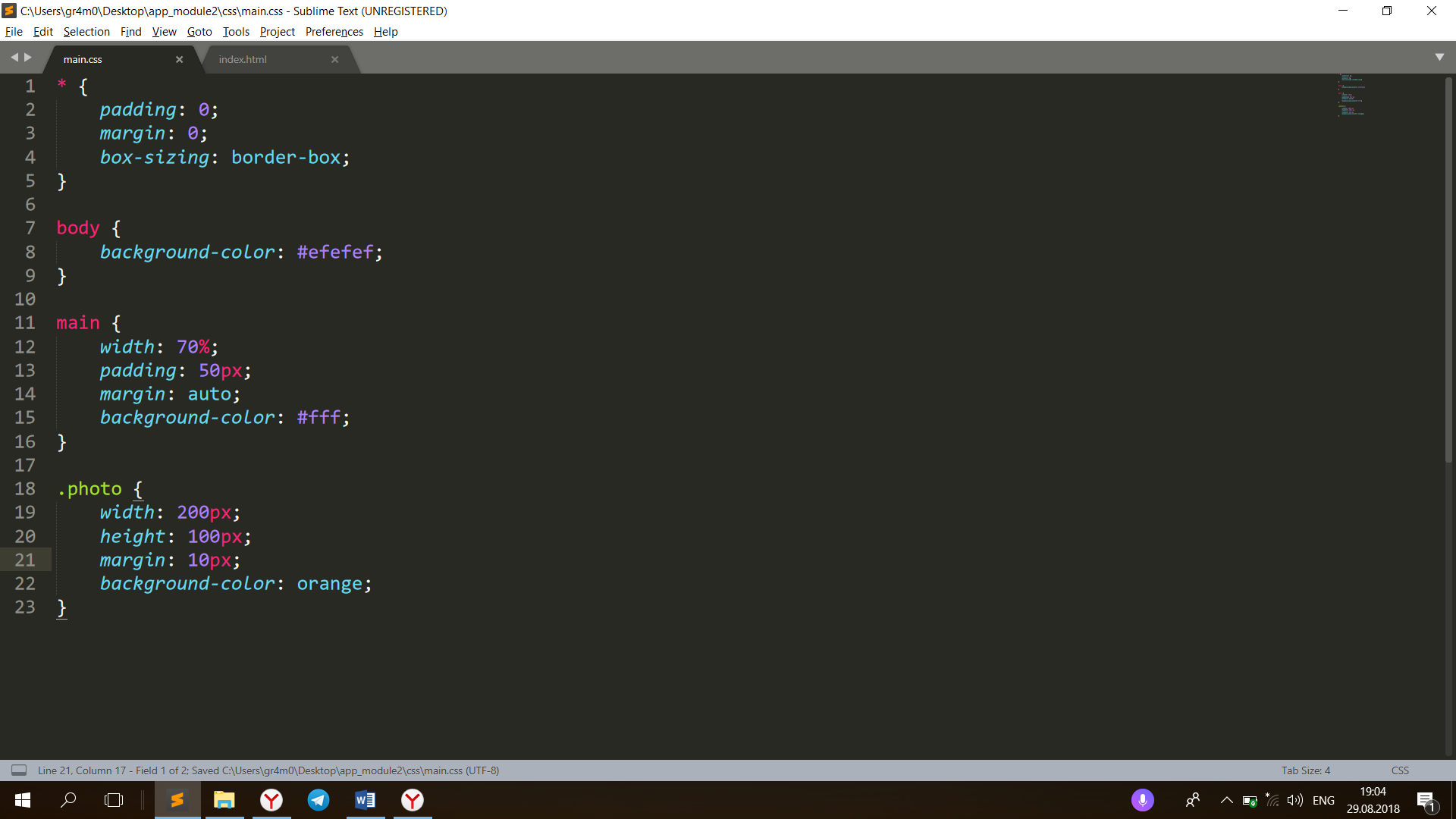


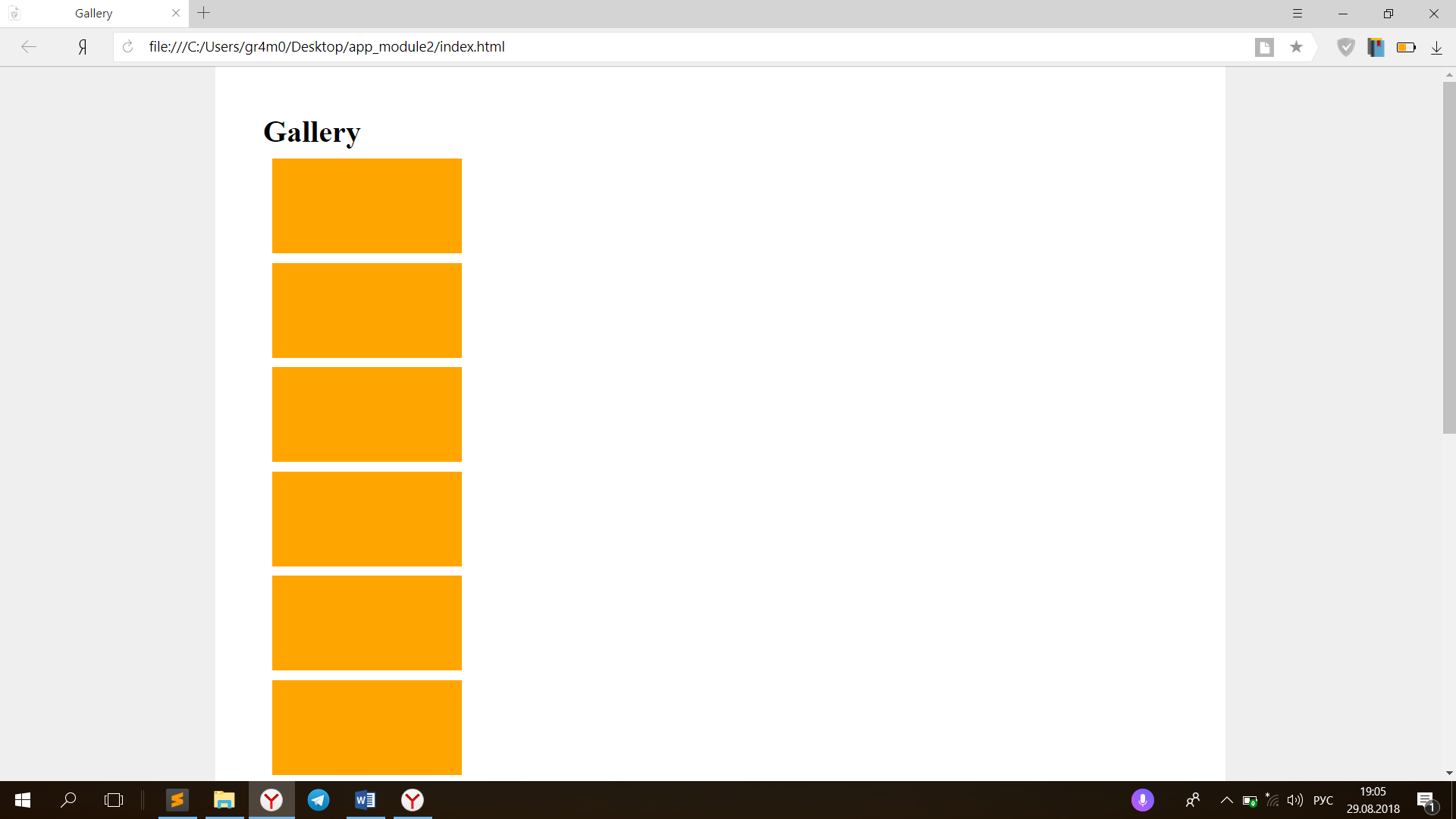
Вот, что мы имеем на текущий момент.



В заключении занятия сделаем наши «дивы» видимыми. Отмечу, что тег div без внутренних элементов имеет нулевую высоту.



Поэтому установим нашим контейнером для изображений размеры и небольшой внешний отступ.



На этом второе занятие завершается. На следующем уроке мы изучим базовые принципы позиционирования элементов с помощью css-свойств.