



OBJECTIF LUNE

How to overflow text into multiple DIV objects

Created By: Rodrigue Noubissie

Last Updated: 20/09/2017

TOC

Introduction.....	3
The method	3
The Script.....	4
Applications	5

Introduction

One often sought requirement is the ability to dynamically flow text from one <DIV> object into one or multiple other <DIV> objects scattered on the same page or any other pages of the same document. In addition, it is also often required to break up a <DIV> object, which has reached the bottom margin of a page to allow it to overflow text onto the next page.

A use case scenario would be a Christmas message spanning multiple lines where there is a requirement for it to overflow, should it not fit into the first box, into multiple positioned boxes, or columns placed at different locations or different pages of the Christmas card.

This guide highlights the simple steps a user can follow to overflow text into multiple <DIV> objects in PrintShopMail, PlanetPress or PReS Connect.

The method

The method consists in:

- ☐ Placing two separate <div> objects or positioned boxes on the page and assign each a unique CSS ID.
- ☐ Inserting a paragraph element (<p></p>) element inside each of the <div> objects
- ☐ Dragging and dropping your data mapper or database field inside the main <div> object
- ☐ Adding a script in the Script pane, which targets the current section as the *Selector*. *Example Selector: [section="Section 1"]*
- ☐ Writing a script which does the following:
 - Calculate the height of the main <div> container.
 - Calculate the height of the <p> element when content is loaded inside the main <div> container.
 - Compare the height of the <p> element to the height of the <div> container, then remove the last word from <p> and assign it to a temporary array variable until the remaining text in <p> can fit inside the <div> container.
 - Reassemble all the words from the temporary array variable and then place the resulting text into the paragraph element of the second <div> container.
 - If the second <div container> contains no text, then hide or delete it.

The Script

The above pseudo code can be translated into the following reusable JavaScript function, which of course can be modified by anyone to suit own needs.

The functions accepts three arguments:

- *mainDivID* (mandatory) is the CSS ID of the main <div> element or positioned box container.
- *overflowDivID* (mandatory) is the CSS ID of the second <div> element which will accommodate the overflow text.
- *overflowDivBottomPadding* (optional) is bottom padding of the above second <div> element. It is an optional argument, which defaults to 15px if it is not specified.

```
function overflowingDiv(mainDivID,overflowDivID,overflowDivBottomPadding) {

    var mainDivHeight, p1Height, p2Height;
    var p1, p2, overflowDiv;
    var p1Text, p2Text;
    var lastWord, p1Temp;

    overflowDivBottomPadding = overflowDivBottomPadding || 15;
    mainDivHeight = parseInt(query(mainDivID).css("height"));
    overflowDiv = query(overflowDivID);

    p1 = query(mainDivID + ' p');
    p2 = query(overflowDivID + ' p');
    p1Height = parseInt(p1.css('height'));
    p1Text= p1.text().split(' ');
    p2Text = [];

    while(p1Height && p1Height > mainDivHeight){
        //remove last word from Paragraph1 in main Div
        lastWord = p1Text.pop();

        //prepend the above last word to Parapagh2 in overflow Div
        p2Text.unshift(lastWord + ' ');

        //reassemble Paragrah1 text
        p1Temp = p1Text.join(' ');

        // reassign Paragrah1 text to p1 (minus last word of previous iteration)
        p1.text(p1Temp);
        //recalculate Paragraph1 height
        p1Height = parseInt(p1.css('height'));
    }
    //Assemble Paragraph2 in overflow Div
    p2.text(p2Text.join(''));

    //hide oveflow Div if it is empty
    p2Height = parseInt(p2.css('height'));
    if(p2Height ==0){
        overflowDiv.css('display','none');
    }else{
        p2Height+=overflowDivBottomPadding;
        overflowDiv.css('height', p2Height + 'px');
    }
}
```

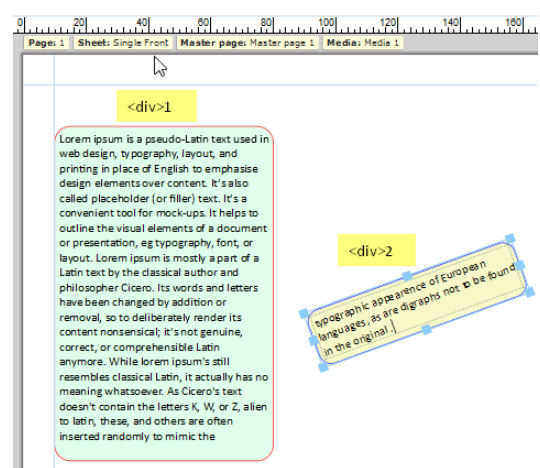
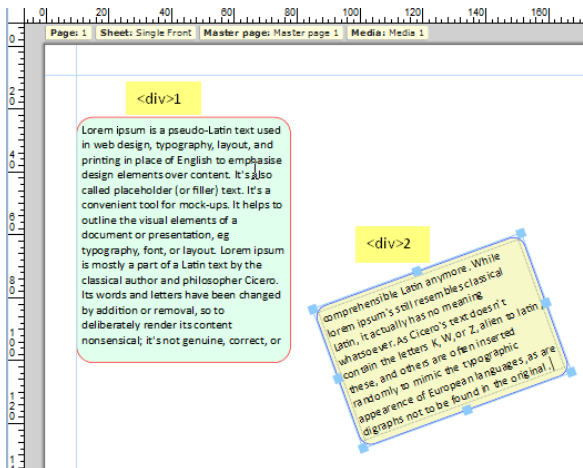
Applications

The attached example demonstrates how `<div>` overflow can be achieved for 2, 3 and even 4 `<div>`. There is an additional example on [Section 4], which simulates a page break within a `<div>` element.

