**Тема 1. Початок роботи**

Переходимо на сайт



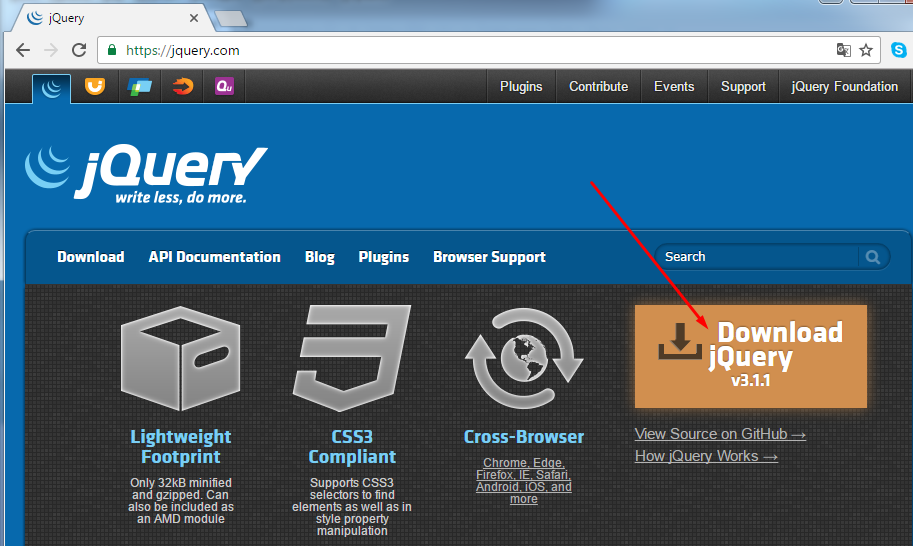
Бібліотека jQuery – це бібліотека, яка написана на мові JavaScript.

Ця бібліотека створена для того, щоб спростити роботу з JavaScript. Це означає, - те, що ми плануємо створити на JavaScript, ми можемо створити набагато швидше і зручніше з використанням jQuery.

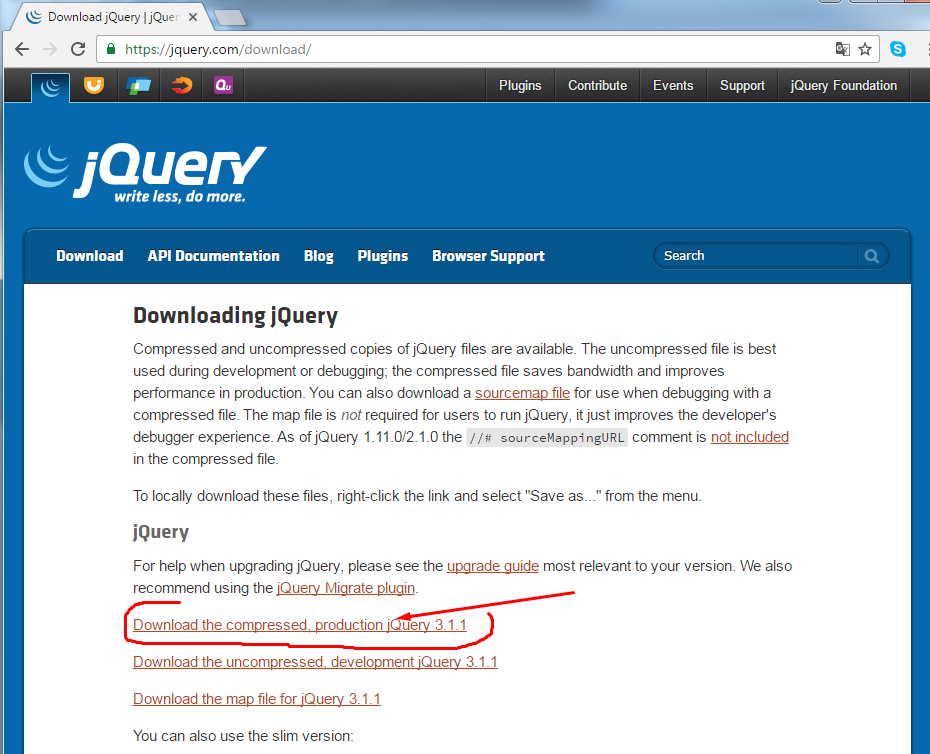
Слоган (лозунг) jQuery є наступним: **Пиши менше, роби більше: write less, do more**.



