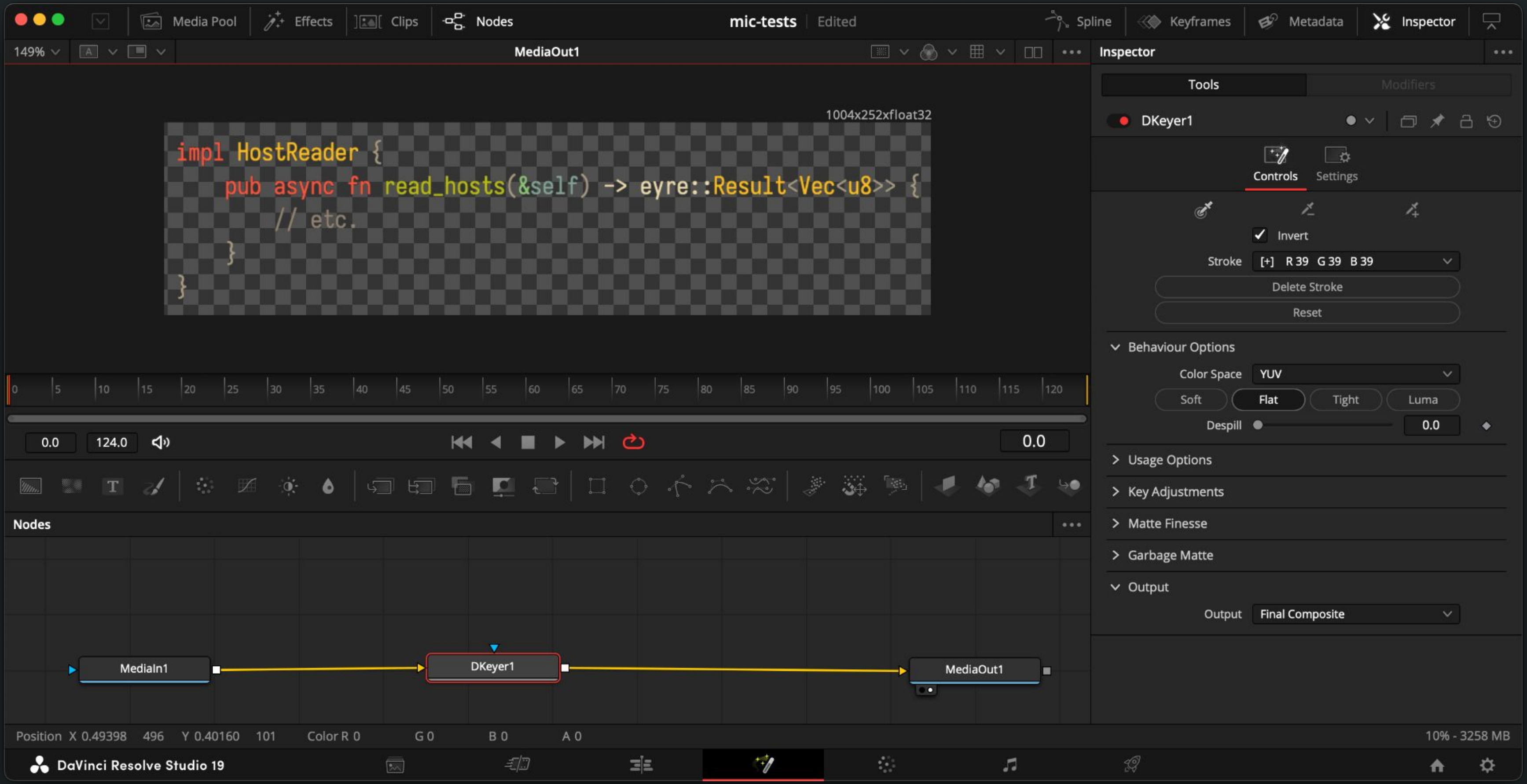
Blackmagic Fusion is weird

or: Migrating from AppleScript to Lua

Previously, on SDR

self-directed research



1004x252xfloat32

```
impl HostReader {
    pub async fn read_hosts(&self) -> eyre::Result<Vec<u8>> {
        // etc.
    }
}
```

