

USING LAVA SHORTCODES

...to Build a Volunteer
Onboarding Portal

Brent Pirolli

CEDARCREEKCHURCH

WHO'S THIS CLOWN?

- Brent Pirolli (BrentP@CedarCreek.tv)
- Senior Director of I.T. at CedarCreek Church
- We have 6 campuses in NW Ohio currently
- On staff for nearly 22 years
- Started as a video editor/production team → I.T. in 2002
- Got into website construction in 2001, almost exclusively using WordPress for web design since 2011... which is where I learned about...

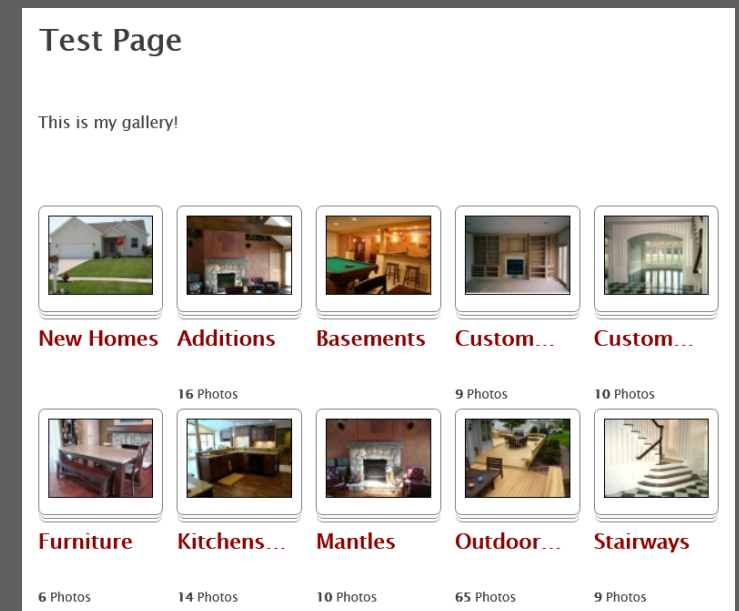
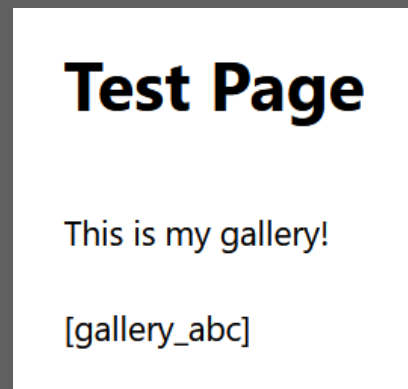


{[shortcodes]}

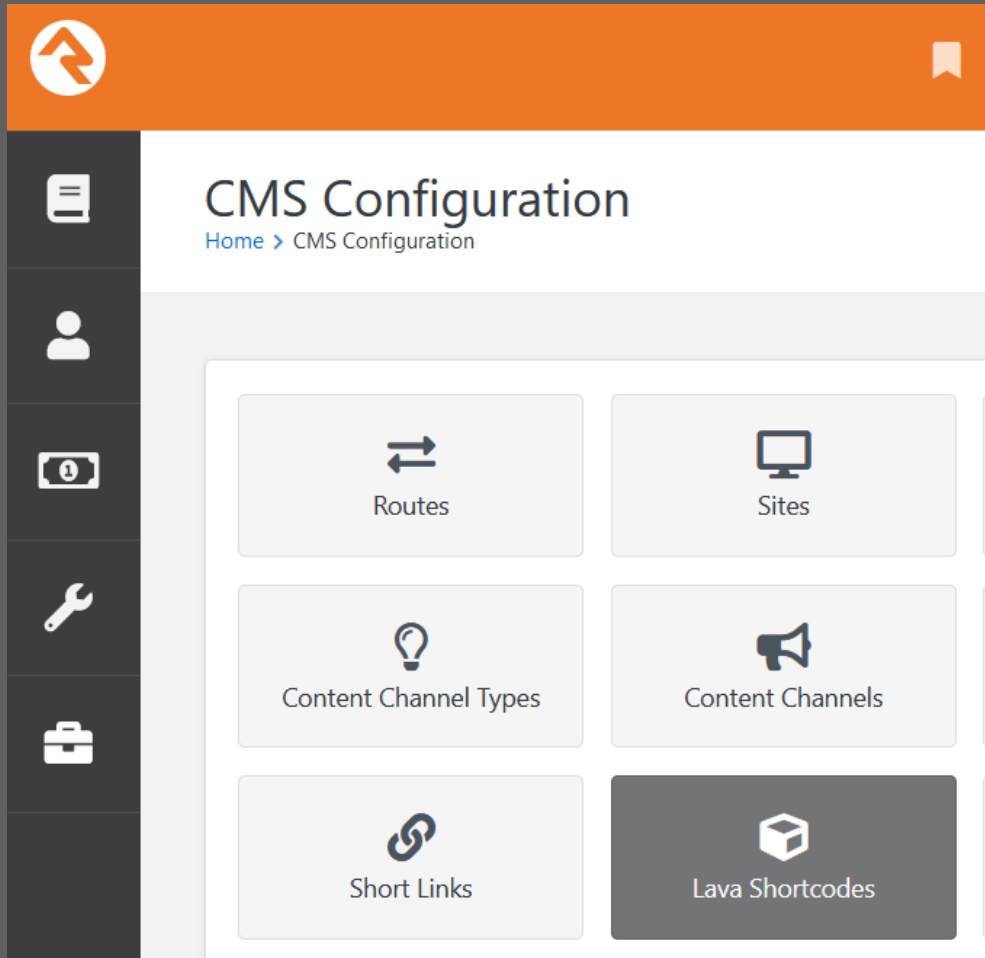


WHAT ARE SHORTCODES?

