



Reusser
Design

WEB. MOBILE. STRATEGY.

JASON BOOTHMAN

- Front-End Architect at Reusser Design
 - Based out of Roanoke, IN - about 2 hours north of Indianapolis
 - Specialize in web development & design, e-commerce, android & iOS app development, content strategy, and more.
- Built my first official website in 1999 as a senior in high school.
- Been developing in ExpressionEngine for about 5 years.



JASON BOOTHMAN

- Married with 3 adorable kids
- Like Star Trek & Star Wars
- Han shot first
- Avid Video Gamer
- Recently switch from iOS to Android
- My favorite number is 12



LET GO AND LET THE USER: GIVING CLIENTS THE CONTROL THEY DESERVE

JASON BOOTHMAN



INTRODUCTION

What if my clients don't deserve control?

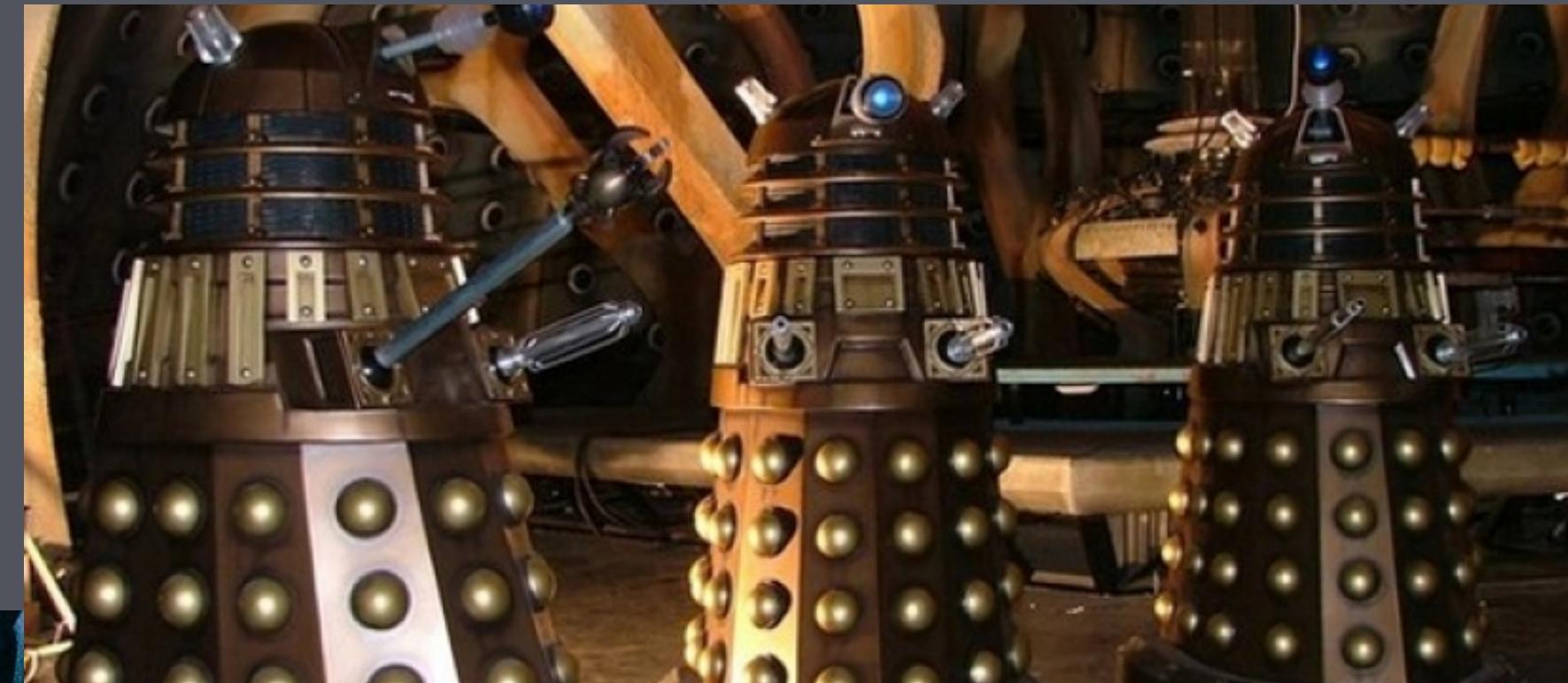
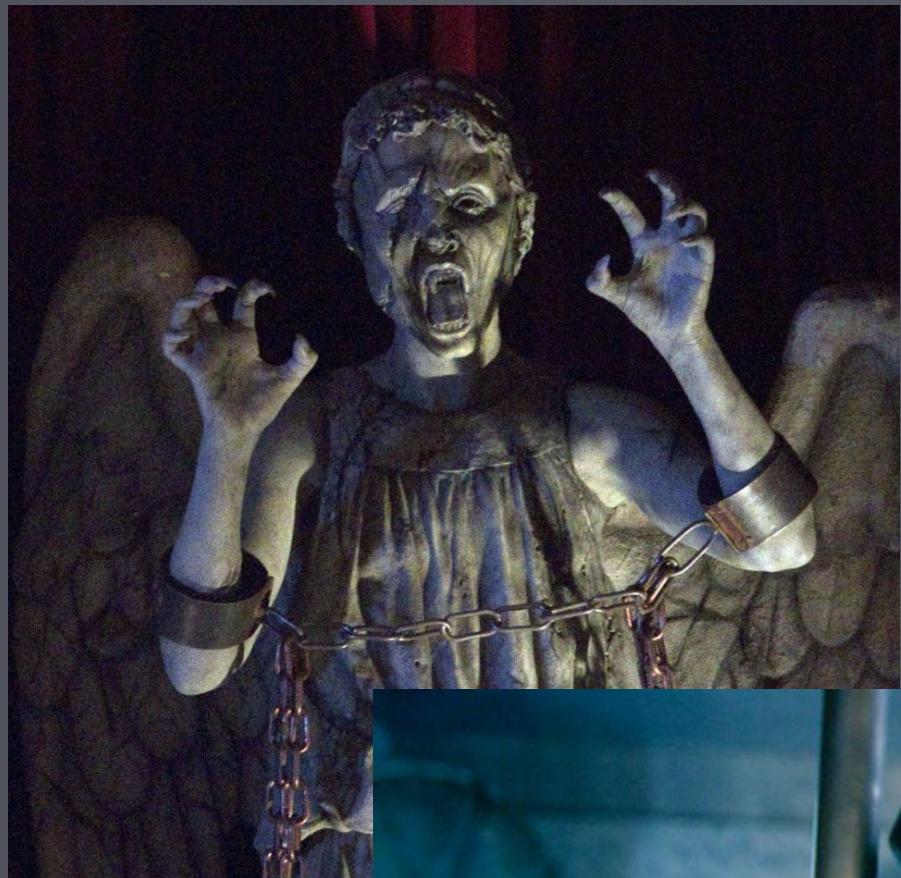


Jason Boothman

INTRODUCTION

WHAT IF MY CLIENTS DON'T DESERVE CONTROL?

You're not alone. We've all felt that way about certain clients.



INTRODUCTION

WHAT IF MY CLIENTS DON'T DESERVE CONTROL?

There are a few reasons why you should want to give every client more control.

- It keeps the “less than ideal” clients from contacting you as often.
- It keeps the “ideal” clients happy and more likely to return for repeat business.
- Ultimately, you should strive to deliver the best product you can, regardless of the client.



INTRODUCTION

WHAT WE'LL LEARN TODAY

- How to create client generated In-Page Navigation.
- How to create SVG Donut Charts on-the-fly.
- How to create a button that allows users to quickly edit the content they are viewing.