☑ Invert

Stroke [+] R 39 G 39 B 39

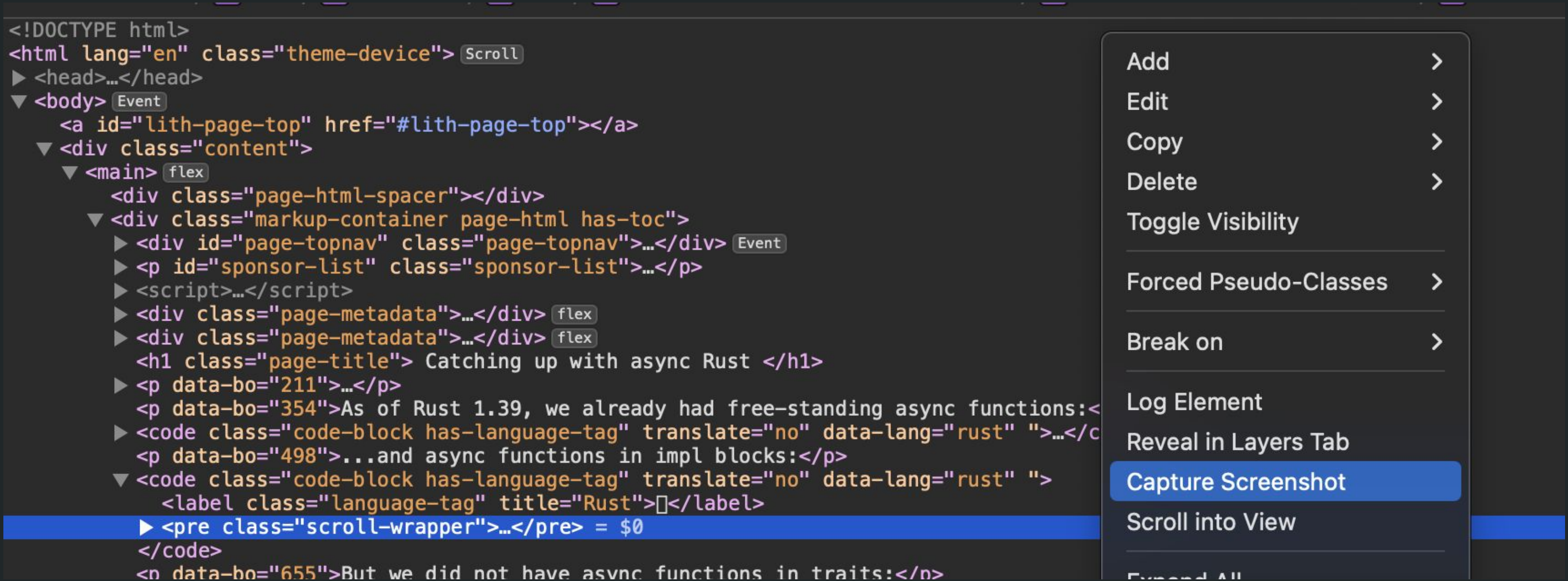
Color Space YUV

Despill 0.0

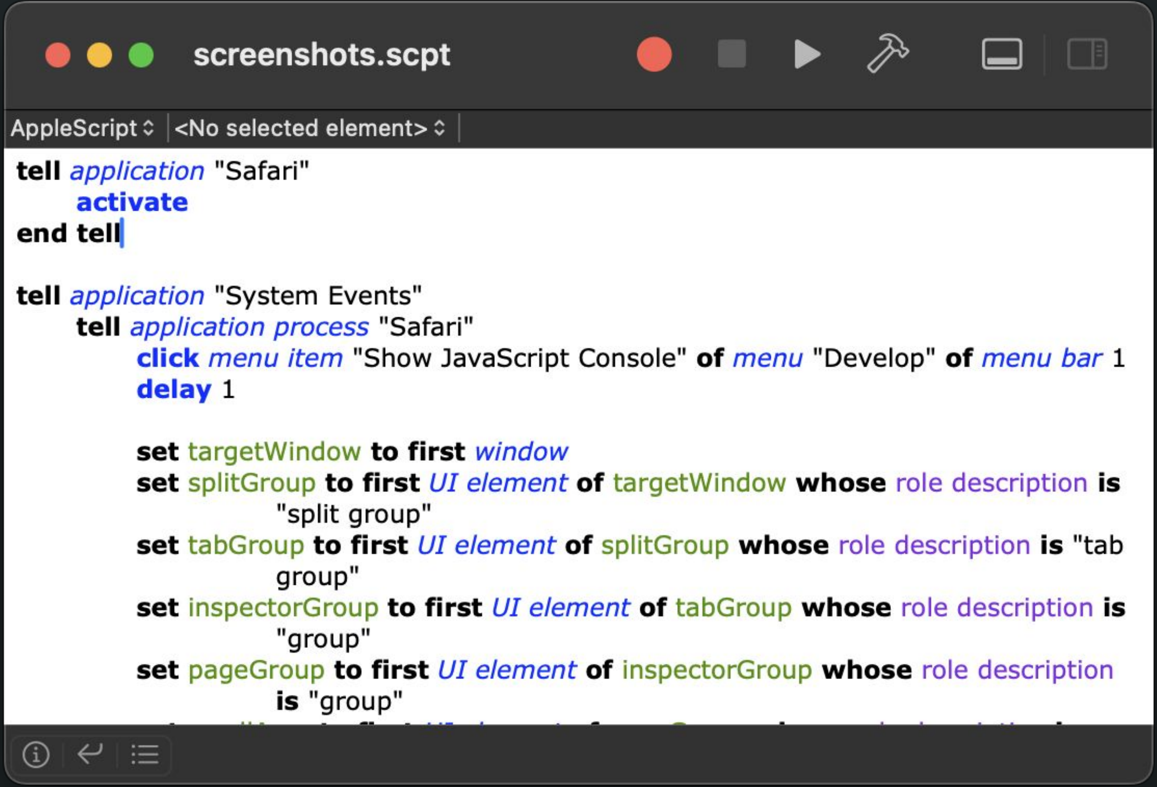
Output Final Composite

Position X 0.49398 496 Y 0.40160 101 Color R 0 G 0 B 0 A 0

10% - 3258 MB



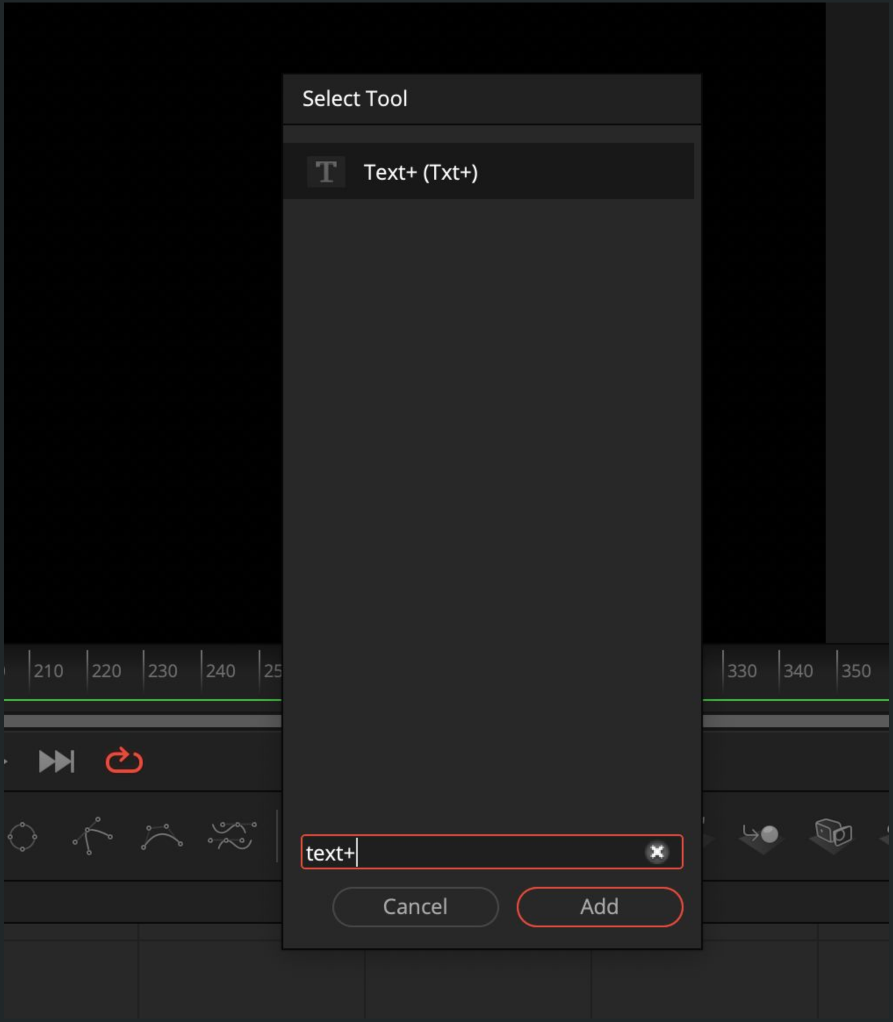
```
<!DOCTYPE html>
<html lang="en" class="theme-device"> Scroll
 ▶ <head>…</head>
 ▼ <body> Event
     <a id="lith-page-top" href="#lith-page-top"></a>
   ▼ <div class="content">
     ▼ <main> flex
         <div class="page-html-spacer"></div>
       ▼ <div class="markup-container page-html has-toc">
         ▶ <div id="page-topnav" class="page-topnav">…</div> Event
         ▶ <p id="sponsor-list" class="sponsor-list">…</p>
         ▶ <script>…</script>
         ▶ <div class="page-metadata">…</div> flex
         ▶ <div class="page-metadata">…</div> flex
           <h1 class="page-title"> Catching up with async Rust </h1>
         ▶ <p data-bo="211">…</p>
           <p data-bo="354">As of Rust 1.39, we already had free-standing async functions:<
         ▶ <code class="code-block has-language-tag" translate="no" data-lang="rust" ">…</c
           <p data-bo="498">...and async functions in impl blocks:</p>
         ▼ <code class="code-block has-language-tag" translate="no" data-lang="rust" ">
             <label class="language-tag" title="Rust">▯</label>
           ▶ <pre class="scroll-wrapper">…</pre> = $0
           </code>
           <p data-bo="655">But we did not have async functions in traits:</p>
```