In WordPress for example, shortcodes are used by many plugins to insert something into a page like a news ticker or photo gallery. Code like `[gallery_abc]` might be used on the page for a plugin to pull in a photo gallery (with the ID of “abc”) for the end user to view.



WHAT ARE SHORTCODES?



Rock handles shortcodes in a similar fashion where you can define the keyword to use, insert what text/code to call, and then insert the keyword on a page using `{[]}` as your brackets.

You can find existing system shortcodes, and add your own by going to: **Admin Tools → CMS Configuration → Lava Shortcodes**

WHAT ARE SHORTCODES?

In “short”... a shortcode is a way to display, insert, or call larger amounts of text or code to a page without having to type it all on that particular page. It's sort of like a global variable you can recall by simply inserting a keyword of your choosing. i.e. `{[define]}` could be used to insert this paragraph

...But there's more! You can also use parameters with shortcodes to cause them to function differently. A more useful example would be the built-in `{[youtube]}` shortcode which uses a video id parameter to know which video to load. i.e. `{[youtube id:'8kpHK4YlwY4']}`

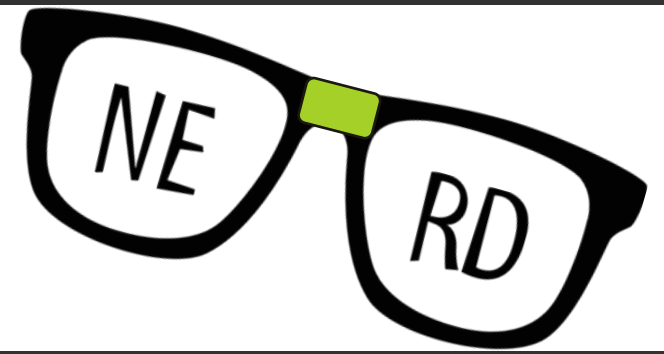
WHY USE LAVA SHORTCODES?

Less Duplicated Content

Speed of Updates

End User Experience

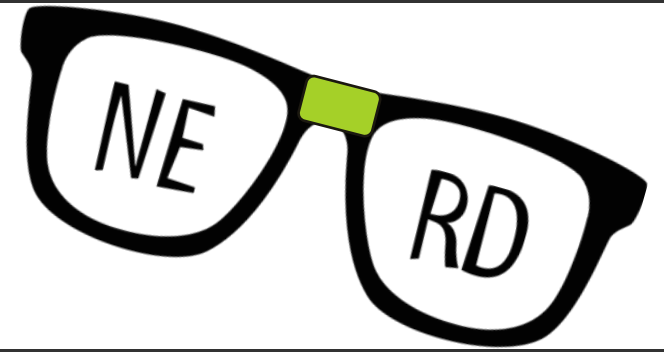
THE TECHNICAL



Disclaimer:

My examples here are not best practice. Some things here you probably should do with CSS Class styling, not shortcodes, but this is an easy way to illustrate what shortcodes do. We'll use LAVA based ones later, but for now, here is some CSS as a proof of concept.

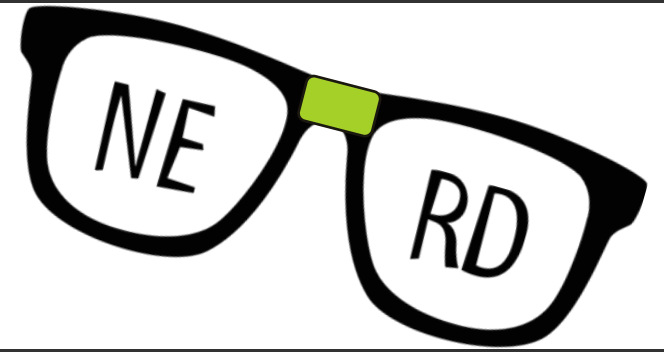
THE TECHNICAL



There are **2** types of shortcodes.

INLINE
&
BLOCK

THE TECHNICAL: **INLINE**

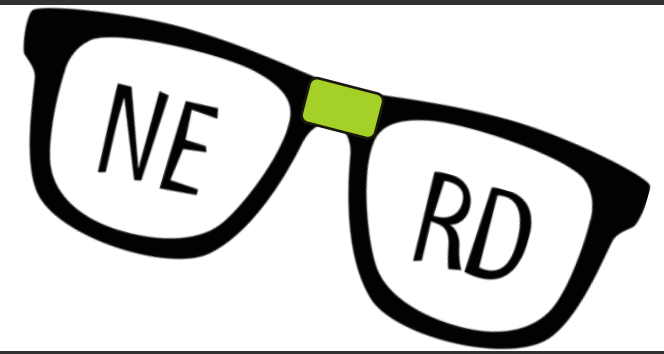


INLINE means everything is contained in a single line, such as:

```
{[ brucebanner ]}
```

Let's build a shortcode that replaces this with an image of Bruce Banner!

THE TECHNICAL: **INLINE**



Add Lava Shortcode

Name •


Active
☒

Tag Name ?

Tag Type ?
☒ Inline ☐ Block

Description
This shortcode will show a level headed scientist.

Documentation ?



Use `{[bruce]}` to show an image of Mark Ruffalo pulled off Wikipedia.