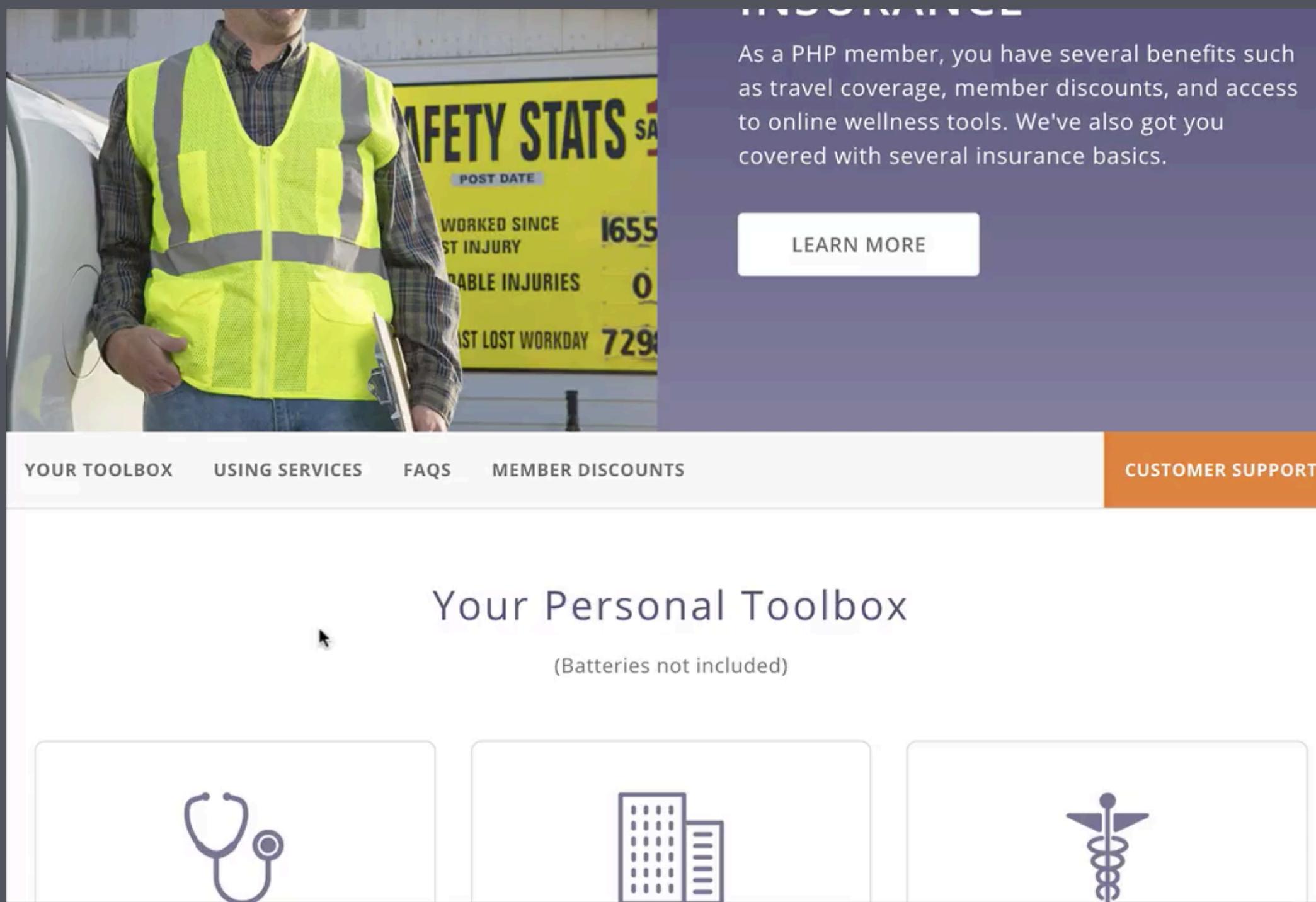
IN-PAGE NAVIGATION



IN-PAGE NAVIGATION

WHAT IS IN-PAGE NAVIGATION?

In-Page Navigation is a navigation bar where a clicked link takes you to a different spot on the page rather than to a separate page altogether.



IN-PAGE NAVIGATION

THE CLIENT NEED

We were building our client a story-driven website, and they wanted the ability for a user to quickly be able to navigate throughout the page.

Our client had to have the ability to create in-page navigation on-the-fly, without outside help, while still maintain a high level of design in its implementation.



IN-PAGE NAVIGATION

THE OLD WAY: HARD-CODED

- Can be highly stylized
- Not flexible
- Not forward-thinking or future proof
- Clients don't have any control
- Requires a developer for any changes



Clients usually aren't happy with this solution as it costs them money every time they want to make a change.

HARD-CODED



IN-PAGE NAVIGATION

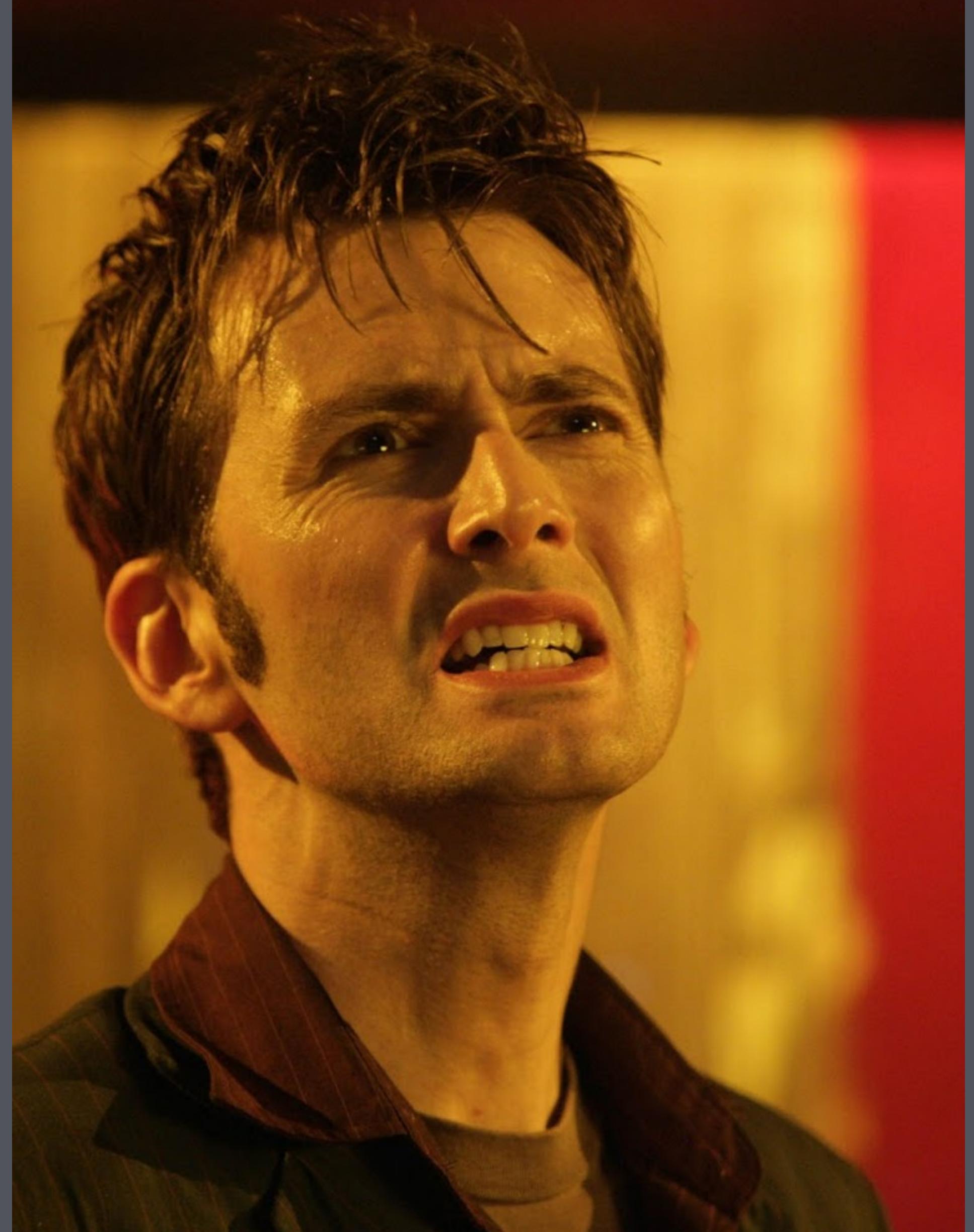
THE OLD WAY: ANCHOR TAGS INSIDE WYSIWYG

- Clients have control over the links and navigation
- Not highly customizable
- It can be kludgy and annoying to implement (for the developer and the client)
- It can be used anywhere - even in places you might not want



Clients usually don't like this solution because it can be confusing to implement and often comprises the design.