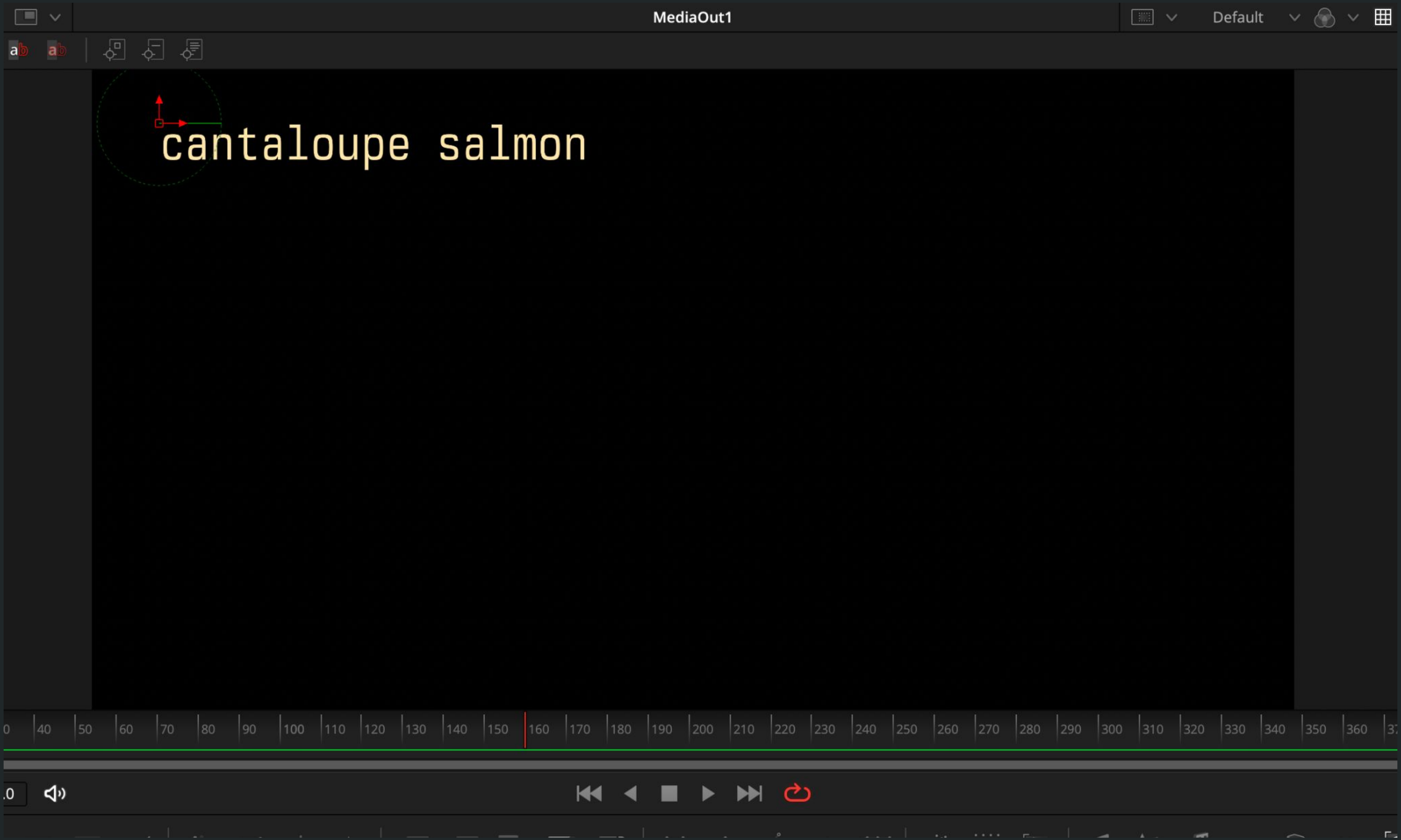


```
tell application "Safari"
	activate
end tell

tell application "System Events"
	tell application process "Safari"
		click menu item "Show JavaScript Console" of menu "Develop" of menu bar 1
		delay 1
		
		set targetWindow to first window
		set splitGroup to first UI element of targetWindow whose role description is "split group"
		set tabGroup to first UI element of splitGroup whose role description is "tab group"
		set inspectorGroup to first UI element of tabGroup whose role description is "group"
		set pageGroup to first UI element of inspectorGroup whose role description is "group"
```