Shortcode Markup •

```
1 
```

Name: Bruce

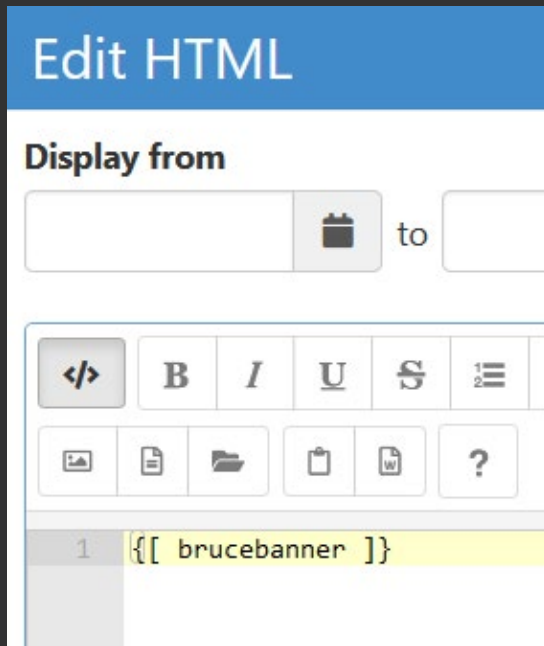
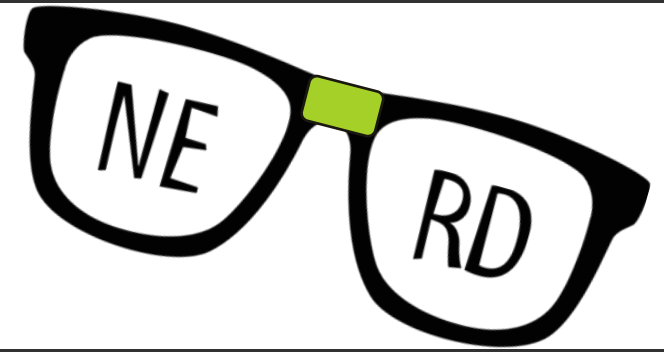
Tag Name: brucebanner

Description: This shortcode will show a level headed scientist

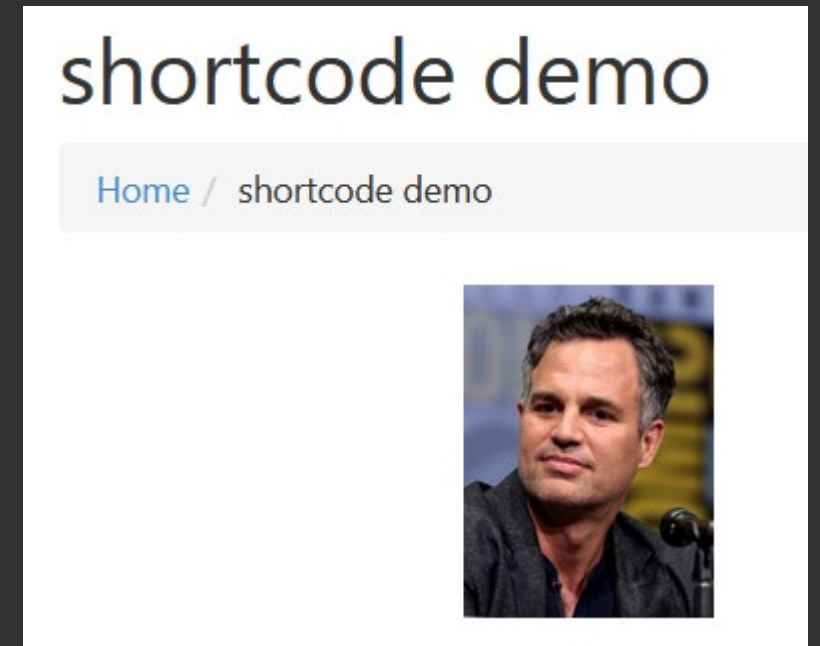
Documentation: Use `{[brucebanner]}` to show an image of Mark Ruffalo pulled off Wikipedia.

Shortcode Markup: ``

THE TECHNICAL: **INLINE**

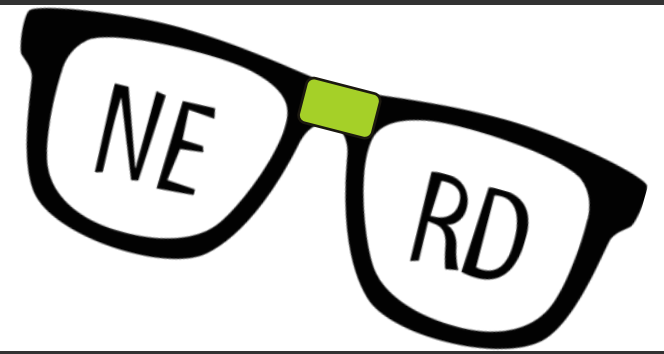


Now whenever anyone uses `<code>{{ brucebanner }}</code>` anywhere on the site, it is replaced with a 100px wide photo of Mark Ruffalo!



Do you see the implications? Instead of what could be hundreds of lines of code, a single keyword can be used on any page that needs whatever function your shortcode provides!

THE TECHNICAL: BLOCK



BLOCK means everything is wrapped in opening/closing tags, such as:

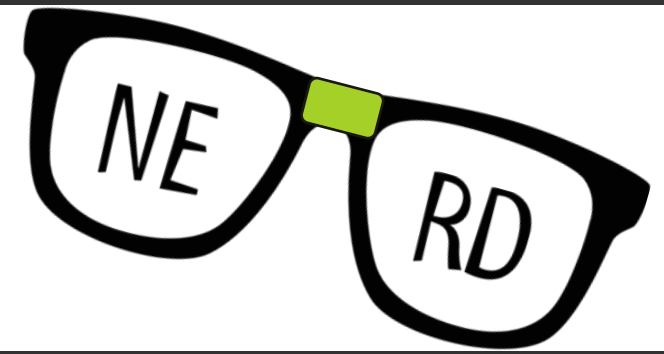
```
{[ hulk ]}
```

```
<p>I'm Angry. Hulk Smash!</p>
```

```
{[ endhulk ]}
```

Everything between the opening/closing tags (in green above) is what the shortcode will reference as `{{ blockContent }}` when we create it. So let's create a shortcode whose job is to... hulkify this...

THE TECHNICAL: BLOCK



Add Lava Shortcode

Name •
Incredible Hulkify

Active ☒

Tag Name i •
hulk

Tag Type i •
☐ Inline ☒ Block

Description
This shortcode will make text angry. You wouldn't like it when it's angry.