ANCHOR TAGS IN WYSIWYG



WE FOUND A BETTER WAY



IN-PAGE NAVIGATION

THE BETTER WAY

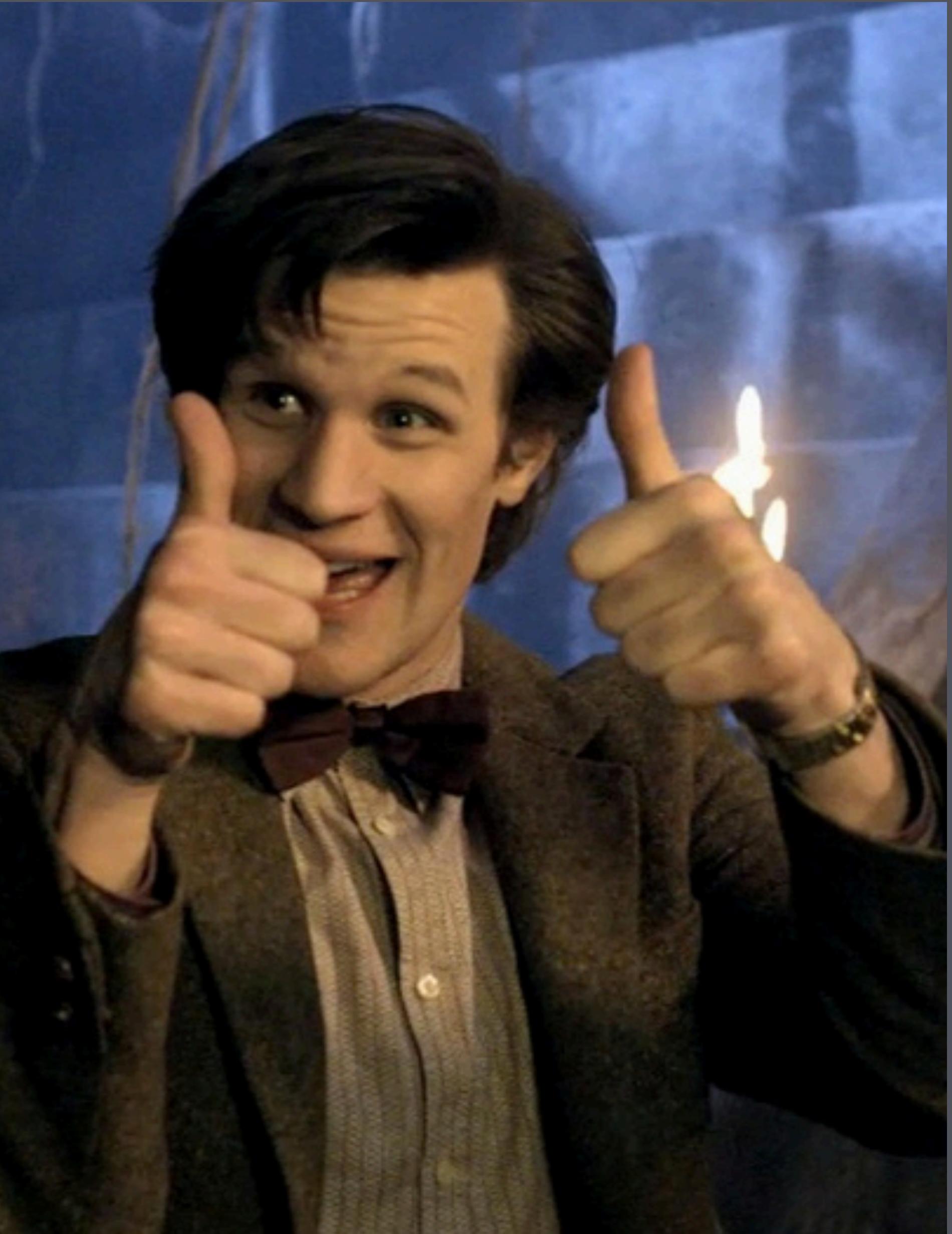
Dynamically building the navigation leveraging ExpressionEngine add-ons Bloqs and Stash.

- Can be highly stylized
- Easy to implement
- Gives the client control over the navigation
- Does not require a developer to make updates



Clients always like this solution because it makes it easy for them to build in-page navigation while still allowing it to be highly stylized.

DYNAMICALLY BUILT NAVIGATION



IN-PAGE NAVIGATION

THE PROCESS

In ExpressionEngine

- Implement required Bloqs for the page/content
- Add a 'Navigation Title' field to each Bloq that the client should be able to link

In your template

- Create links and add them to a Stash list
- Use the Stash list to build the in-page navigation anchor tags



IN-PAGE NAVIGATION

IMPLEMENT REQUIRED BLOQS

Add-On Manager / Bloqs Configuration

Block Types

Block Type	Short Name	Manage
Callout	callout_slab	
Contact	contact_slab	
Content	content_slab	
Content w/Media	content_w_media_slab	
Get Started	get_started_slab	
Images - 2 Columns	images_2_columns_slab	
Images - Up to 2 Wide	images_up_to_2_wide_slab	
Images - Up to 4 Wide	images_up_to_4_wide_slab	
'Learn More' Group	learn_more_group_slab	
'Learn More' Group Slide	learn_more_group_slide_slab	

IN-PAGE NAVIGATION

ADD A “NAVIGATION TITLE” FIELD

The screenshot shows a form configuration interface with the following fields:

- Type:** Text Input (selected)
- Name ***: Navigation Title
- Short Name ***: navigation_title
- Instructions:** Field instructions that appear in the publish form.
Optional: Used to add a link to the in-page navigation.
- Require field?** (Toggle switch)

IN-PAGE NAVIGATION

CREATE LINKS AND THE STASH LIST

Here is a glimpse at the code we will be looking at one feature at a time.

```
{content_slab}
  {if navigation_title}
    <a name="{exp:ce_str:ing url_title}{navigation_title}{/exp:ce_str:ing}"></a>

    {exp:stash:append_list name="navigation" parse_tags="yes"}
      {stash:item_title}{navigation_title}{/stash:item_title}
      {stash:item_link}
        {exp:ce_str:ing url_title parse="inward"}{navigation_title}{/exp:ce_str:ing}
      {/stash:item_link}
    {/exp:stash:append_list}
  {/if}
<section>
  <div class="max_content_width">
    <section class='slab content'>
      {body}
    </section>
  </div>
</section>
{/content_slab}
```

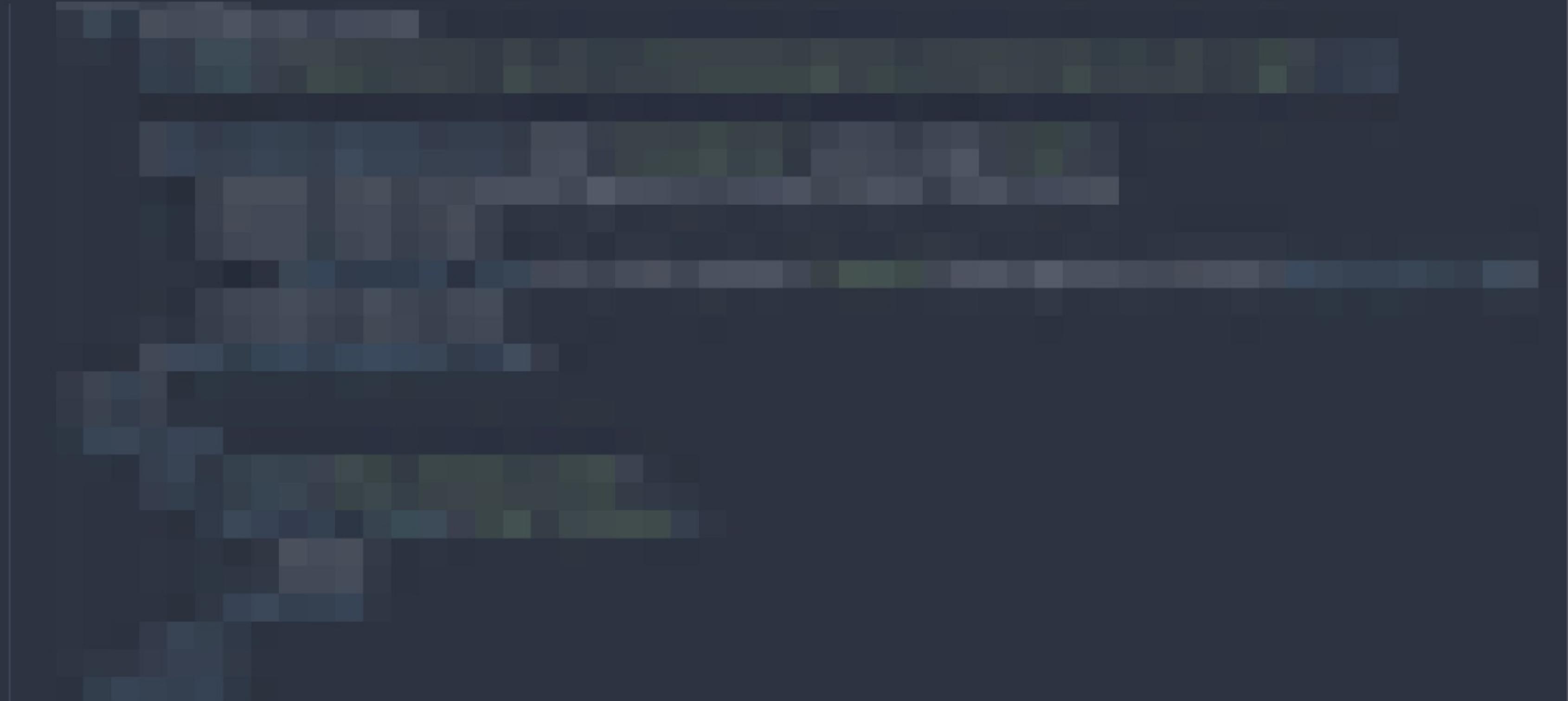


IN-PAGE NAVIGATION

CREATE LINKS AND THE STASH LIST

Open the bloq
field.