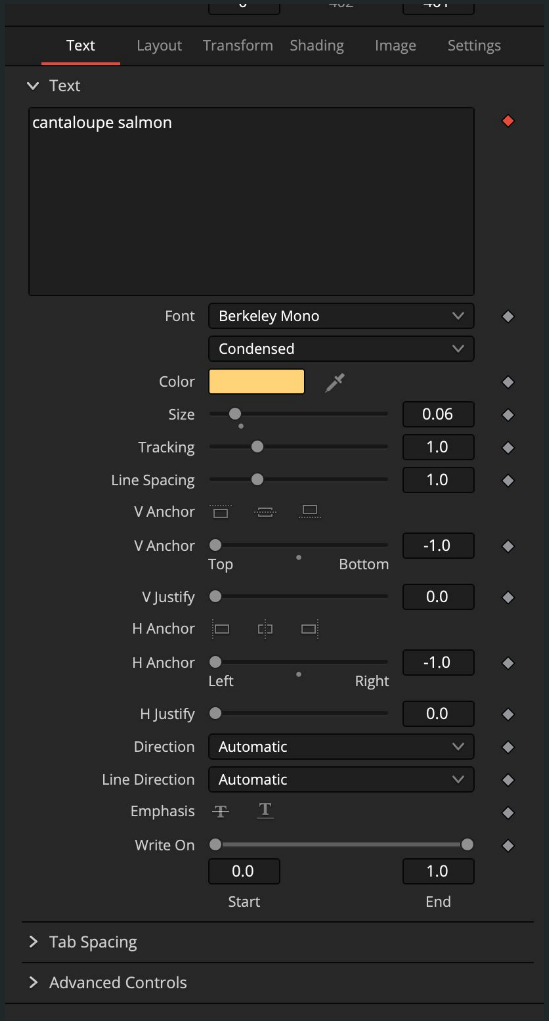
Fusion can render text!

self-directed research





cantaloupe salmon



Text | Layout | Transform | Shading | Image | Settings

Text

cantaloupe salmon

Font: Berkeley Mono

Condensed

Color

Size: 0.06

Tracking: 1.0

Line Spacing: 1.0

V Anchor

V Anchor: -1.0 (Top – Bottom)

V Justify: 0.0

H Anchor

H Anchor: -1.0 (Left – Right)