[Section 1]: Two `<div>` overflow

The example in [Section 1] demonstrates a how to flow text from one positioned box object to another. To replicate it:

- Add two positioned boxes with CSS IDs `#mainDiv`, `#overflowDiv` on the page and format them in any way you like.
- Add an empty `<p></p>` element in each of the `<div>` elements.
- Drag and drop the field `@Message@` into the main positioned box with ID `#mainDiv`
- Add a script in the Script pane (`2DivOverflow`), set its Selector to `[section="Section 1"]`
- Copy paste the above function in the script, preceded by the following line
`overflowingDiv('#mainDiv','#overflowDiv');`
- Text should flow from `#mainDiv` to `#overflowDiv`. If you increase `#mainDiv's` height by dragging it, `#overflowDiv` should gradually decrease in size and content and eventually disappear off the page.



[Section 3]: Four `<div>` overflow

The example in [Section 3] demonstrates how to flow text into 4 different positioned boxes placed at different locations with the last positioned box being on the second page. To replicate it:

- Add four positioned boxes with CSS IDs `#stDiv`, `#ndDiv`, `#rdDiv` on the first page and `#thDiv` on the second page. Then format them in any way you like.
- Add an empty `<p></p>` element in each of the `<div>` elements.
- Set the maximum heights `#ndDiv` and `#rdDiv` positioned boxes should have. Open the `content_print_styles.css` and add the following CSS code:

```
/* [Section 3] CSS */
#ndDiv{max-height: 40mm; }
#rdDiv{max-height: 30mm; }
```

- Drag and drop the field `@Message@` into the main positioned box with ID `#stDiv`
- Add a script in the Script pane (`4DivOverflow`), set its Selector to `[section="Section 3"]`
- Assuming that the above `overflowingDiv()` already exists in your template, copy and paste the following code in the new script:

```
overflowingDiv('#stDiv','#ndDiv');
overflowingDiv('#ndDiv','#rdDiv');
overflowingDiv('#rdDiv','#thDiv');
```

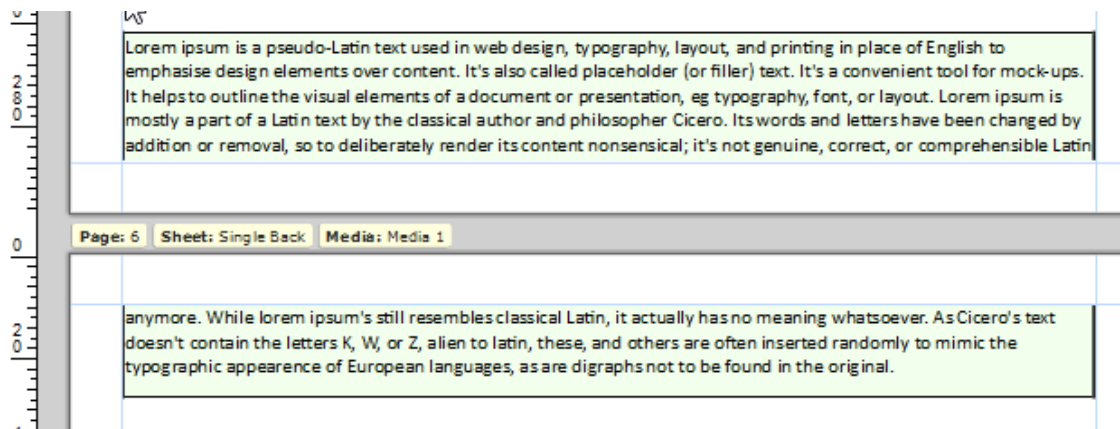
- As a result, a long enough text should overflow into all the `<div>` and unused `<div>` (except the first) should disappear off the pages if they are empty.

[Section 4]: Simulate a Page Break within a positioned box

The example in [Section 4] simply simulates how one could overflow a DIV object, which has reached the bottom margin of the current page, onto the next page.

The steps for replicating this example are similar to the steps used for [Section 1] except for the positions of the two positioned boxes, which need to be at the bottom margin of current page and top margin of next page.

The text should then flow automatically from the positioned box at the bottom margin of the current page to the positioned box located at the top margin of the following page.



Increasing the height of the box at the bottom margin of the current page should decrease the height and content of the positioned box on the following page.