{content_slab}



{/content_slab}



IN-PAGE NAVIGATION

CREATE LINKS AND THE STASH LIST

Check to see if the user filled out the 'Navigation Title' field.

```
{content_slab}
  {if navigation_title}
    [REDACTED]
  {/if}
{/content_slab}
```



IN-PAGE NAVIGATION

CREATE LINKS AND THE STASH LIST

Use the CE String add-on to convert the inputted title to a url title to ensure it is 'link friendly'.

```
{content_slab}
  {if navigation_title}
    <a name="{exp:ce_string url_title}{navigation_title}{/exp:ce_string}"></a>
  {/if}
{/content_slab}
```



IN-PAGE NAVIGATION

CREATE LINKS AND THE STASH LIST

Here is a glimpse at the code we will be looking at one feature at a time.

```
{content_slab}
  {if navigation_title}
    <a name="{exp:ce_str:ing url_title}{navigation_title}{/exp:ce_str:ing}"></a>

    {exp:stash:append_list name="navigation" parse_tags="yes"}
      {stash:item_title}{navigation_title}{/stash:item_title}
      {stash:item_link}
        {exp:ce_str:ing url_title parse="inward"}{navigation_title}{/exp:ce_str:ing}
        {/stash:item_link}
      {/exp:stash:append_list}
    {/if}

  {/content_slab}
```



IN-PAGE NAVIGATION

CREATE LINKS AND THE STASH LIST

Now that the stash list has been set up, display your content as you normally would.

```
{content_slab}
  {if navigation_title}
    <a name="{exp:ce_str:ing url_title}{navigation_title}{/exp:ce_str:ing}"></a>

    {exp:stash:append_list name="navigation" parse_tags="yes"}
      {stash:item_title}{navigation_title}{/stash:item_title}
      {stash:item_link}
        {exp:ce_str:ing url_title parse="inward"}{navigation_title}{/exp:ce_str:ing}
      {/stash:item_link}
    {/exp:stash:append_list}
  {/if}
  <section>
    <div class="max_content_width">
      <section class='slab content'>
        {body}
      </section>
    </div>
  </section>
{/content_slab}
```



BUILD THE NAVIGATION

Parse through the stash list we just created to create the navigation menu.

```
{layout:set name="navigation"}  
  <div class="page_links">  
    {exp:stash:get_list name="navigation" process="end"}  
      <a href="#{item_link}">{item_title}</a>  
    {/exp:stash:get_list}  
  </div>  
{/layout:set}
```



BUILD THE NAVIGATION

Make sure to use the 'process="end"' tag when calling your stash list so you call it AFTER it has been created.

```
{layout:set name="navigation"}  
  <div class="page_links">  
    {exp:stash:get_list name="navigation" process="end"}  
      <a href="#{item_link}">{item_title}</a>  
    {/exp:stash:get_list}  
  </div>  
{/layout:set}
```



IN-PAGE NAVIGATION

THE FINAL PRODUCT

SAFETY STATS

POST DATE

WORKED SINCE LAST INJURY 1655

DABLE INJURIES 0

LAST LOST WORKDAY 729

YOUR TOOLBOX USING SERVICES FAQS MEMBER DISCOUNTS CUSTOMER SUPPORT

MEMBER DISCOUNTS

As a PHP member, you have several benefits such as travel coverage, member discounts, and access to online wellness tools. We've also got you covered with several insurance basics.

LEARN MORE

CUSTOMER SUPPORT

Your Personal Toolbox

(Batteries not included)

Customer Support

Healthcare

Business

Wellness

IN-PAGE NAVIGATION

ONE THING TO NOTE

Alternatively, Fluid Fields could be used, but I chose Bloqs because...

- Bloqs gives me more control over what content types can have a navigation item.
- Fluid Fields were not available when I developed this solution.



THOUGHT BREAK

Never stop looking for new ways to implement old features. Just because we've always done something a certain way doesn't mean we always should.

Jason Boothman



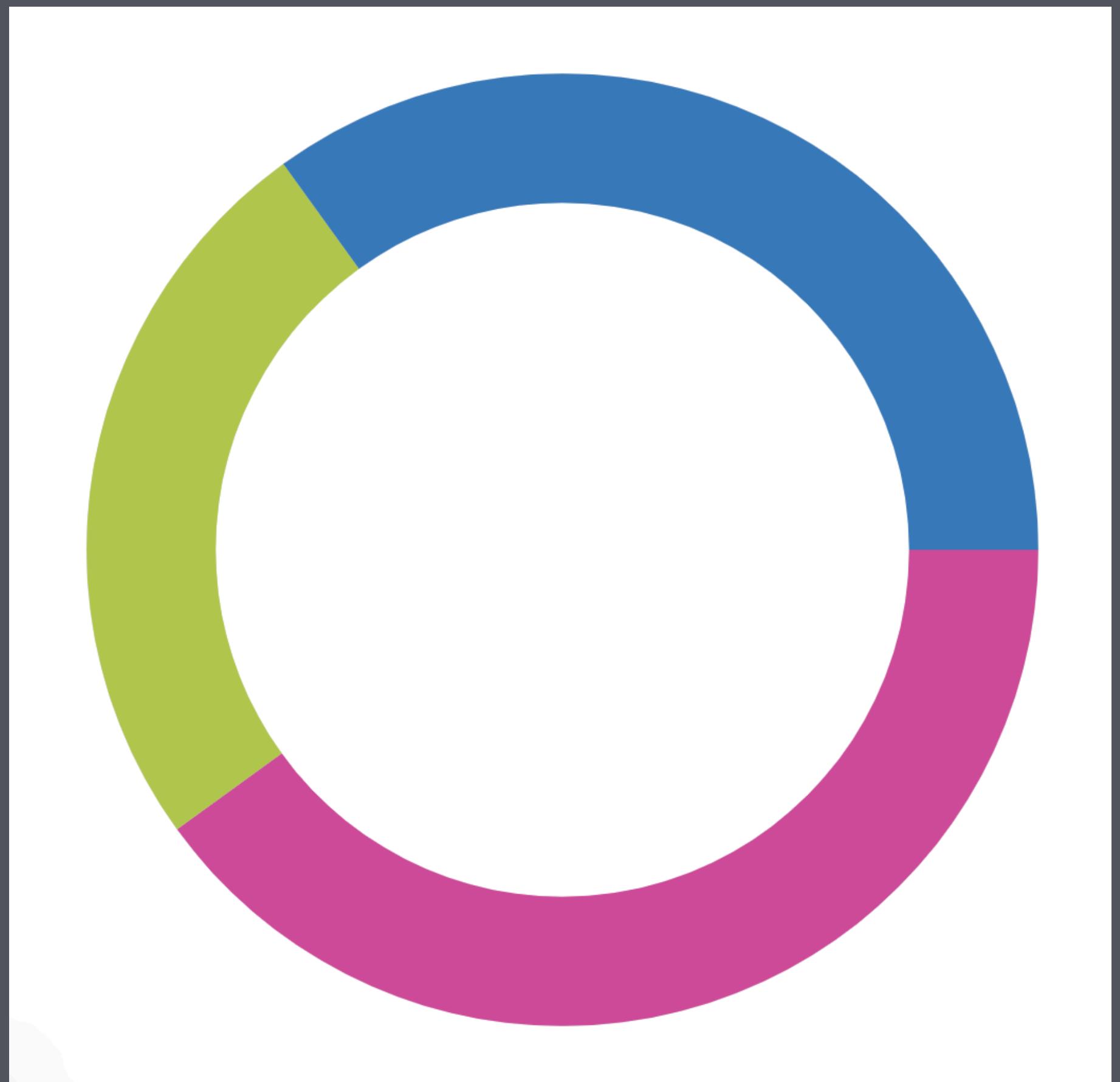
SVG DONUT CHARTS



SVG DONUT CHARTS

WHAT IS AN SVG DONUT CHART?

Simply put - a donut chart...in the SVG
(scalable vector graphic) format.



SVG DONUT CHARTS

THE CLIENT NEED

As a non-profit organization, our client wanted to be able to dynamically include annual reports on their new website.

Seeing as how they don't have a designer in-house, we had to develop a way for them to be able to build their own donut charts with nothing but the data.



SVG DONUT CHARTS

THE OLD WAY: DONUT CHARTS AS IMAGES

- Can be highly stylized
- Clients don't have control over the chart
- Not flexible
- Not future proof
- Requires a designer for changes



Clients usually don't like this solution as it gives them no control and costs money for updates.

DONUT CHARTS AS IMAGES



**WE FOUND A BETTER WAY
...WITH HELP**



SVG DONUT CHARTS

THE HELP

Medium.com ran an article in late 2016 titled “Scratch-made SVG Donut & Pie Charts in HTML5” which showed how to manually create an SVG pie. I took that concept and modified it to work dynamically leveraging ExpressionEngine fields.