H Justify: 0.0

Direction: Automatic

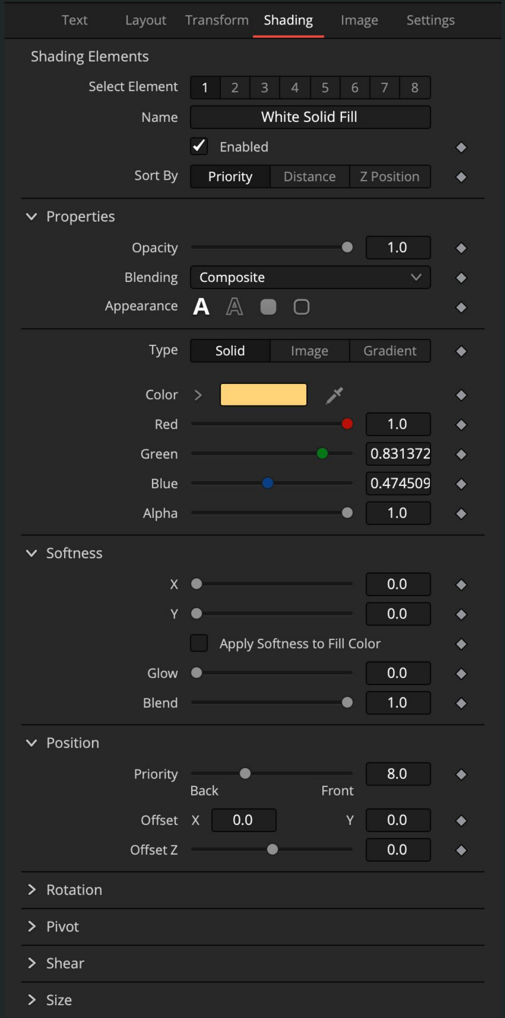
Line Direction: Automatic

Emphasis

Write On: Start 0.0, End 1.0

Tab Spacing

Advanced Controls



Text | Layout | Transform | Shading | Image | Settings

Shading Elements

Select Element: 1 2 3 4 5 6 7 8

Name: White Solid Fill

☑ Enabled

Sort By: Priority | Distance | Z Position

Properties

Opacity: 1.0

Blending: Composite

Appearance

Type: Solid | Image | Gradient

Color

Red: 1.0

Green: 0.831372

Blue: 0.474509

Alpha: 1.0

Softness

X: 0.0

Y: 0.0

□ Apply Softness to Fill Color

Glow: 0.0

Blend: 1.0

Position

Priority: 8.0 (Back – Front)