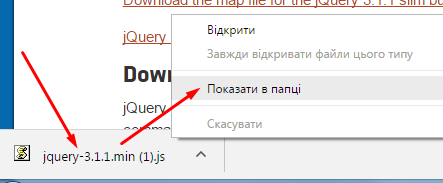
Вибираємо Download jQuery (завантажити jQuery)

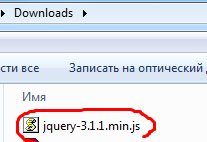


Одержуємо



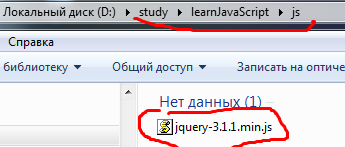
Тут вибираємо для завантаження стиснуту (compressed) версію jQuery.





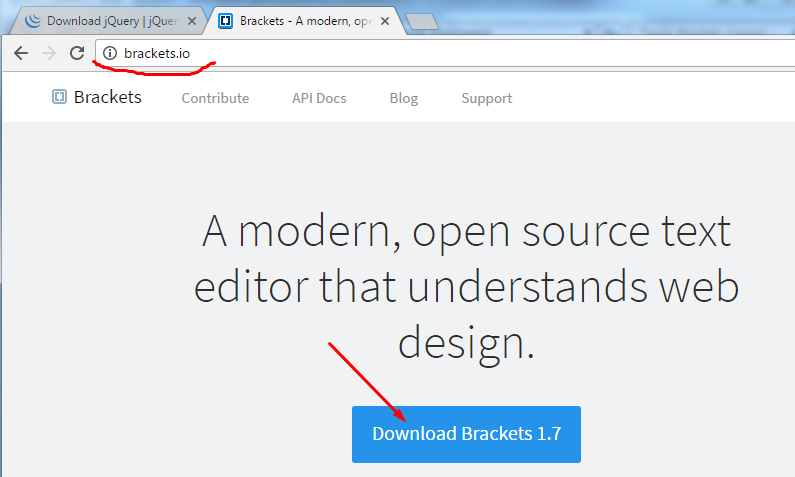
Отже, бібліотека jQuery у нас є.

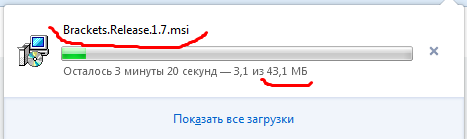
Розташуємо цю бібліотеку у зручному місці для доступу.



Перед тим, як перейти до роботи над проектом, виберемо редактор для роботи з кодом.

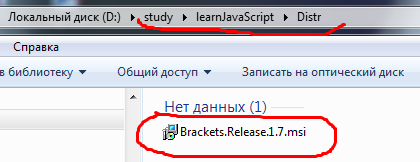
Для роботи будемо користуватись редактором коду Brackets. Завантажити цей редактор можна із сайту