Scratch-made SVG Donut & Pie Charts in HTML5

Mark Caron

<https://medium.com/@heyoka/scratch-made-svg-donut-pie-charts-in-html5-2c587e935d72>



SVG DONUT CHARTS

THE BETTER WAY

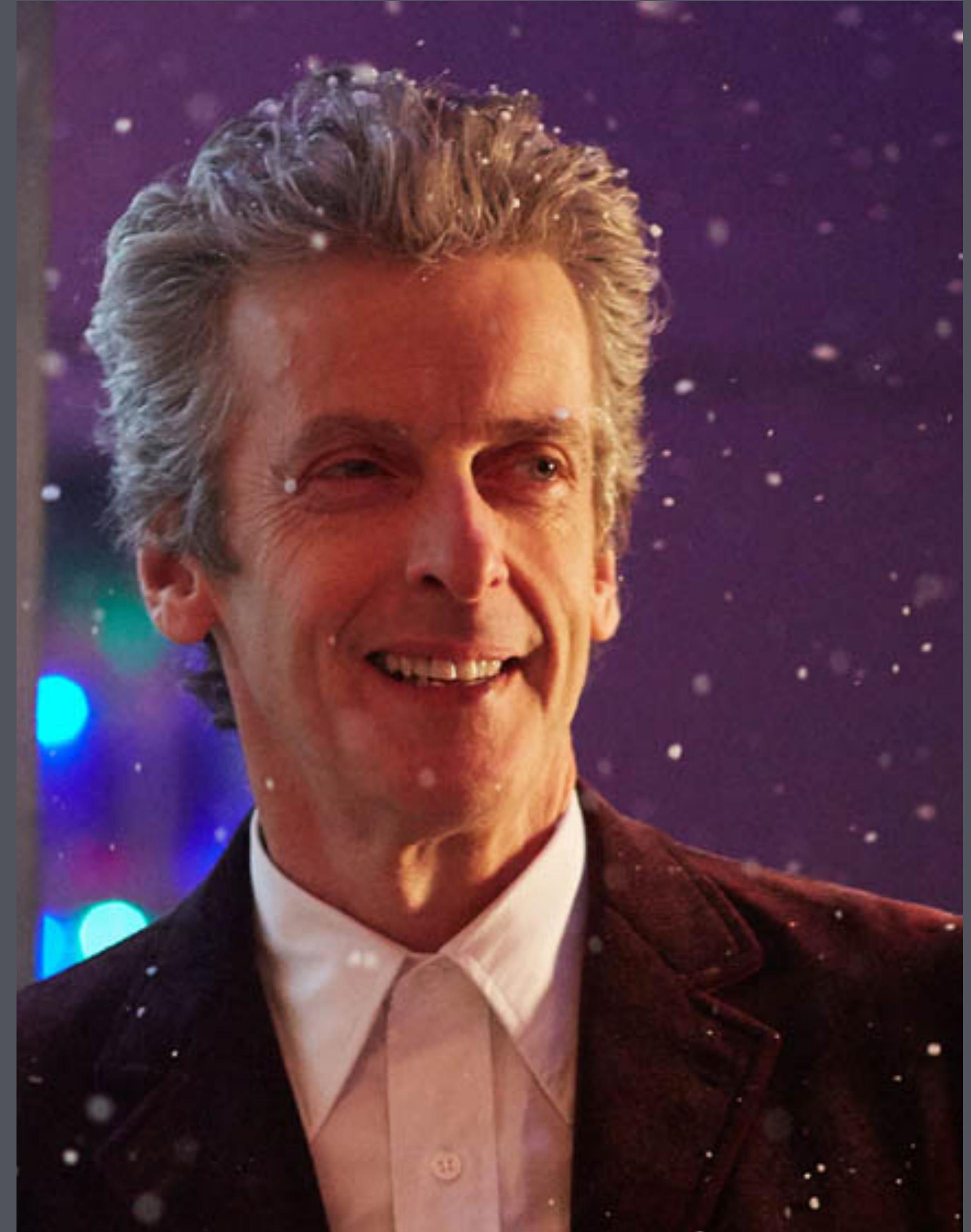
Dynamically building the donut chart using the Grid field, PHP, and fun maths.

- Not as hard to implement as it sounds
- Gives the client control over building a custom pie chart
- Does not require a developer to make updates



Clients always like this solution because it makes it is easy to maintain and doesn't cost them extra money to update.

DYNAMICALLY BUILT DONUT CHART



SVG DONUT CHARTS

THE PROCESS

In ExpressionEngine

- Create a grid field to support the data

In your template

- Pass the grid data to a PHP file
- Loop through the data set
- Do fun maths
- Display the HTML on the page



MAKING THE CHART BY HAND

To start, lets make an SVG circle. We are going to assume our circle has a circumference of 100 to give us an easy starting point.

- circle radius = Circumference / $2 * \pi$ = $100 / 6.28318530718 = 15.91549430918954$
- cx & cy attributes are the center of the circle - slightly larger than the radius = 21
- SVG viewBox double cx & cy = 42



SVG DONUT CHARTS

MAKING THE CHART BY HAND

Let's look at the value in the previous slide plugged into the code.

```
<svg width="100%" height="100%" viewBox="0 0 42 42" class="donut">
  <circle class="donut-ring" cx="21" cy="21" r="15.91549430918954"
    |   fill="transparent" stroke="#d2d3d4" stroke-width="5"></circle>
  <circle class="donut-segment" cx="21" cy="21" r="15.91549430918954"
    |   fill="transparent" stroke="#cd4b99" stroke-width="5"></circle>
</svg>
```

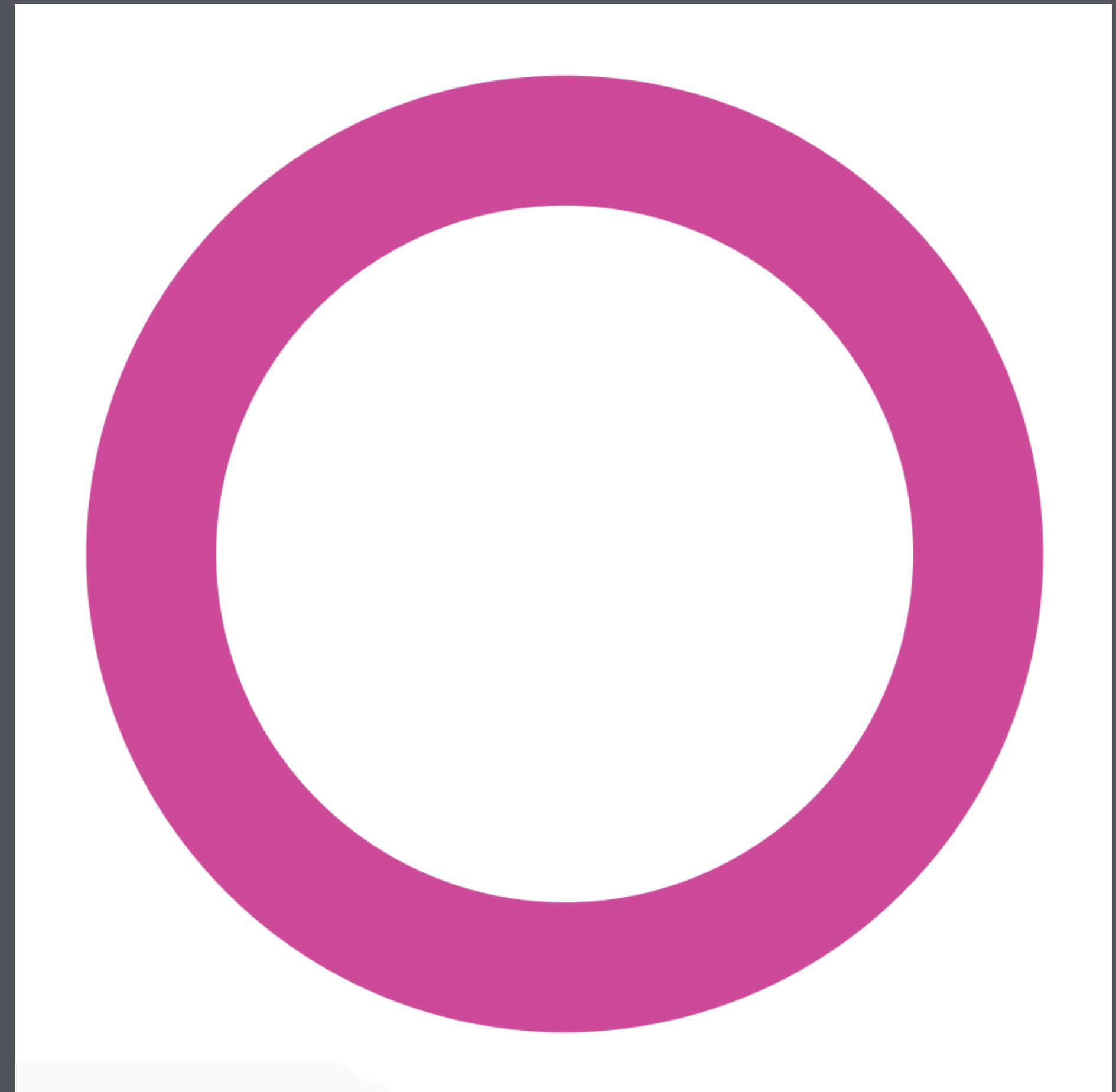


SVG DONUT CHARTS

MAKING THE CHART BY HAND

Given the values in the previous slide, we should end up with an SVG circle that looks something like this.