Offset X: 0.0 Y: 0.0

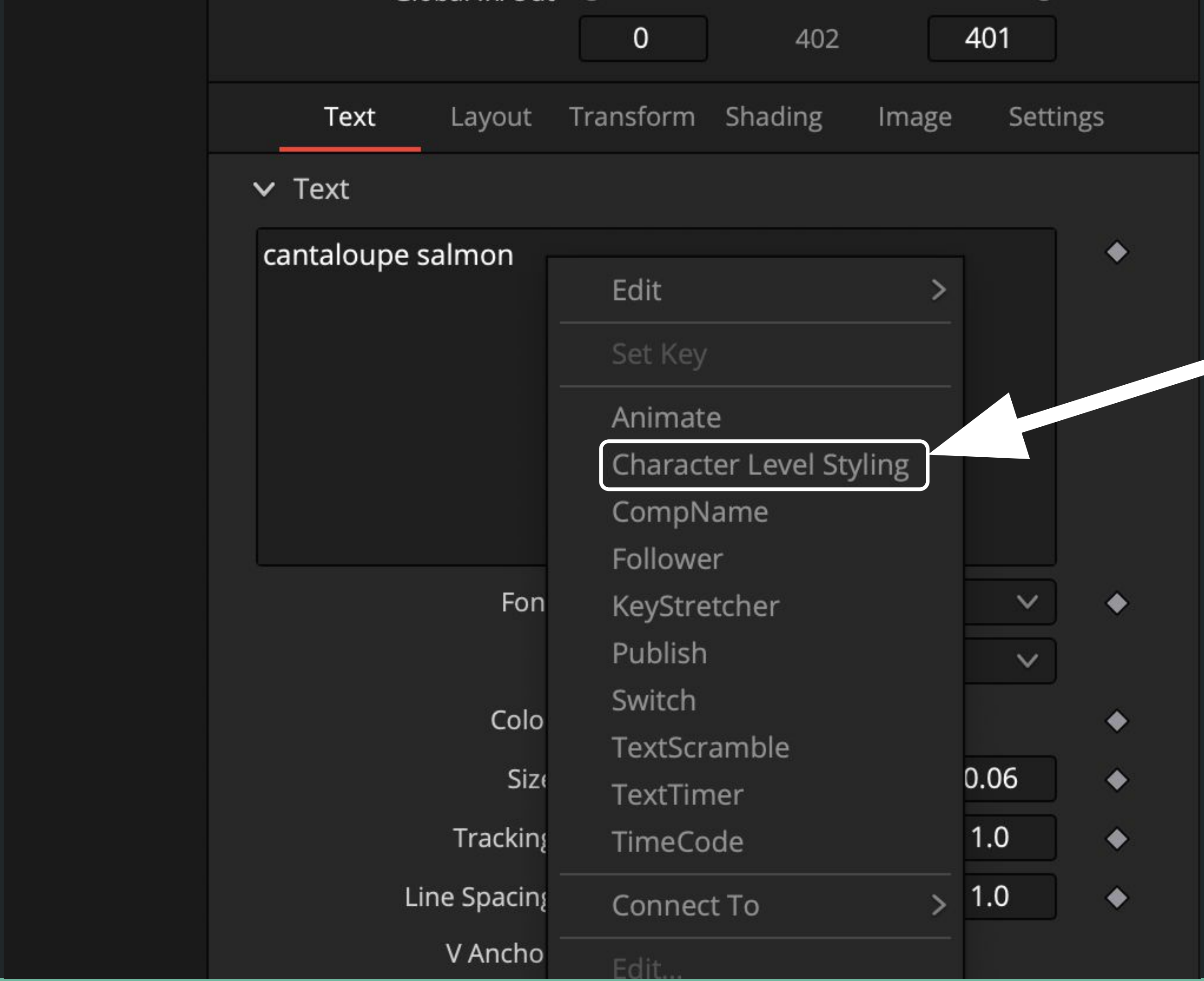
Offset Z: 0.0

Rotation

Pivot

Shear

Size

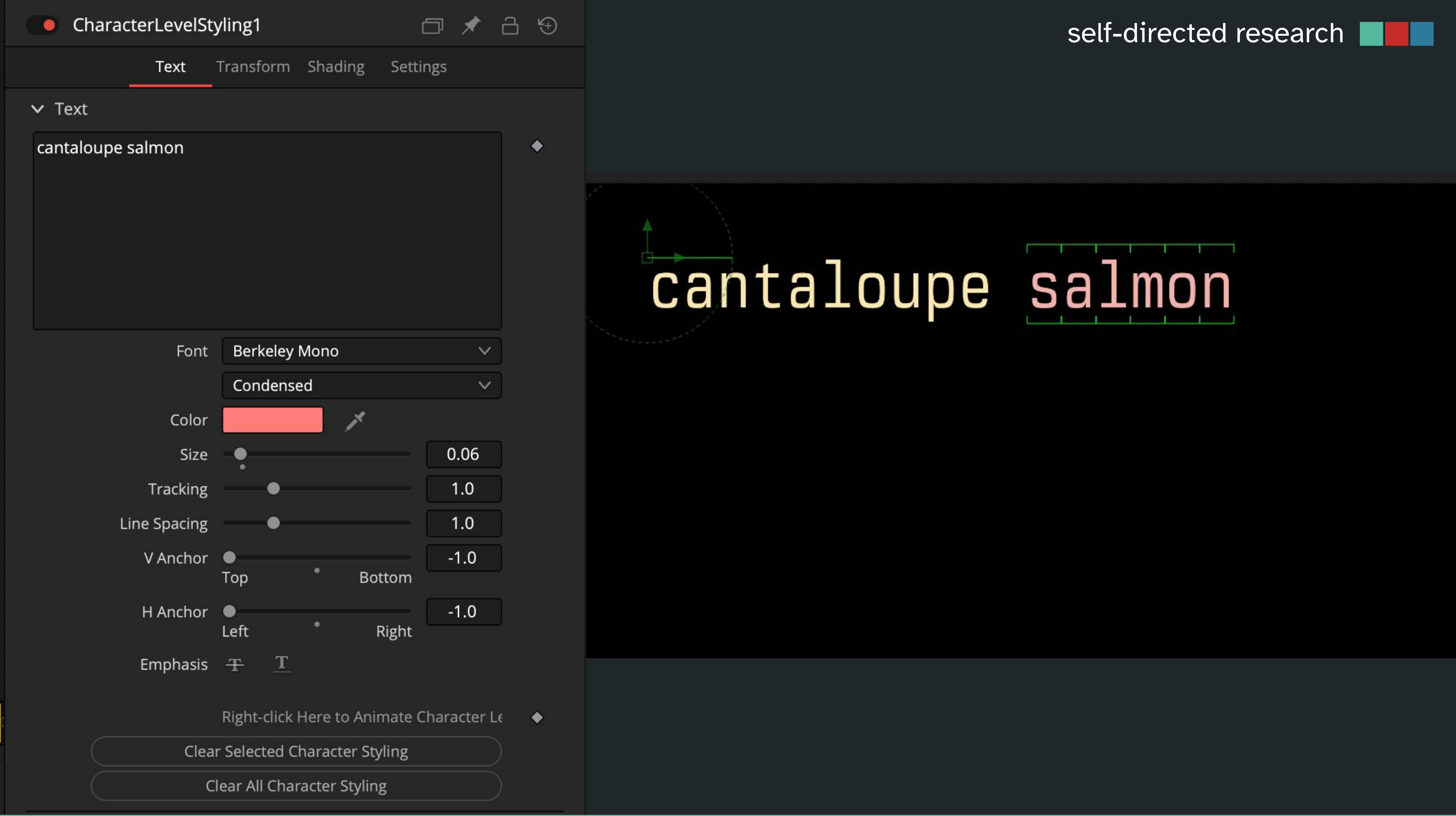


0 402 401

Text Layout Transform Shading Image Settings

Text

cantaloupe salmon



CharacterLevelStyling1

Text
Transform
Shading
Settings

Text

cantaloupe salmon

Font: Berkeley Mono

Condensed

Color

Size: 0.06

Tracking: 1.0

Line Spacing: 1.0

V Anchor: -1.0 (Top – Bottom)