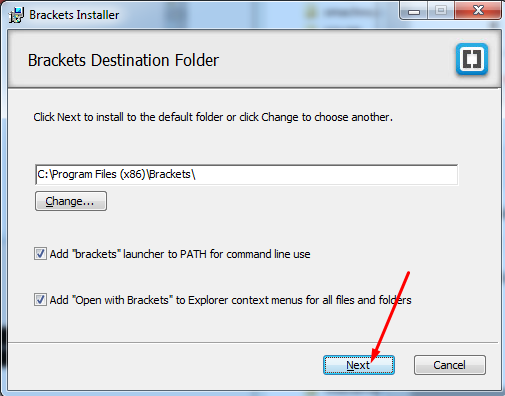


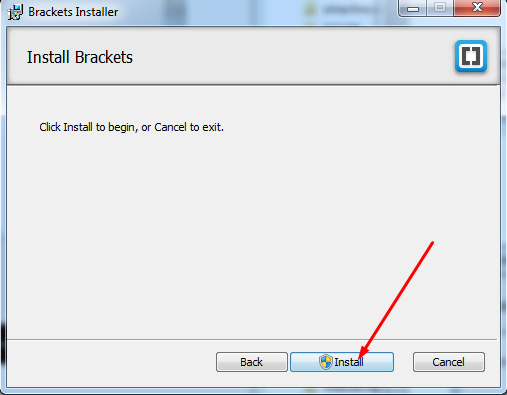
Бачимо, що дистрибутив займає 43,1 Мб.

Переносимо в папку

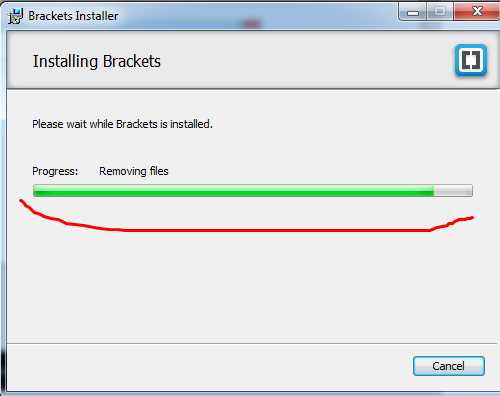


Встановлюємо програму Brackers

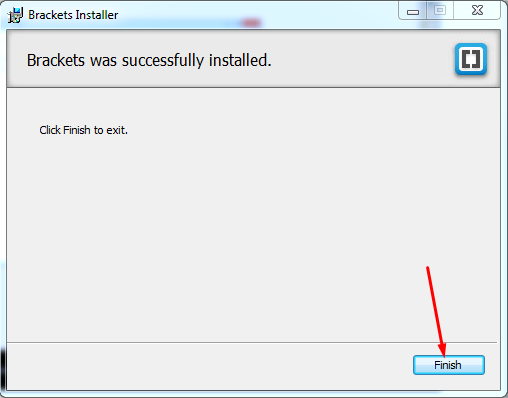




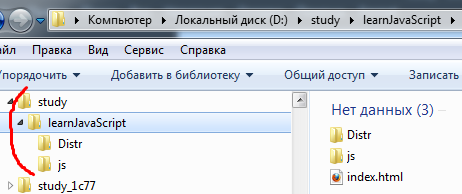
Чекаємо декілька хвилин, поки програма встановиться …



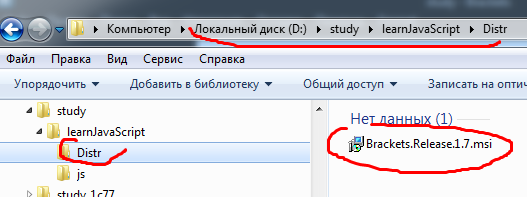
Врешті-решт програма встановилася.



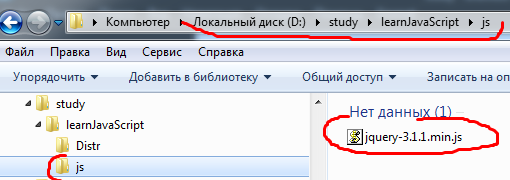
Створимо файл index.html в папці learnJavaScript. Одержимо наступну структуру папок і файлів



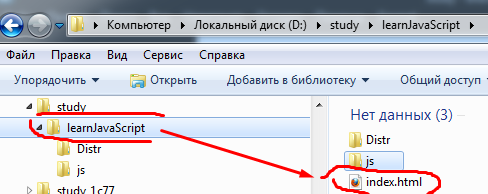
При цьому в папці Distr знаходиться дистрибутив Brackets