**Keep in mind that while we chose a circumference of 100, since SVGs are scalable, it doesn't really matter how big we end up making them in the end - it will always look pretty.*



MAKING THE CHART BY HAND

Next, lets take that segment we just made and have it only cover 85% of the chart by setting the *stroke-dasharray* attribute - a two value array delimited by a space.

- The *stroke-dasharray* value is set to '85 15'.
- This will draw a dashed stroke for 85 units and then a gap for 15.
- Since we're using a circle whose value totals 100, it will just loop around once.



MAKING THE CHART BY HAND

Let's look at the value in the previous slide plugged into the code.

```
<svg width="100%" height="100%" viewBox="0 0 42 42" class="donut">
  <circle class="donut-ring" cx="21" cy="21" r="15.91549430918954"
    fill="transparent" stroke="#d2d3d4" stroke-width="5"></circle>

  <circle class="donut-segment" cx="21" cy="21" r="15.91549430918954"
    fill="transparent" stroke="#cd4b99" stroke-width="5"
    stroke-dasharray="85 15"></circle>
</svg>
```

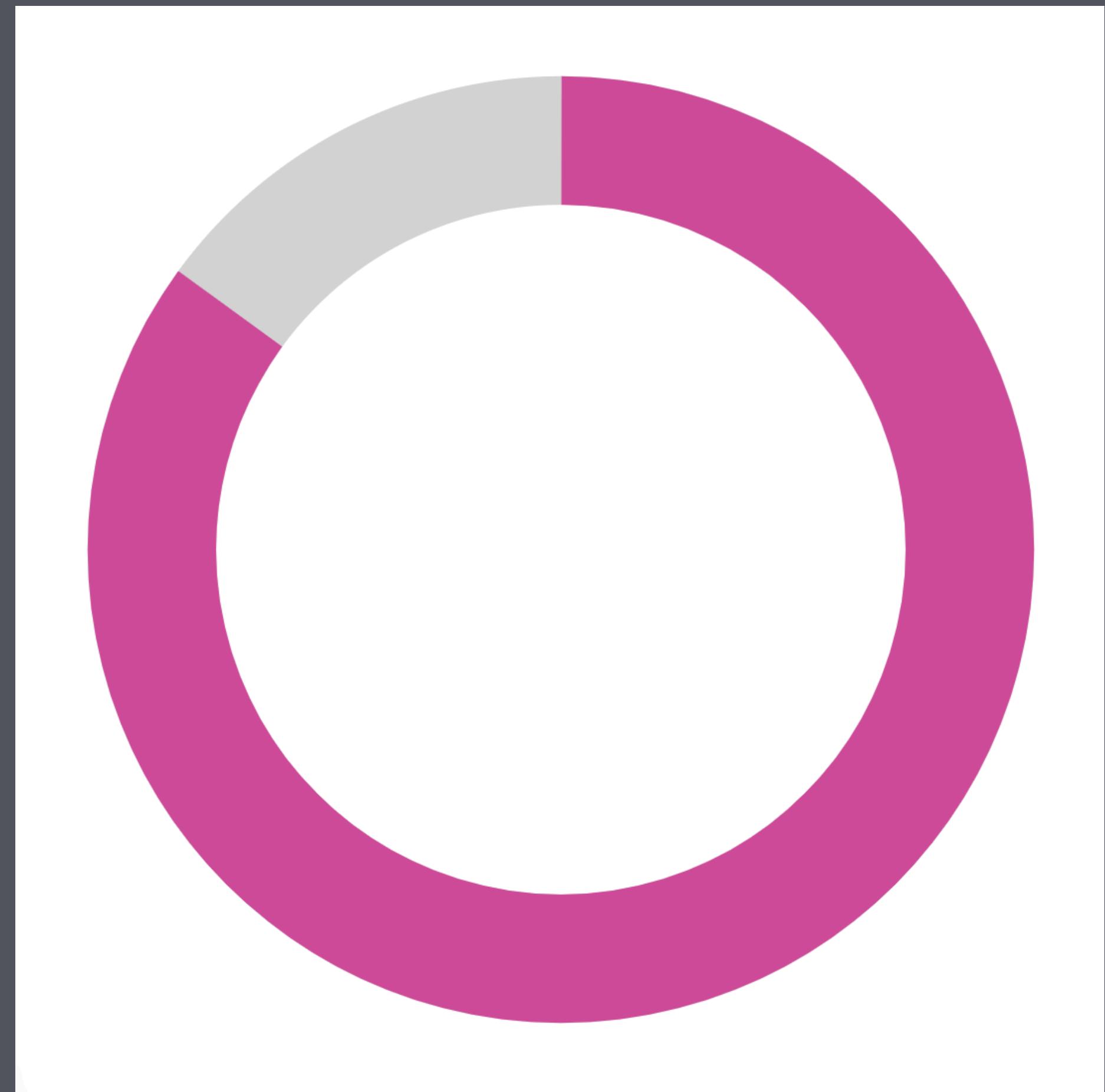


SVG DONUT CHARTS

MAKING THE CHART BY HAND

Given the values in the previous slide, we should end up with an SVG that looks something like this.

Our segment now takes up 85% of the circle and we can see the 'donut-ring' backdrop that we created showing through.



MAKING THE CHART BY HAND

Finally, lets add a second segment to fill the remaining gap.

- The *stroke-dasharray* value is set to '15 85'.
- This will draw a dashed stroke for 15 units and then a gap for 85.
- Since we don't want the segments to overlap, we need to give the new segment a *stroke-dashoffset*.
- New Segment Offset = Circumference - all preceding segments total length = $100 - 85 = 15$.



SVG DONUT CHARTS

MAKING THE CHART BY HAND

Let's look at the value in the previous slide plugged into the code.

```
<svg width="100%" height="100%" viewBox="0 0 42 42" class="donut">
  <circle class="donut-ring" cx="21" cy="21" r="15.91549430918954"
    fill="transparent" stroke="#d2d3d4" stroke-width="5"></circle>

  <circle class="donut-segment" cx="21" cy="21" r="15.91549430918954"
    fill="transparent" stroke="#cd4b99" stroke-width="5"
    stroke-dasharray="85 15"></circle>
  <circle class="donut-segment" cx="21" cy="21" r="15.91549430918954"
    fill="transparent" stroke="#b1c94e" stroke-width="5"
    stroke-dasharray="15 85"></circle>
</svg>
```

