# Getting started

* Please watch the video Basics of WEB and WEB TA before doing these tasks;
* Create new project in Visual Studio of type Unit Test Project – MsTest (or simply Unit Test Project);
* Through NuGet, add the following packages:
  1. Selenium.WebDriver
  2. Selenium.WebDriver.ChromeDriver. Version of this package MUST match Chrome version on your machine. If it doesn’t and Chrome doesn’t auto-update – just change package version.

* Add the following code to your default unit test to verify everything works:

IWebDriver driver = new ChromeDriver(); // Creates a new Chrome instance and opens the browser

driver.Navigate().GoToUrl("https://www.bbc.com"); // Navigates to a page by address

## Other useful bits of code

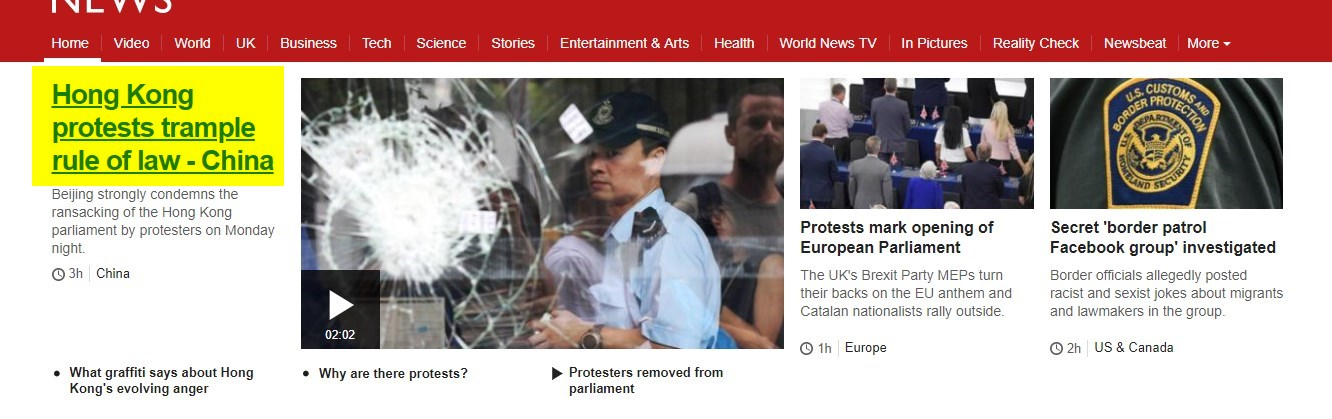
IWebElement element = driver.FindElement(By.XPath("xpath")); // Replace “xpath” with correct xpath. This returns an IWebElement interface of the element found by Xpath. This interface can then be used to click the element or perform other actions (such as getting its text, checking visibility).

Assert.AreEqual("text", element.Text); // Checks that the 2 objects (in this case, string “text” and the text contents of the element you found) are equal, throws exception if not. Since the exception will fail your entire test, your test result (pass or fail) will reflect check results.

# Tasks

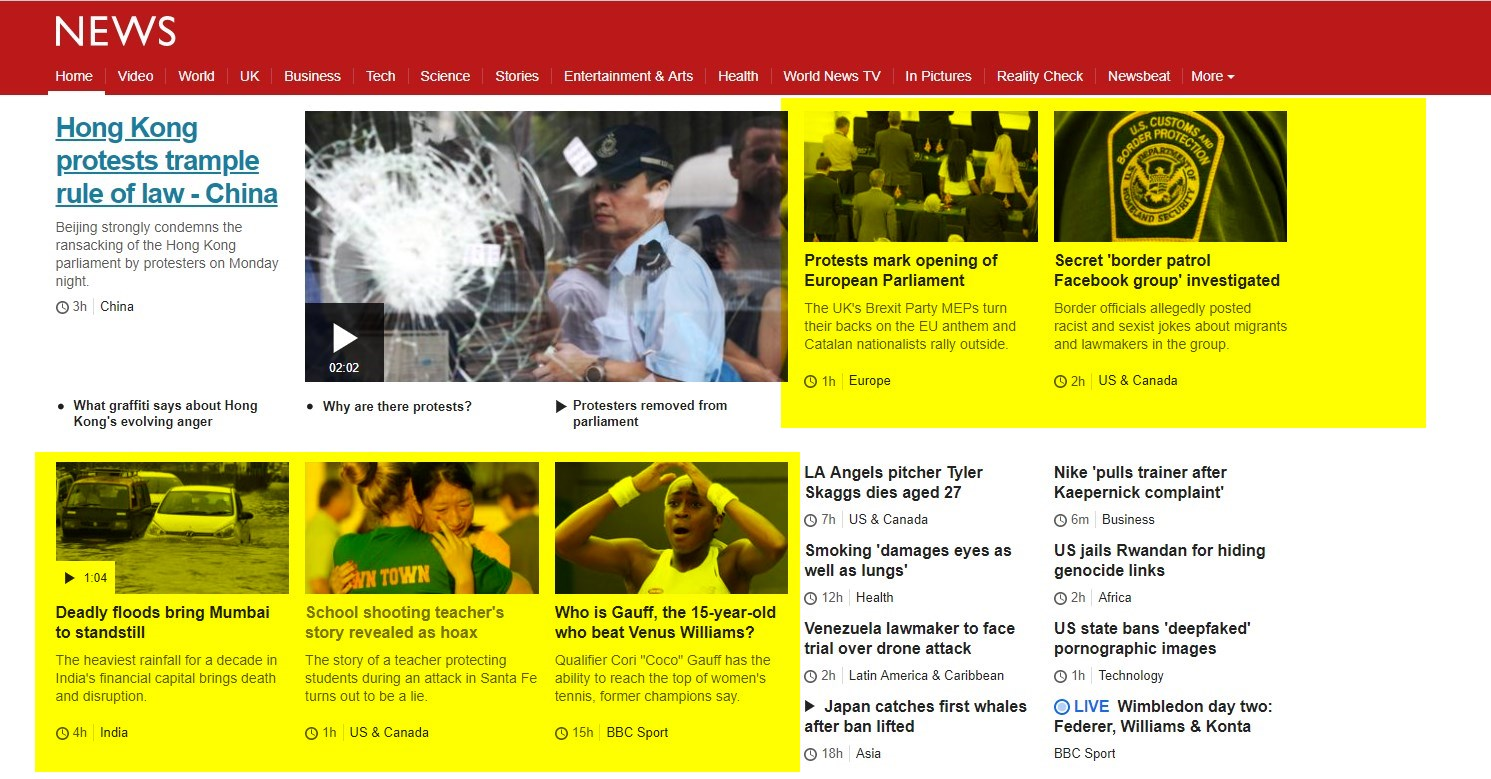
You don’t need anything other than the code bits above and what you already know to solve these tasks. Let your trainer know if you get totally stuck.

1. Add a test that goes to BBC, clicks on News and checks the name of the headline article against a hard-coded value:



1. Add another test that does the same thing, except checks secondary article titles against a hard-coded List<String> of Expected Results.

Imagine that you are testing the BBC site on a test environment, and you know what the titles will be. Your test, then, checks that these titles are in fact present on the page. The test should pass if all the titles are found, and fail if they are not found. Secondary titles are these:



1. Write a test that stores the text of the Category link under the headline article. Then, it should enter this text in the Search bar. Then, it should check the name of the first article against a hardcoded value, like in the first test. Illustrated below (article name to check highlighted in red):