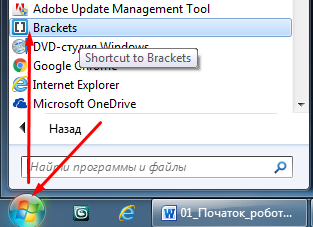
А в папці js знаходиться бібліотека jQuery

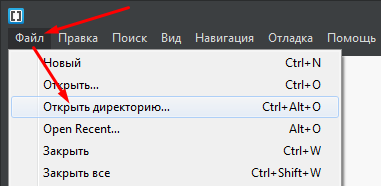


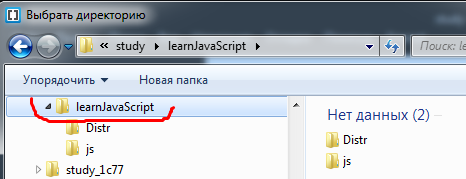
А файл index.html знаходиться у кореневій папці learnJavaScript

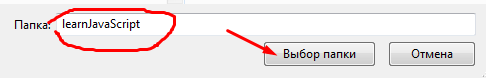


Запускаємо програму

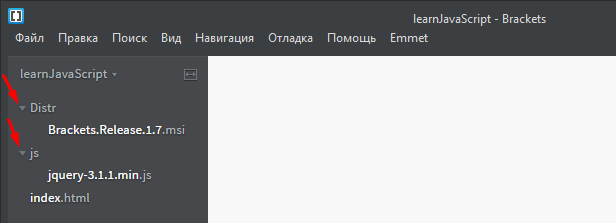








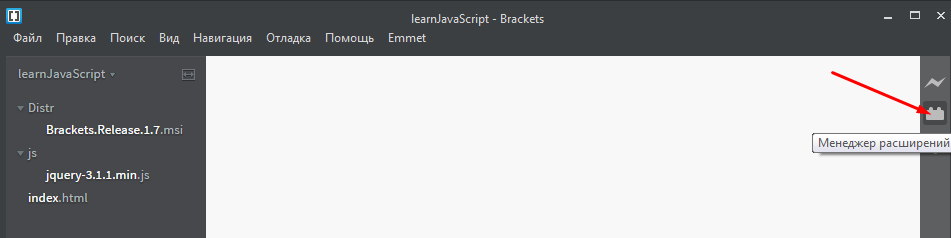
В результаті поточною папкою стала папка d:\study\learnJavaScript



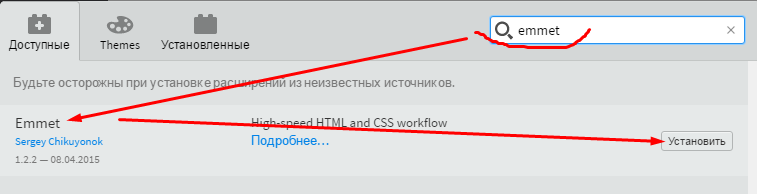
Виконавши клік біля Distr та клік біля js, можемо побачити вміст цих папок – відповідно дистрибутив Brackets та бібліотеку jQuery.

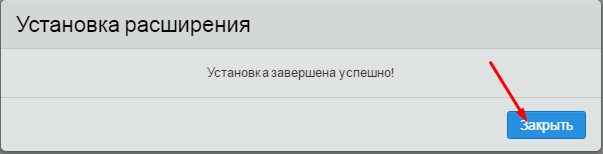
Наше завдання – в пустому файлі index.html створити каркас html-файлу.