Documentation i

Use `{{ hulk }}` tags around text to make text green, larger, and in a Google Font called "Frijole".

Shortcode Markup •

```
1 <style>
2 @import url('https://fonts.googleapis.com/css2?family=Frijole&display=swap');
3 </style>
4 <span style="color: green; font-size: 3em; font-family: 'Frijole';">
5 {{ blockContent }}
6 </span>
```

Name:
Incredible Hulkify

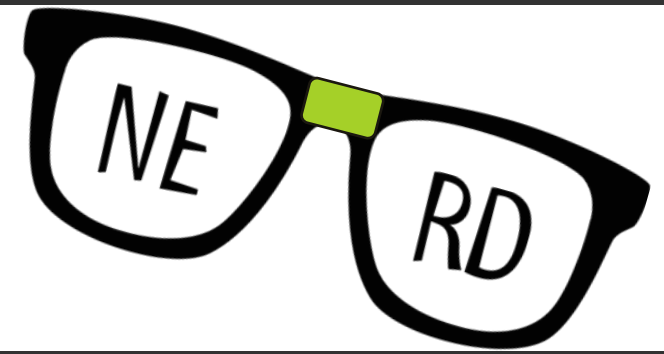
Tag Name:
hulk

Description:
This shortcode will make text angry. You wouldn't like it when it's angry.

Documentation:
Use `{{ hulk }}` tags around text to make text green, larger, and in a Google Font called "Frijole".

Shortcode Markup:
`<style>`
 `@import url('https://fonts.googleapis.com/css2?family=Frijole&display=swap');`
`</style>`
 ``
 `{{ blockContent }}`
 ``

THE TECHNICAL: BLOCK



Edit HTML

Display from to

to

```
1  {[ hulk ]}  
2  <p>I'm Angry. Hulk Smash!</p>  
3  {[ endhulk ]}
```

Now whenever anyone wraps text in `{[hulk]}` and `{[endhulk]}` tags anywhere on the site, it is completely hulkified!



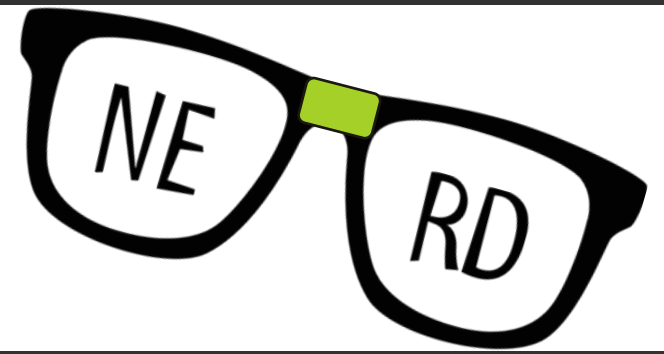
shortcode demo

[Home](#) / shortcode demo

I'M ANGRY.
HULK SMASH!

So we passed the code between our opening/closing tags back as `{{ blockContent }}` to our BLOCK shortcode...
Can we pass content back when using INLINE shortcodes?

THE TECHNICAL: BLOCK VS. INLINE

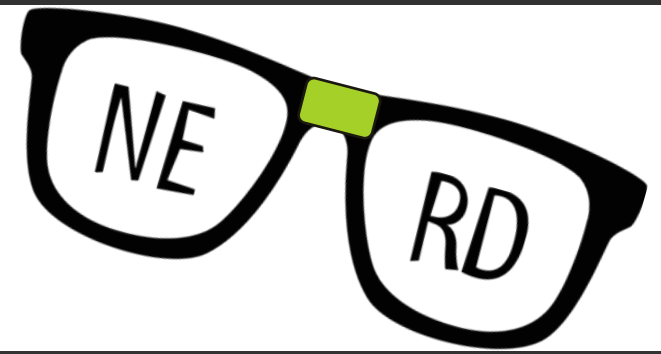


Similarly to how BLOCK shortcodes use `{{ blockContent }}` to pass back content to the shortcode, INLINE shortcodes can use `{{ content }}` to pass back. So you COULD accomplish our previous example something like:

```
{[ hulk content:<p>Hulk SMASH!</p>' ]}
```

Everything inside the content parameter (in green above) is what the INLINE shortcode would reference as `{{ content }}` when created.

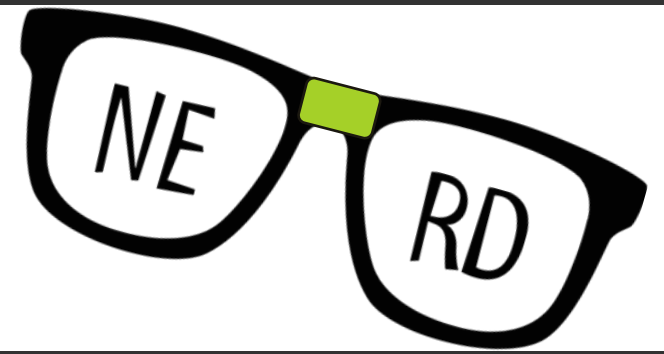
THE TECHNICAL: BLOCK VS. INLINE



So if this is possible... why have BLOCK at all? Why not just pass everything in INLINE shortcodes?

1. Because INLINE can't handle content with line breaks or single quotes in it. If you noticed, our BLOCK example said, "I'm Angry. Hulk Smash!" whereas our INLINE example just said, "Hulk Smash!" because the single quote in "I'm" would have broken it.
2. It also makes things way harder to read with complicated shortcodes that have a lot of content being passed, or with multiple parameters, as it would be a single long string. Don't do that to the next person looking at your code. You're better than that. Clean it up using BLOCK.

