

Healthy Infant Page Convention

There are 7 pieces of information needed to make a healthy infant page. However, only a few pieces are needed at any one time.



The LEVEL is the first piece and corresponds directly to the PALS level.

The next piece is the Section which corresponds to which header the page appears under.

The Section

1

- [Discover](#)
 - Welcome
 - [Signals](#)

2

- [Check-in](#)

3

- [Video](#)

4

- [Activity](#)

5

- [Coach Call](#)

Care must be taken here because sections are always labeled starting at 1.

Check a page in the level to make sure what sections are there.

1

- [Discover](#)
 - Objectives
 - [Signals](#)
 - [Sensitivity](#)
 - [Maintaining](#)
 - [Introducing](#)
 - [Words & Actions](#)
 - [Learning](#)
 - [Activities](#)
 - [Active learning](#)
 - [Congratulations!](#)

2

- [Video](#)

3

- [Activity](#)


4

- [Coach Call](#)

At this point if one were entering this into a spreadsheet for an example page in Level 1 Section 1 it would look like this:

B	C
Level	Section
1	1

Next is the page number but it is complicated by subsections.

- [Discover](#)
 - Objectives
 - [Signals](#)
 - [Sensitivity](#)
 - [Maintaining](#)
 - [Introducing](#)
 - [Words & Actions](#)
 - [Learning](#)
 - [Activities](#)
 - [Active learning](#)
 - [Congratulations!](#)
 - [Video](#)
 - [Activity](#)
 - [Coach Call](#)
- Subsections
- 

In order to give the subsections flexibility each subsection gets 100 pages of room. This makes things simpler in the long run.

- Page Number
- [Discover](#)
 - Objectives 101
 - [Signals](#) 201
 - [Sensitivity](#) 301
 - [Maintaining](#) 401
 - [Introducing](#) 501
 - [Words & Actions](#) 601
 - [Learning](#) 701
 - [Activities](#) 801
 - [Active learning](#) 901
 - [Congratulations!](#) 1001
 - [Video](#) 1201

As you can see the beginning page of each subsection starts at some multiple of 100 plus 1. However after that adding pages to the subsection is as simple as adding 1. Here is a small example of what it might look like in practice.

B	C	D	E	F	G	H	I
Level	Section	Page number	page from	page next	is_left_nav	final_page	title
1	1	101					Welcome
1	1	102					stuff1
1	1	103					stuff2
1	1	201					Signals

To make the subsection page title show up in the Navigation element on the left of the page the is_left_nav property must be set to 1 and pages that aren't the beginning of a subsection should be set to 0.

B	C	D	E	F	G	H	I
Level	Section	Page number	page from	page next	is_left_nav	final_page	title
1	1	101			1		Welcome
1	1	102			0		stuff1
1	1	103			0		stuff2
1	1	201			1		Signals

The next issue to deal with is the final_section_page (labeled final_page in the picture) which ought to be named final_subsection_page but is not because of old code. Basically, if the page is the last page in a subsection set it to 1 and if it isn't leave it blank.

B	C	D	E	F	G	H	I
Level	Section	Page number	page from	page next	is_left_nav	final_page	title
1	1	101			1		Welcome
1	1	102			0		stuff1
1	1	103			0	1	stuff2
1	1	201			1		Signals

To recap, the is_left_nav field, and the final_section_page are used to build the Navigation element on the left of the page. The example would produce the following:

- [Discover](#)
 - Welcome
 - [Signals](#)

All that is left is to fill in the page_from and page_next.

B	C	D	E	F	G	H	I
Level	Section	Page number	page from	page next	is_left_nav	final_page	title
1	1	101			1		Welcome
1	1	102			0		stuff1
1	1	103			0	1	stuff2
1	1	201			1		Signals

This process is a bit like stringing beads on a necklace. The pages are the beads and the page_from and next provide the string. However, that analogy is limited. This concept is actually known as a doubly linked list.

http://en.wikipedia.org/wiki/Doubly-linked_list

Its a very flexible and powerful concept.