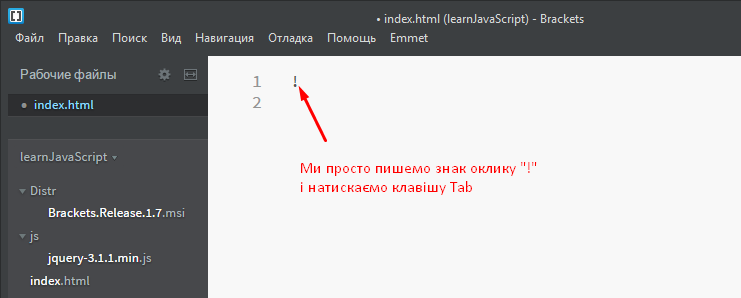
Щоб не писати все вручну, переходимо в Менеджер розширень



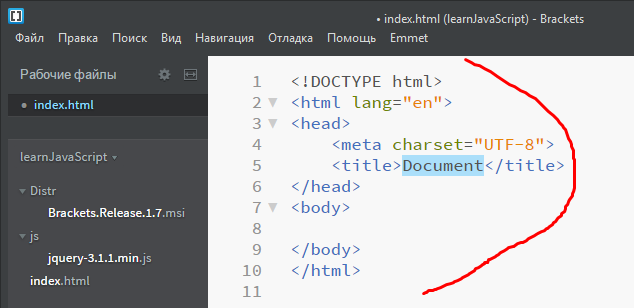
Далі в доступних розширеннях введемо назву плагіну.





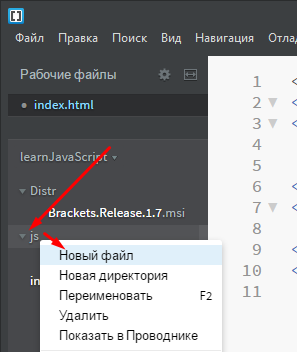


В результаті одержуємо

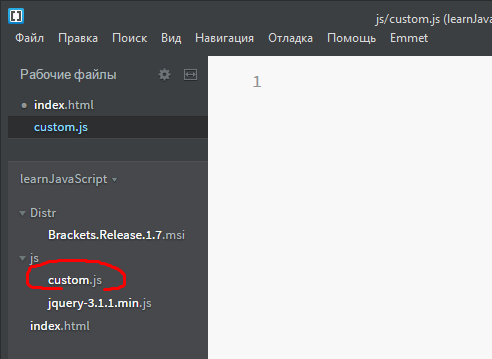


В папці js створимо ще один файл custom.js (в ньому ми будемо писати самостійно код).

В контекстному меню вибираємо Новий файл



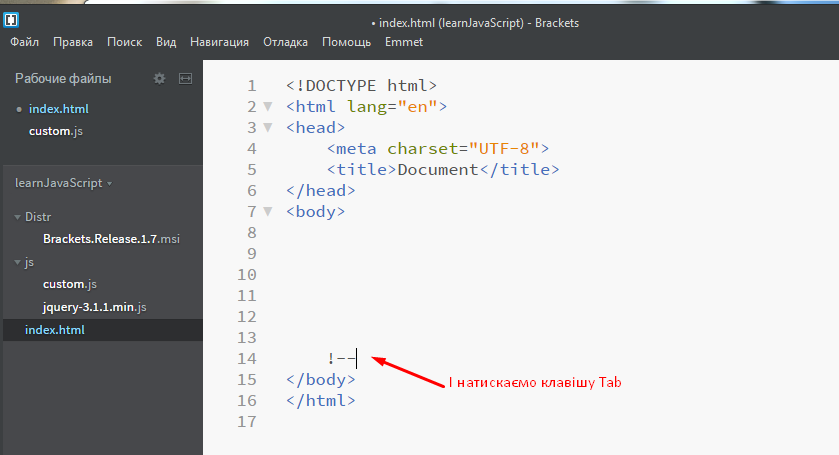
І створюємо файл custom.js



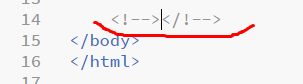
Отже, у нас в папці js є 2 файли з розширенням js. Їх треба підключити в індексному файлі.