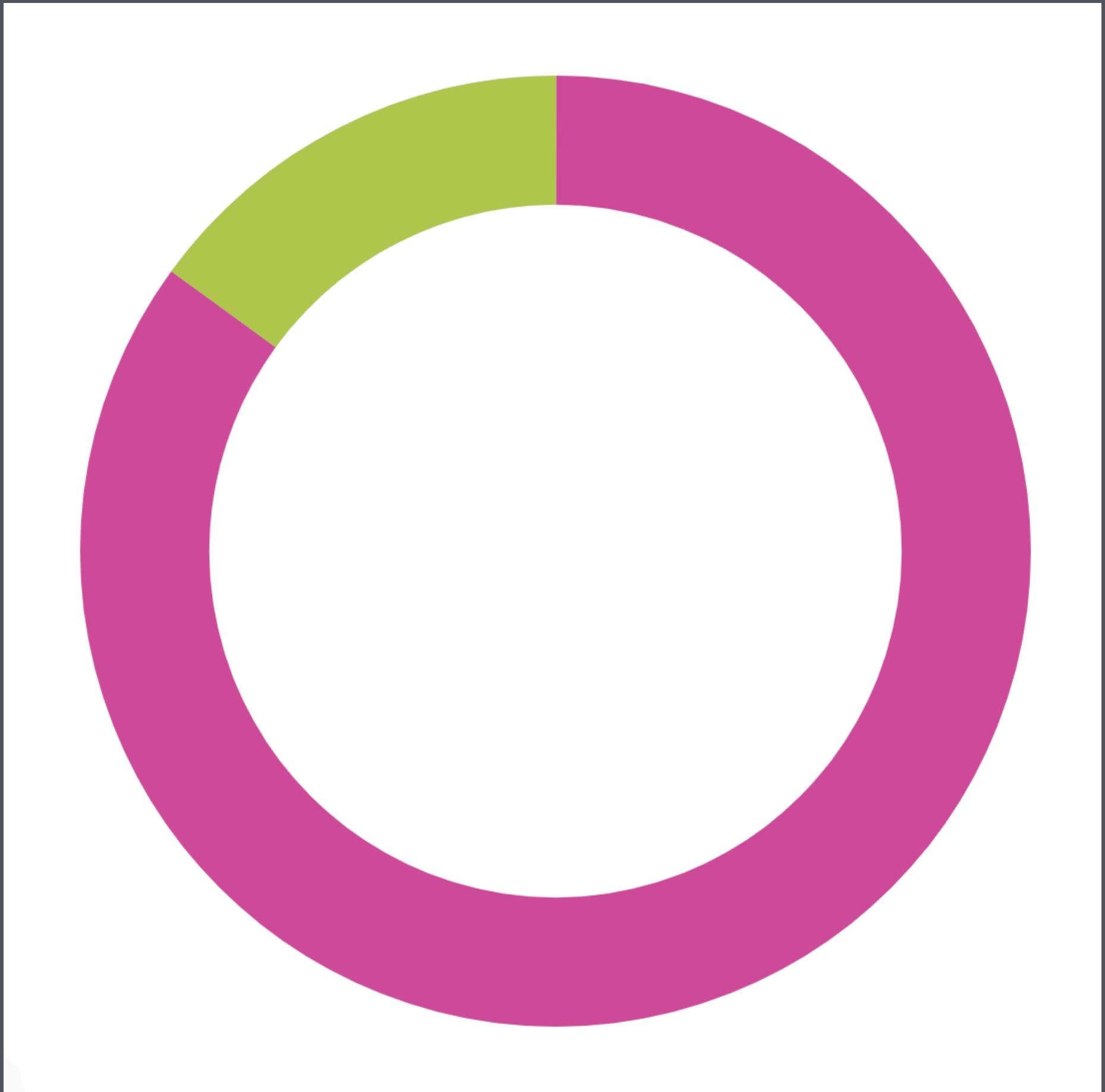


MAKING THE CHART BY HAND

Given the values in the previous slide, we should end up with a donut chart that looks something like this.

The new segment we added takes up the remaining 15% of space and starts where the previous segment left off.

Now that we know the math behind it, adding additional segments is pretty easy to do.



LET'S MAKE THIS THING DYNAMIC!



SVG DONUT CHARTS

IN EXPRESSIONENGINE

- Make a grid field named 'Donut Chart'
- Add a text field titled 'Data Label'
- Add a number field titled 'Data Value'.



New Field

Type

Grid

Name *

Donut Chart

Short name *

No spaces. Underscores and dashes are allowed.

donut_chart

Grid Fields

+ Data Label (text)

+ Data Value (no punctuation) (text)

Save Save & New Sa

SVG DONUT CHARTS

IN THE TEMPLATE

- Pass data to our PHP file
- We are using RD PHP add-on, but could also use an embed or custom add-on.

```
<div class="chart">
| {exp:rd_easy_php file="assets/php/chart.php"
|   chart_data="{donut_chart}{donut_chart:data_label}||{donut_chart:data_value},{/donut_chart}"}
</div>
```



SVG DONUT CHARTS

THE PHP

Set an array of colors for the donut segments.

- Colors can be Hex, RGB, RGBa, Named, HSL or HSLa values.
- Make sure you have enough colors to cover the max number of donut chart segments.
- In this example, the client is limited to 10 segments in a donut chart.



```
$colors = array(  
    '#004f98',  
    '#599dd6',  
    '#c3102f',  
    '#ffb229',  
    '#f76b00',  
    '#939598',  
    '#012c62',  
    '#c3102f',  
    '#f4901d',  
    '#ffffff'  
) ;
```

THE PHP

Put data into an array we can loop through to build the segments.

- Explode data sent from our EE template to the PHP file using the comma delimiter and add to the \$data array.
- Loop through each row in the \$data array to set segment values.



```
$data = explode(',', , $template_vars['chart_data']);

foreach ($data as $val) {
    list($label, $amount) = explode('||', $val);

    $temp['label'] = $label;
    $temp['amount'] = $amount;
    $temp['segment_percentage'] = 0;
    $temp['remaining_percentage'] = 0;
    array_push($chart_data, $temp);

    $total_amount += $temp['amount'];
}
```

SVG DONUT CHARTS

THE PHP

Set individual segment data.

```
foreach ($chart_data as $i => &$row) {  
    $segment_percentage = "";  
    $segment_percentage = (($row['amount'] / $total_amount) * 100);  
  
    $row['segment_percentage'] = $full_percentage;  
    $row['remaining_percentage'] = 100 - $full_percentage;  
}
```



SVG DONUT CHARTS

THE PHP

Use the segment data we just set to build the SVG segments and adjust the offset.

```
$segments = "";
$filled_offset = "";
$stroke_offset = "";

foreach ($chart_data as $i => &$row) {

    $segments .= '<circle class="donut-segment" cx="21" cy="21" r="15.91549430918954"
fill="transparent" stroke-width="5"
stroke="' . $colors[$i] . '"'
stroke-dasharray="' . $row["segment_percentage"] . ' ' . $row["remaining_percentage"] . '"'
stroke-dashoffset="' . $stroke_offset . '"></circle>';

    $filled_offset += $row["full_percentage"];
    $stroke_offset = (100 - $filled_offset);
}
```



SVG DONUT CHARTS

THE PHP

Build the SVG code for the donut chart.

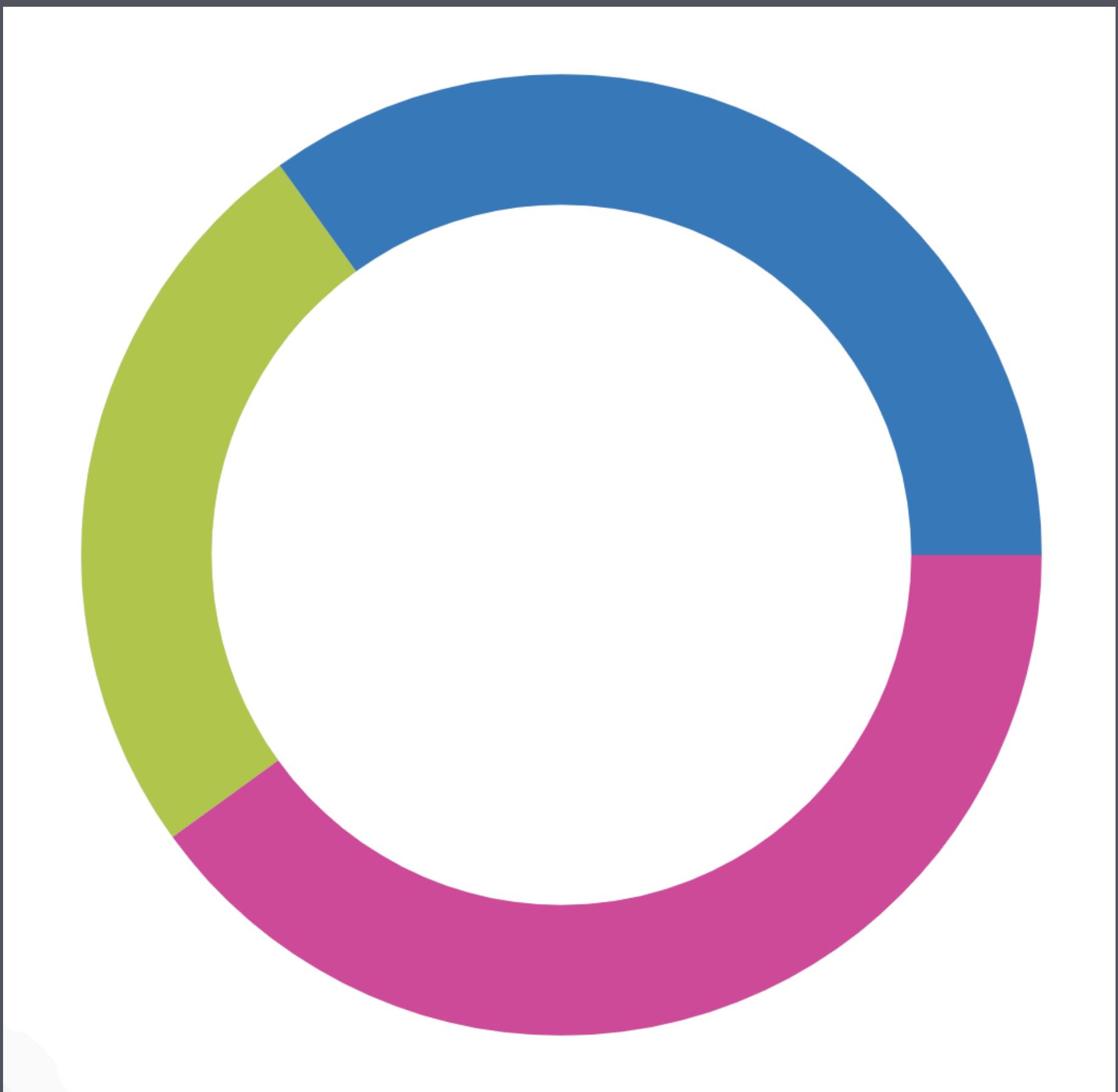
```
$html = '';  
  
$html = '<svg width="100%" height="100%" viewBox="0 0 42 42" class="donut">  
    <circle class="donut-ring" cx="21" cy="21" r="15.91549430918954"  
        fill="transparent" stroke="#d2d3d4" stroke-width="5"></circle>';  
  
$html .= $segments;  
  
$html .= '</svg>';  
  
return $html;
```



SVG DONUT CHARTS

THE FINAL PRODUCT

And that's it. Your SVG code will be automatically built out and displayed on the page.



