Experimenting APIs being relative to Layout Viewport

ymalik@chromium.org, bokan@chromium.org

Summary

I need some help testing the inert-visual-viewport experiment which basically makes the “visual viewport” non-existent to developers using scroll related methods outlined in the CSSOM View Module. Please let me know if you see any strange bugs with fixed position elements or any unusual behavior caused by my change (summarized below).

Overview

Having window.scroll properties being relative to the visual viewport breaks some pages under pinch-zoom.

Consider this example:

1) Go to http://www.nytimes.com/
2) Pinch-zoom in
    - Android/Pixel: Two-finger zoom in gesture on the screen
    - Macbook: Two-finger zoom in gesture on trackpad
3) Click on the gear button in the top right corner to bring down the fixed-style settings menu.

Expected: The settings menu pops up right beneath the gear button
Actual: The settings menu pops up at an offset from the gear button

The plan is to have all APIs reflect the layout viewport. This has been implemented behind the “inert-visual-viewport” flag.

To test this, start chrome with the inert-visual-viewport command-line flag and perform the steps above. The settings menu should now appear right beneath the gear button.

The following API methods were originally relative to the visual viewport, but will be relative to the layout viewport if the inert-visual-viewport flag is set:

core/frame/Window.idl	**core/dom/Element.idl**
long innerWidth;	double scrollTop;

long innerHeight;	double scrollLeft;
double scrollX;	
double scrollY;	
double pageXOffset;	
double pageYOffset;	
scroll(double x, double y);	
scrollTo(double x, double y);	
scrollBy(double x, double y);	

See this to see what other API methods are relative to.

Addendum: Edge uses the layout viewport for the Element.idl methods but still uses the visual viewport for all the window.idl methods