Робиться це в розділі body перед самим закриттям розділу body.

Для того, щоб було все наочно, спочатку створимо коментар.



Одержуємо



Можемо писати коментар. Підключаємо бібліотеку jQuery.



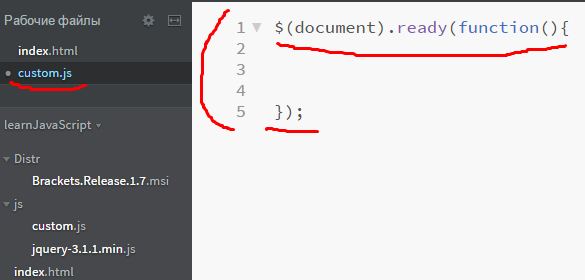
А далі маємо право підключати всі файли, які пов’язані з jQuery/



Тут звертаємо увагу на те, якщо ми підключимо спочатку якийсь js-файл, а потім лише бібліотеку jQuery, то у нас бібліотека jQuery працювати не буде.

Відкриємо файл custom.js

Відразу пишемо конструкцію

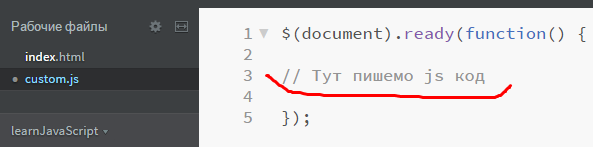


А всередині цієї конструкції ми маємо право писати різні скрипти на jQuery.

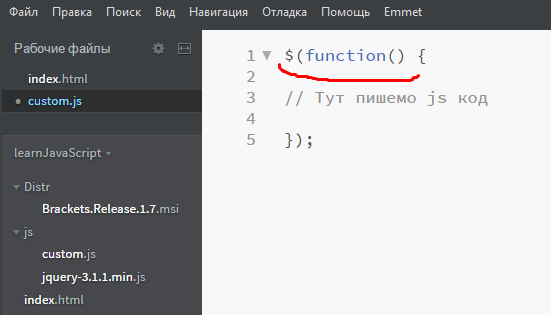
Що ж означає ця конструкція?

Вона говорить про наступне. Потрібно діждатись повного завантаження html-документа і тільки потім виконувати скрипти, які будуть написані в тілі даної функції.

Коментар в JavaScript можна писати так:



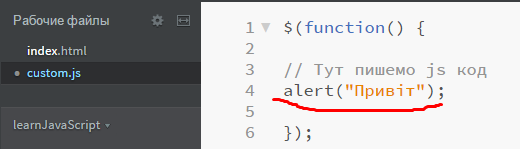
Таку конструкцію писати довго, тому те саме пишуть більш коротко.



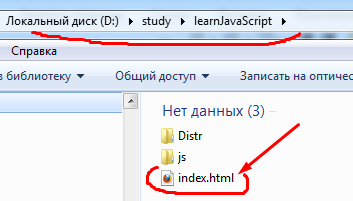
Така форма запису запам’ятовується швидше. А виконує вона те ж само.

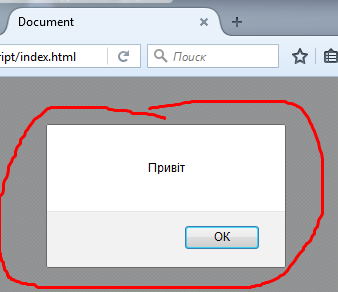
Переконаємось в тому, що все працює.

Зробимо так, щоб після того, як завантажиться html-сторінка, вивелось на екран спливаюче вікно з текстом «Привіт».