THOUGHT BREAK

It's okay to use previously developed methods and modify them for your own purposes.

Jason Boothman



THE 'EDIT' BUTTON



THE EDIT BUTTON

WHAT ARE WE DOING?

Creating a button that allows users to quickly edit the content they are viewing.



Spirituality Books Authors New Sale

Edit

cred Reading
2019 Guide to Daily Prayer
or: Pope's Worldwide Prayer Network

6.95

[Table of Contents](#)
[Read a Sample](#)
Format: Paperback
Publication date: August 3, 2018

Quantity: **ADD TO CART**

Buy available from

THE EDIT BUTTON

THE CLIENT NEED

With hundreds of products and multiple channels, our clients were finding it cumbersome to find the entry they wanted to edit within ExpressionEngine. It's not even that it was necessarily difficult, but most the time they were browsing the website when they saw something needed changed and were unhappy with the extra step of having to find that product again within the CMS.

With this in mind, we needed to devise a way for them to be able to easily, and quickly, edit an entry that they had browsed to on the front-end of the website.



THE EDIT BUTTON

THE ENTRY TEMPLATE

Create a layout set named edit_button_id which includes the entry_id of the entry you want the user to be able to quickly access from the website.



```
{layout="site/_wrapper"}  
  
{exp:channel:entries  
    channel="page"  
    limit="1"  
    require_entry="yes"  
    disable="categories|category_fields|member_data|pagination"}  
  
{layout:set name="title"}{title}{/layout:set}  
{layout:set name="edit_button_id"}{entry_id}{/layout:set}  
  
{tp_repeable_content}  
  
{if no_results}  
    {redirect="404"}  
{/if}  
{/exp:channel:entries}
```

THE EDIT BUTTON

THE WRAPPER TEMPLATE

If the layout:set exists and the member is logged in to the appropriate member group, display the edit button with a quick link to the admin.

```
{if (logged_in_group_id == '1'  
    || logged_in_group_id == '6')  
    && layout:edit_button_id}  
  
    <a href="/admin.php/cp/publish/edit/entry/{layout:edit_button_id}"  
        target="blank"  
        class="edit-entry-button">Edit</a>  
  
{/if}
```



THE EDIT BUTTON

THE CSS

Style the button as you see fit.

- Set the *position* to 'fixed' to make sure it always stays on the screen.
- Use the *transform* property to rotate the button & text so it doesn't stick out into the page as much.



```
.edit-entry-button {  
    background-color: #f15a22;  
    color: #fff;  
    border-radius: .375rem .375rem 0 0;  
    position: fixed;  
    top: 0;  
    right: 0;  
    padding: .5rem 1rem;  
    transform: rotate(-90deg);  
    transform-origin: 100% 100%;  
    z-index: 9999;  
}
```

THE EDIT BUTTON

THE FINAL PRODUCT

A button that allows users to quickly edit the content they are viewing.

This could also be modified to work on pages where there are multiple entries, such as news listings, products pages, etc.

Put this on every site!



Spirituality Books Authors New Sale

Edit

cred Reading
2019 Guide to Daily Prayer
or: Pope's Worldwide Prayer Network

6.95

[Table of Contents](#)
[Read a Sample](#)

Format: Paperback
Publication date: August 3, 2018

Quantity: **ADD TO CART**

Buy available from

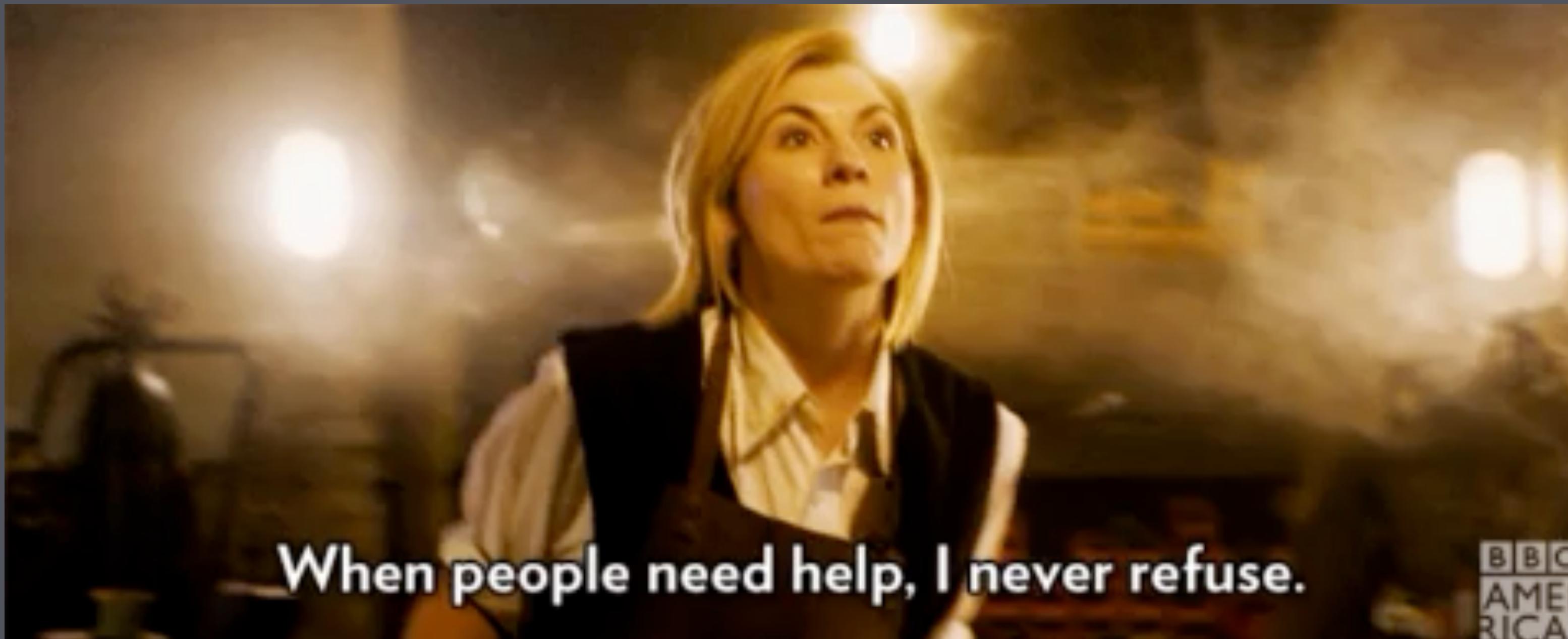
THOUGHT BREAK

The greatest innovations come out of necessity, not a desire to deliver the best product.

Jason Boothman



QUESTIONS?



When people need help, I never refuse.

BBC
AMERICA



LET GO AND LET THE USER: GIVING CLIENTS THE CONTROL THEY DESERVE

THANK YOU FOR YOUR TIME

<http://rdurl.io/letgo2018>

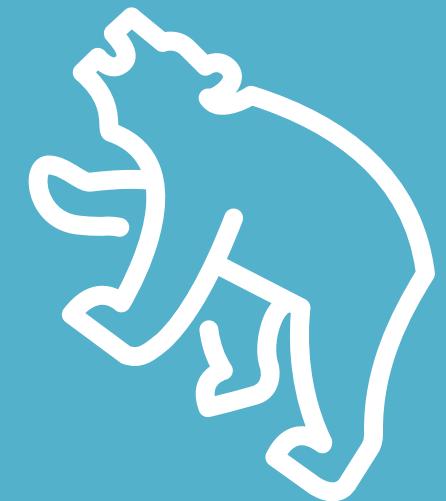
Jason Boothman

jason@reusserdesign.com // @JasonBoothman

Nick Johnson

nick@reusserdesign.com // @thebrawnyman





Thank you!

260.676.2261 // reusserdesign.com // @reusserdesign

Reusser Design
150 S. Main Street // Roanoke, IN 46783