THE TECHNICAL: MULTIPLE PARAMETERS



You aren't limited to just `{{ blockContent }}` or `{{ content }}` being passed to the shortcode. In fact, multiple parameters can be used to alter the shortcode output. Let's look at the built-in YouTube shortcode. Here we see it requires an id parameter for the video id, but has optional parameters as well to customize the output:

- width
- showinfo
- controls
- autoplay

YouTube

Creates a responsive YouTube embed from just a simple video id.

Embedding a YouTube video is easy, right? Well what if you (see what we did there) the time it takes to get a video up o

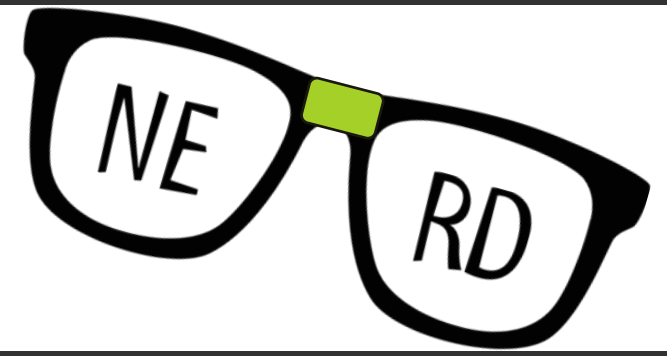
Basic Usage:

```
{[ youtube id:'8kpHK4YIwY4' ]}
```

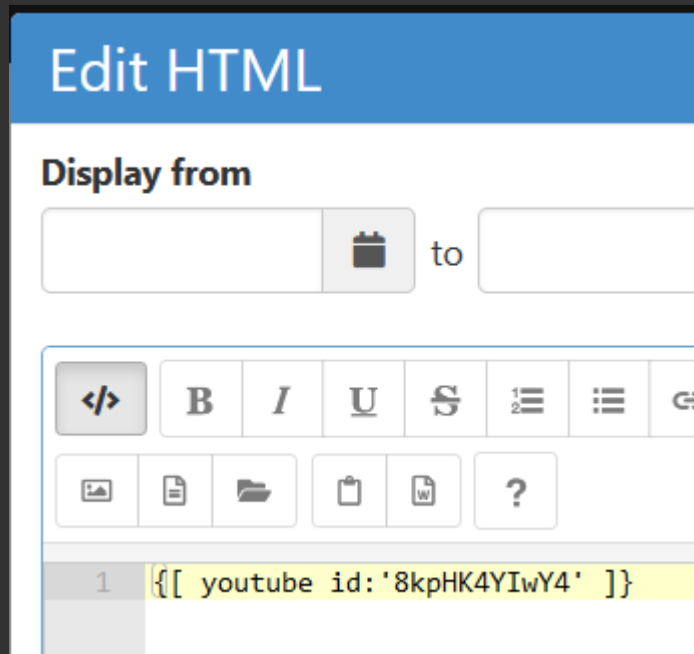
This will put the video with the id provide onto your page i

- id (required) – The YouTube id of the video.
- width (100%) – The width you would like the video to
- showinfo (false) – This determines if the name of the
- controls (true) – This determines if the standard YouT
- autoplay (false) – Should the video start playing once

THE TECHNICAL: MULTIPLE PARAMETERS



Using only the required **id** parameter gives us this output:

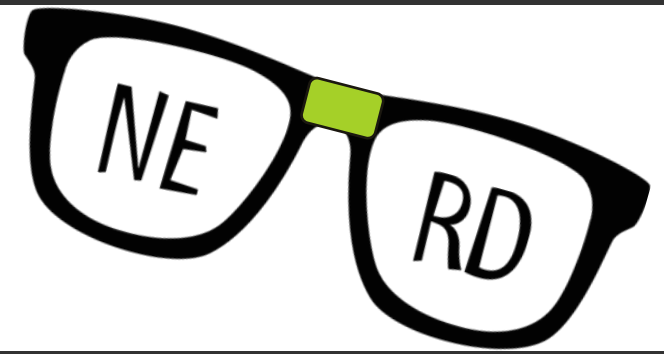


shortcode demo

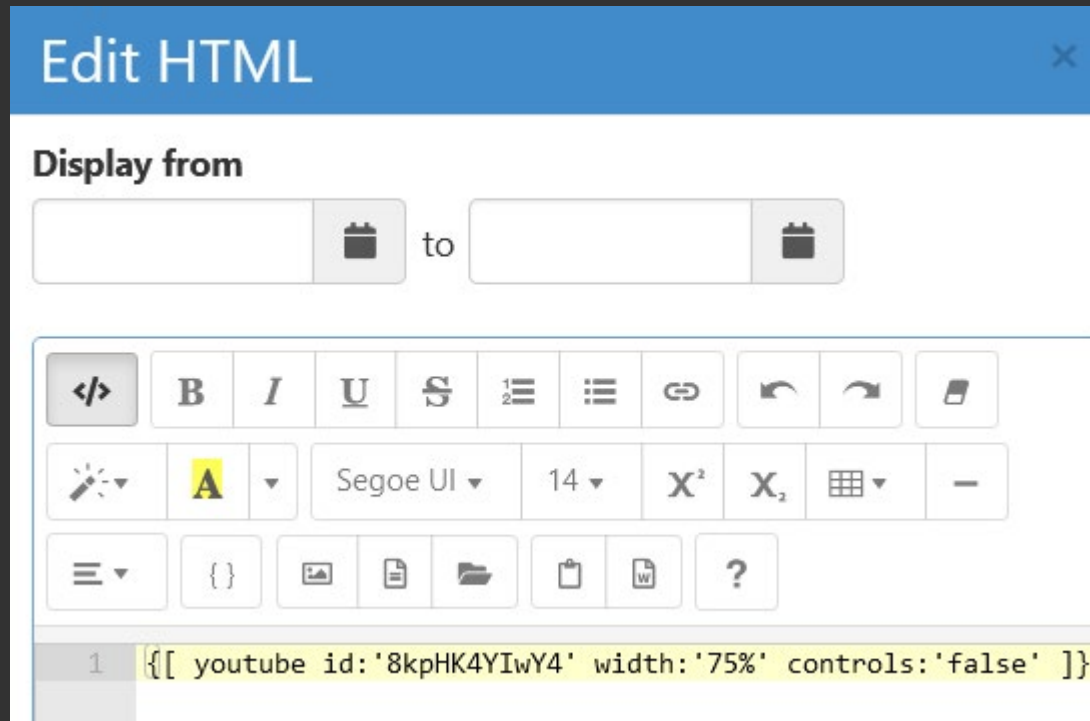
[Home](#) / shortcode demo



THE TECHNICAL: MULTIPLE PARAMETERS



Using the **width** and **controls** parameters we can customize the output to 75% width and hide the playback controls:



BEFORE

shortcode demo

[Home](#) / [shortcode demo](#)



AFTER

shortcode demo

[Home](#) / [shortcode demo](#)



“Enough with the nerd stuff... Let’s see what we came here for!”

USING LAVA SHORTCODES



...to Build a Volunteer
Onboarding Portal

LET'S GET PRACTICAL

After people are “ready” to get involved, how can they know if they are READY?

- Did they fully complete Growth Track?
- Do they need a background check?
- Did they pass an audition?
- Do they need training?
- Did they do an interview?
- Do they need to shadow others?
- Do they need to sign agreements? (behavior or confidentiality)
- Did they submit a writing sample?
- WHO DO THEY NEED TO TALK TO?

LET'S GET PRACTICAL



QR code redirects to <https://my.cedarcreek.tv/onboarding>

DREAMTEAM ONBOARDING

[Login](#) to view the requirements for each serving area and which you have or haven't met. You must fulfill these requirements before serving in that particular role.

Serving Areas

Choose an area to view the requirements:

[Arts \(Band / Music\)](#)

[Arts \(Lights, Video, Sound, Stage\)](#)

[Baptism Team](#)

[Care Team](#)

[Celebrate Recovery Team](#)

[Family Ministries Team](#)

[First Impressions Team](#)

[Group Leader](#)

[Growth Track Team](#)

[Living It Out Team](#)

[Online Campus Moderator](#)

[Prayer Team](#)

[Pre-Marriage Mentors](#)

Requirements

You are not logged in. Login at the top right to see the requirements for your account.

Have you met all the requirements for the area you want to join?

[Apply Here](#)

DREAMTEAM ONBOARDING

- 1) Choose a Serving Area on the left to view the requirements for that role.
- 2) You will see your status for each of the requirements on the right.
- 3) You must fulfill these requirements before serving in that particular role.

Serving Areas

Choose an area to view the requirements:

[Arts \(Band / Music\)](#)

[Arts \(Lights, Video, Sound, Stage\)](#)

[Baptism Team](#)

[Care Team](#)

[Celebrate Recovery Team](#)

[Family Ministries Team](#)

[First Impressions Team](#)

[Group Leader](#)

[Growth Track Team](#)

[Living It Out Team](#)

[Online Campus Moderator](#)

[Prayer Team](#)

[Pre-Marriage Mentors](#)

Requirements

Choose an area on the left to see if (and when) you've met the requirements for that area.

Have you met all the requirements for the area you want to join?

[Apply Here](#)

DREAMTEAM ONBOARDING

- 1) Choose a Serving Area on the left to view the requirements for that role.
- 2) You will see your status for each of the requirements on the right.
- 3) You must fulfill these requirements before serving in that particular role.

Serving Areas

Choose an area to view the requirements:

[Arts \(Band / Music\)](#)

[Arts \(Lights, Video, Sound, Stage\)](#)

[Baptism Team](#)

[Care Team](#)

[Celebrate Recovery Team](#)

[Family Ministries Team](#)

[First Impressions Team](#)

[Group Leader](#)

[Growth Track Team](#)

[Living It Out Team](#)

[Online Campus Moderator](#)

[Prayer Team](#)

[Pre-Marriage Mentors](#)

Requirements

Pre-Marriage Mentors

✔ Growth Track Week 1: completed 7/7/2018

✔ Growth Track Week 2: completed 5/12/2018

✔ Growth Track Week 3: completed 5/19/2018

✔ Growth Track Week 4: completed 5/26/2018

✘ Pre-Marriage Mentor Interview: not complete

Contact us at groups@CedarCreek.tv for NextSteps.

✔ Group Leader Agreement: completed 2/7/2022 (valid for 58 more days)

✔ Confidentiality Agreement: completed 8/20/2019

✔ SYMBIS Training: completed 7/17/2019

Have you met all the requirements for the area you want to join?

[Apply Here](#)

Serving Areas

Choose an area to view the requirements:

[Arts \(Band / Music\)](#)

[Arts \(Lights, Video, Sound, Stage\)](#)

[Baptism Team](#)

[Care Team](#)

[Celebrate Recovery Team](#)

[Family Ministries Team](#)

[First Impressions Team](#)

[Group Leader](#)

[Growth Track Team](#)

[Living It Out Team](#)

[Online Campus Moderator](#)

[Prayer Team](#)

[Pre-Marriage Mentors](#)

Requirements

Group Leader

- ✔ **Growth Track Week 1:** completed 7/7/2018
- ✔ **Growth Track Week 2:** completed 5/12/2018
- ✔ **Growth Track Week 3:** completed 5/19/2018
- ✔ **Growth Track Week 4:** completed 5/26/2018
- ✔ **Group Leader Interview:** completed 10/9/2019
- ✔ **Group Leader Training:** completed
- ✔ **Group Leader Agreement:** completed 2/7/2022 (valid for 58 more days)

Serving Areas

Choose an area to view the requirements:

[Arts \(Band / Music\)](#)

[Arts \(Lights, Video, Sound, Stage\)](#)

[Baptism Team](#)

[Care Team](#)

[Celebrate Recovery Team](#)

[Family Ministries Team](#)

[First Impressions Team](#)

[Group Leader](#)

[Growth Track Team](#)

[Living It Out Team](#)

[Online Campus Moderator](#)

[Prayer Team](#)

[Pre-Marriage Mentors](#)

Requirements

Online Campus Moderator

✔ **Growth Track Week 1:** completed 7/7/2018

✔ **Growth Track Week 2:** completed 5/12/2018

✔ **Growth Track Week 3:** completed 5/19/2018

✔ **Growth Track Week 4:** completed 5/26/2018

✘ **Social Media Background Check:** not complete

Contact us at BackgroundChecks@CedarCreek.tv to request a new background check.