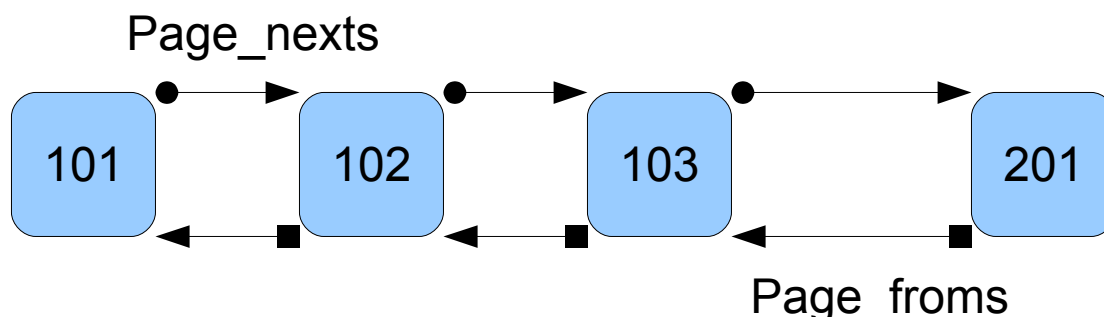
Anyway, to fill in the page_next field for a page you put the page_number of the page you want to go to next from that page. Its fairly self descriptive.

B	C	D	E	F	G	H	I
Level	Section	Page number	page from	page next	is_left_nav	final_page	title
1	1	101		102	1		Welcome
1	1	102		103	0		stuff1
1	1	103		201	0	1	stuff2
1	1	201			1		Signals

Likewise for the page_from put in the page_number just before.

B	C	D	E	F	G	H	I
Level	Section	Page number	page from	page next	is_left_nav	final_page	title
1	1	101		102	1		Welcome
1	1	102	101	103	0		stuff1
1	1	103	102	201	0	1	stuff2
1	1	201	103		1		Signals

Here is a picture of the situation after completing.

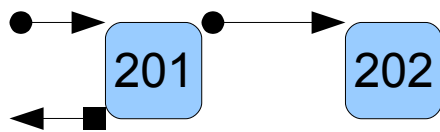


Thats great but what if you want to insert a new page? When working with linked-lists there are 3 situations.

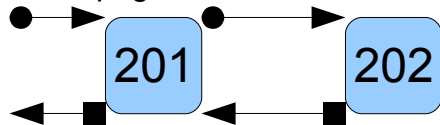
Inserting at the end



For the last page set the page_next to the new page

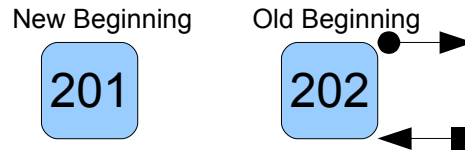


Then set the page_from on the new page

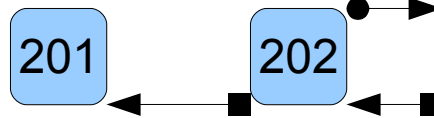


If the old end was or is a final_section_page (has a value of 1) then you must make it 0 and give the new page a value of 1.

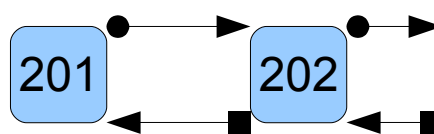
Inserting at the beginning



Set the page_next in the old beginning to the new beginning.

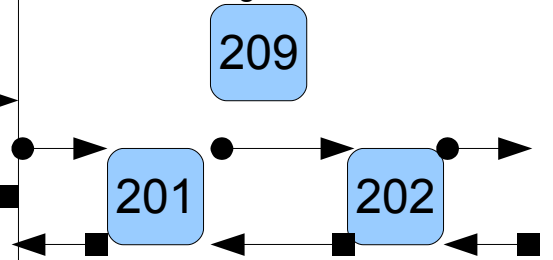


Set the page_next on the new beginning to the old beginning.

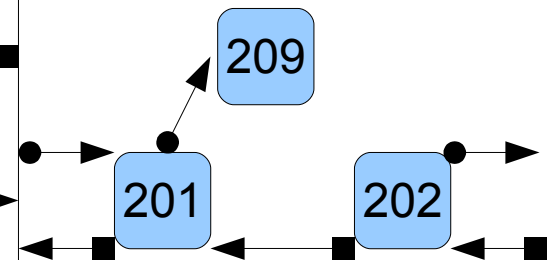


If the old beginning was a left_nav element (value of 1) then change it to 0 and give the new beginning a left_nav value of 1.