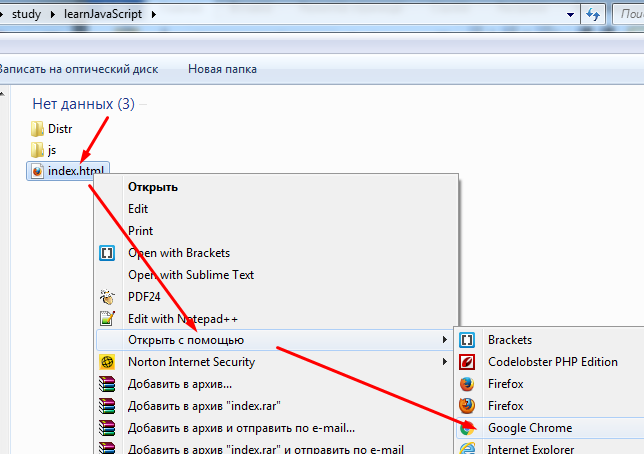


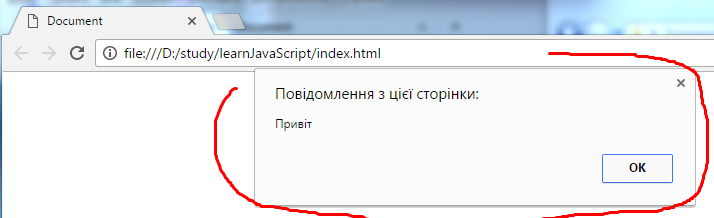
Відкриємо файл index.html





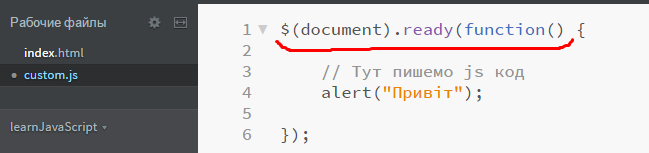
Відкриємо цей само index.html в браузері chrome.





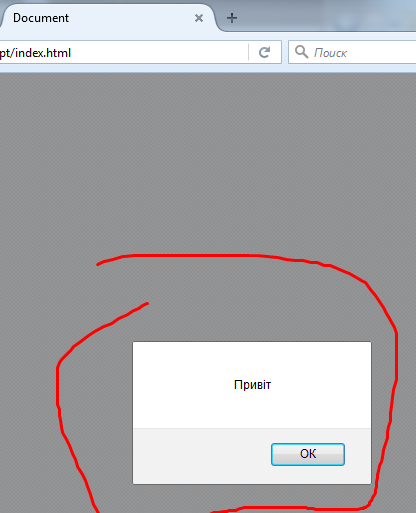
Отже, конструкція $(function() … працює і браузері firefox, і в браузері chrome.

Перевіримо іншу конструкцію.

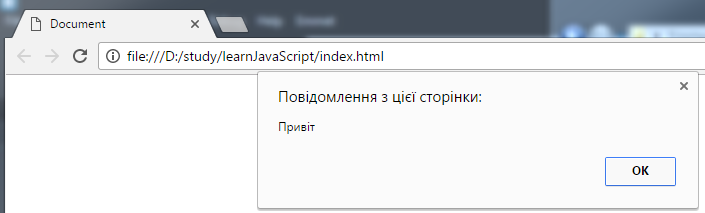


Зберігаємо зміни (Ctrl+S) і обновляємо в браузерах по черзі.

Cпочатку firefox

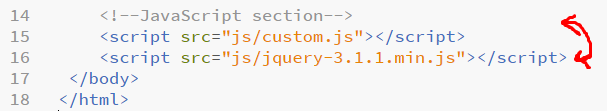


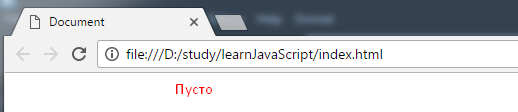
Далі в браузері chrome.



Все те ж само.

Подивимось, що буде, якщо підключити custom.js до підключення бібліотеки jQuery.





В цьому випадку ніякий скрипт не виконався і спливаюче вікно не з’явилось. Вся причина тільки в тому, що даний скрипт custom.js підключився **перед** бібліотекою jQuery.

Повертаємо назад