✘ **DreamTeam Interview:** not complete

Contact us at dreamteam@CedarCreek.tv for NextSteps.

✘ **Chat Moderator Agreement:** not complete

You can complete your Chat Moderator Agreement [HERE](#)

Serving Areas

Choose an area to view the requirements:

[Arts \(Band / Music\)](#)

[Arts \(Lights, Video, Sound, Stage\)](#)

[Baptism Team](#)

[Care Team](#)

[Celebrate Recovery Team](#)

[Family Ministries Team](#)

[First Impressions Team](#)

[Group Leader](#)

[Growth Track Team](#)

[Living It Out Team](#)

[Online Campus Moderator](#)

[Prayer Team](#)

[Pre-Marriage Mentors](#)

Requirements

Group Leader

✔ Growth Track Week 1: completed 7/7/2018	}	{ [ServeReqGrowthTrack] }
✔ Growth Track Week 2: completed 5/12/2018		
✔ Growth Track Week 3: completed 5/19/2018		
✔ Growth Track Week 4: completed 5/26/2018		
✔ Group Leader Interview: completed 10/9/2019	}	{ [ServeReqGLInt] }
✔ Group Leader Training: completed	}	{ [ServeReqGLTraining] }
✔ Group Leader Agreement: completed 2/7/2022 (valid for 58 more days)	}	{ [ServeReqGLAgreement] }


Code view:

```
<div id="servereqs8">
  <h4>Group Leader</h4>
  { [ ServeReqGrowthTrack ] }
  { [ ServeReqGLInt ] }
  { [ ServeReqGLTraining ] }
  { [ ServeReqGLAgreement ] }
</div>
```

LET'S GET PRACTICAL – EXAMPLES

{[ServeReqArtsAudition]}

A Basic Person Attribute Requirement Check:

Name *	Active
<input type="text" value="ServeReq - Arts Audition"/>	<input checked="" type="checkbox"/>
Tag Name ⓘ *	Tag Type ⓘ *
<input type="text" value="ServeReqArtsAudition"/>	<input checked="" type="radio"/> Inline <input type="radio"/> Block
Description	
<input type="text" value="Displays if a person has completed an Arts Audition."/>	
Documentation ⓘ	
<div></div> <p>Use this lava: {[ServeReqArtsAudition]}</p> <p>It will return something like:</p> <p>Arts Audition: completed 4/11/2022</p> <p>or</p> <p>Arts Audition: not complete</p>	

LET'S GET PRACTICAL – EXAMPLES

{[ServeReqArtsAudition]}

A Basic Person Attribute Requirement Check:

Shortcode Markup

```
1 <p>
2 {% assign artsaud = CurrentPerson | Attribute:'AUDDate' %}
3 {% if artsaud != '' %}
4   <i class="far fa-check-circle" style="color:#64fc23;"></i> <b>Arts Audition:</b> <span style="color:#64fc23;">completed {{ artsaud }}</span>
5 {% else %}
6   <i class="far fa-times-circle" style="color:#c20000;"></i> <b>Arts Audition:</b> <span style="color:#c20000;">not complete</span><br />
7   <div class="alert alert-danger">Contact us at <a href="mailto:arts@cedarcreek.tv">Arts@CedarCreek.tv</a> to request an audition.</div>
8 {% endif %}
9 </p>
```

```
1 <p>
2 {% assign artsaud = CurrentPerson | Attribute:'AUDDate' %}
3 {% if artsaud != '' %}
4   <i class="far fa-check-circle" style="color:#64fc23;"></i> <b>Arts Audition:</b> <span style="color:#64fc23;">completed {{ artsaud }}</span>
5 {% else %}
6   <i class="far fa-times-circle" style="color:#c20000;"></i> <b>Arts Audition:</b> <span style="color:#c20000;">not complete</span><br />
7   <div class="alert alert-danger">Contact us at <a href="mailto:arts@cedarcreek.tv">Arts@CedarCreek.tv</a> to request an audition.</div>
8 {% endif %}
9 </p>
```

LET'S GET PRACTICAL – EXAMPLES

{[ServeReqBackgroundCheck]}

More Complex Use Case – Calculating Dates and Various If/Else Checks:

Name *	Active
<input type="text" value="ServeReq - BackgroundCheck"/>	<input checked="" type="checkbox"/>
Tag Name <i>i</i> *	Tag Type <i>i</i> *
<input type="text" value="ServeReqBackgroundCheck"/>	<input checked="" type="radio"/> Inline <input type="radio"/> Block
Description	
<input type="text" value="Displays whether or not a person has a completed background check on file within the past 365 days."/>	
Documentation <i>i</i>	
<div><div><div></></div><div>B</div><div>I</div><div>S</div><div>↻</div><div>A</div><div>✎</div><div>☰</div><div>☰</div><div>📄</div><div>☰</div><div>🖼️</div><div>📄</div><div>📁</div><div>📄</div><div>📄</div><div>↶</div><div>↷</div></div><div>Use this lava: {[ServeReqBackgroundCheck]} It will return something like: Background Check: completed 1/11/2019 (valid for 14 more days) or Background Check: expired on 11/21/2017 or Background Check: not completed or Background Check: Approved (Not required as a minor) or Background Check: Approved (Police Certificate on file)</div></div>	