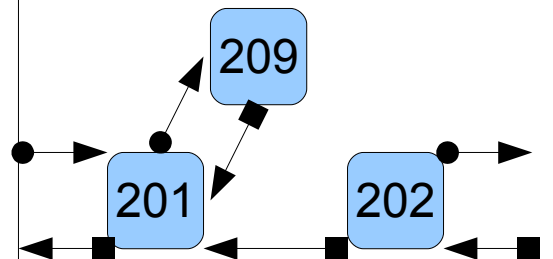
Inserting in the middle



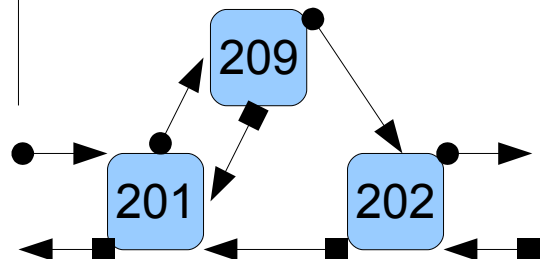
Set the page_next of the first page to the new page.



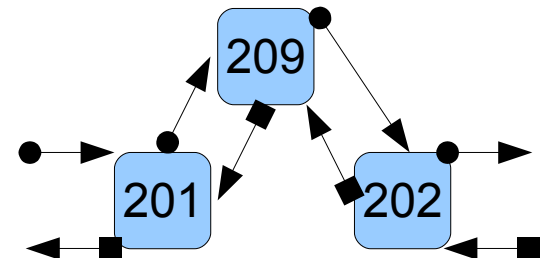
Then set the page_from page of the new page to the first page.



Now set the page_next of the new page to the second page.



Finally set the page_from of the second page to the new page.



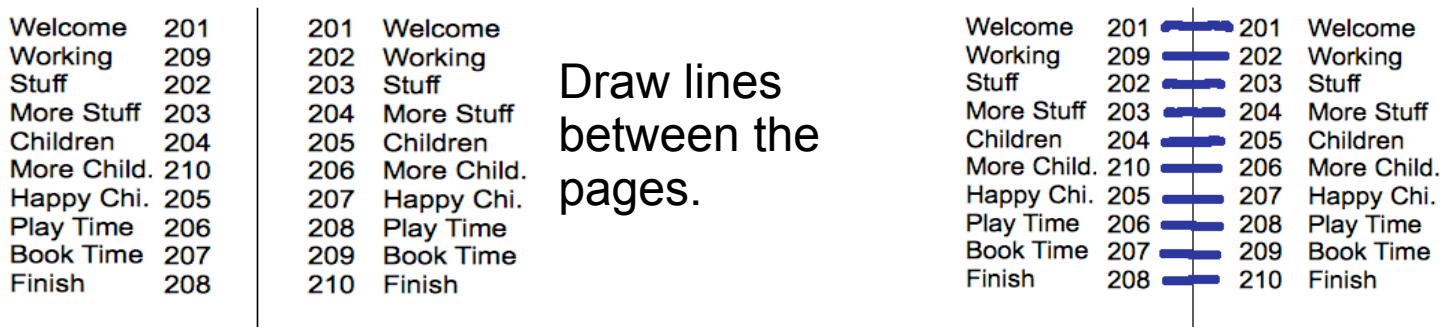
You might be confused by the last example inserting a new page into a section. In particular how 209 could be in between 201 and 202? Strictly speaking a linked list doesn't care about order, its all about the structure of list. 201->209->202 is a valid list. Temporarily this is fine.

For testing purpose this will be very handy. That fact allows us to view the pages and test the pages without being reordered!

Eventually though every subsection must be reordered because of the `furthest_page_viewed` tracking logic expects the pages to be in order. However, reordering the pages is a task that actually only needs to be done once. You may insert and move pages around the program all you want as long as you reorder them once you're all done making edits to a subsection.

How to Re-order a Subsection

Take a piece of paper and write down the page number and titles in the order they are currently in. Then change the numbers to look right.