LET'S GET PRACTICAL – EXAMPLES

{[ServeReqBackgroundCheck]}

More Complex Use Case – Calculating Dates and Various If/Else Checks:

```
1 <p>
2 {% assign backgroundcheck = CurrentPerson | Attribute:'BackgroundCheckDate' %}
3 {% assign policecert = CurrentPerson | Attribute:'69442_PoliceOfficerCertificateStatus' %}
4 {% if CurrentPerson.Age <= '17' %}
5   <i class="far fa-check-circle" style="color:#64fc23;"></i> <b>Background Check:</b> <span style="color:#64fc23;">Approved</span> (Not required for a minor)
6 {% elseif policecert == 'Approved' %}
7   <i class="far fa-check-circle" style="color:#64fc23;"></i> <b>Background Check:</b> <span style="color:#64fc23;">Approved</span> (Police Certificate on file)
8 {% else %}
9   {% if backgroundcheck != '' %}
10     {% assign backgroundcheckdate = backgroundcheck | AsDateTime %}
11     {% assign limitdate = 'Now' | DateAdd:-365 | AsDateTime %}
12     {% assign backgroundcheckexpireddate = backgroundcheckdate | DateAdd: 365 | AsDateTime %}
13     {% if backgroundcheckdate >= limitdate %}
14       <i class="far fa-check-circle" style="color:#64fc23;"></i> <b>Background Check:</b> <span style="color:#64fc23;">completed {{ backgroundcheck }}</span>
15       (valid for {{ backgroundcheckexpireddate | DaysUntil }} more days)<br />
16     {% else %}
17       <i class="far fa-times-circle" style="color:#c20000;"></i> <b>Background Check:</b> <span style="color:#c20000;">expired on {{ backgroundcheckexpireddate }}
18       </span><br />
19       <div class="alert alert-danger">Contact the Director of Students at your Campus</div>
20     {% endif %}
21   {% else %}
22     <i class="far fa-times-circle" style="color:#c20000;"></i> <b>Background Check:</b> <span style="color:#c20000;">not completed</span><br />
23     <div class="alert alert-danger">Contact us at <a href="mailto:backgroundchecks@cedarcreek.tv">BackgroundChecks@CedarCreek.tv</a> to request a new
24     background check.</div>
25   {% endif %}
26 {% endif %}
27 </p>
```


LET'S GET PRACTICAL - TIPS

Set a standard naming convention to group codes together. I used “ServeReq – ” as a standard starting point.

ServeReq - Celebrate Recovery Covenant

[○ Show Details](#)

Displays if a person has completed a CR Covenant or not.

ServeReq - Celebrate Recovery Interview

[○ Show Details](#)

Displays if a person has completed a CR Interview or not.

ServeReq - Chat Moderator Agreement

[○ Show Details](#)

Displays if a person has completed their Chat Moderator Agreement or not and next steps.

ServeReq - Child Safety Training

[○ Show Details](#)

Displays whether or not a person has completed Child Safety Training.

ServeReq - Confidentiality Agreement

[○ Show Details](#)

Displays whether or not a person has a Confidentiality Agreement on file.

ServeReq - ConsentForm

[○ Show Details](#)

Displays whether or not a person has a Consent Form on file.

LET'S GET PRACTICAL - TIPS

Be sure to fill in helpful
Description and Documentation
fields to show people how to use
your shortcodes.

ServeReq - Celebrate Recovery Covenant

☒ Show Details

Displays if a person has completed a CR Covenant or not.

Use this lava: `{{ ServeReqCRCovenant }}`

It will return something like:

CR Leader Covenant: completed 4/11/2022

or

CR Leader Covenant: not complete



ServeReq - Celebrate Recovery Interview

☐ Show Details

Displays if a person has completed a CR Interview or not.

ServeReq - Chat Moderator Agreement

☐ Show Details

Displays if a person has completed their Chat Moderator Agreement or not and next steps.

LET'S GET PRACTICAL - TIPS

The bottom of the shortcode creation page has the option to grant additional rights. Only grant was is needed; especially if your shortcode allows user input. Hackers insert SQL commands and such in search fields and if not properly sanitized, that input can compromise a site. Basic Person Attribute lookups won't need any permissions here.

Enabled Lava Commands

- | | | |
|---|---|---|
| <input type="checkbox"/> All | <input type="checkbox"/> Execute | <input type="checkbox"/> Search |
| <input type="checkbox"/> Cache | <input type="checkbox"/> InteractionContent | <input type="checkbox"/> Sql |
| <input type="checkbox"/> CalendarEvents | <input type="checkbox"/> ChannellItemWrite | <input type="checkbox"/> WebRequest |
| <input type="checkbox"/> EventScheduledIns
tance | <input type="checkbox"/> InteractionWrite | <input type="checkbox"/> WorkflowActivate |
| | <input type="checkbox"/> RockEntity | |

LET'S GET PRACTICAL



If you'd like to dig deeper into what Lava Shortcodes can do, there is a developer resource called "The Long & Short on Shortcodes" available at

<https://community.rockrms.com/developer>

Ben Murphy has a great recipe as well called "A Love Poem to Lava Shortcodes" that is worth checking out:

<https://community.rockrms.com/recipes/249>

LET'S GET PRACTICAL

If you'd like to reference how we built this Volunteer Onboarding Portal, check out the recipe entitled *"Build a Volunteer Onboarding Portal using Shortcodes"* for copy/paste code and step-by-step instructions.

<https://community.rockrms.com/recipes/301>

...or just go to <https://community.rockrms.com> and search for "shortcodes" for that and more recipes using shortcodes.

Any Questions?