Now update the `page_number` of every page on the left in the database to be the one on the right. Then go back through and make sure the `page_from` and `page_next` look right.

D	E	F	I
Page number	page from	page next	title
103	102	201	stuff2
201	103	209	Welcome
209	201	203	Working
203	209	204	Stuff

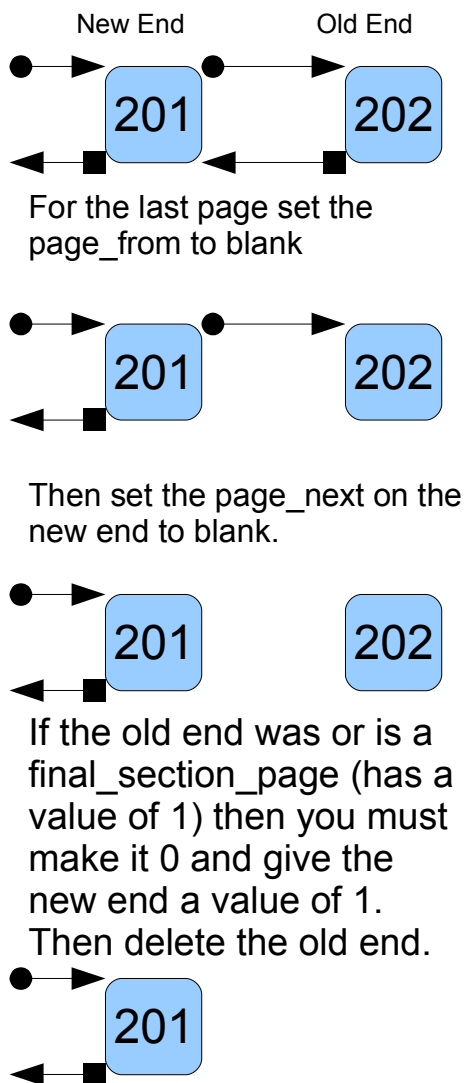
D	E	F	I
Page number	page from	page next	title
103	102	201	stuff2
201	103	202	Welcome
202	201	203	Working
203	202	204	Stuff

Deletion

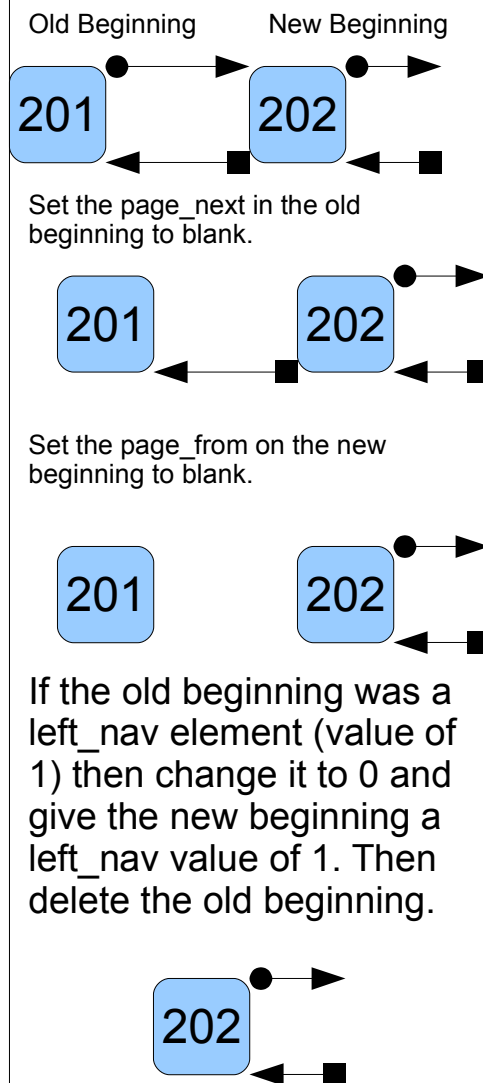
There are certainly cases where removing a page may be necessary. It is important to distinguish between a substitution and an outright deletion. In the case of a substitution the navigation elements need not be changed only the content.

If deletion is called for then there are again 3 situations.

Deleting the end



Deleting the beginning



Deleting from the middle