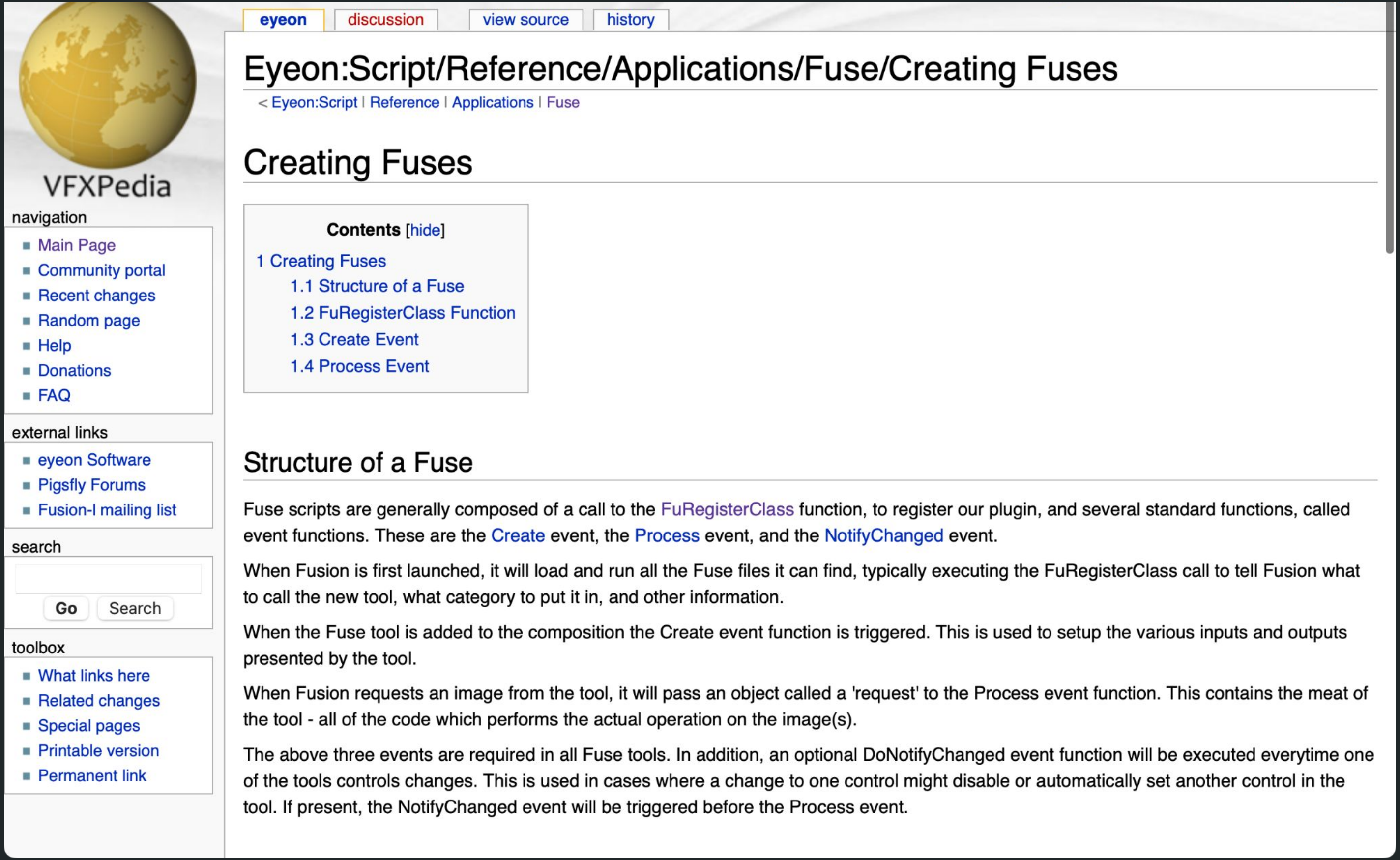
H Anchor: -1.0 (Left – Right)

Emphasis

Right-click Here to Animate Character Le
Clear Selected Character Styling
Clear All Character Styling
cantaloupe salmon

Can we automate *that*?

Fuses



Eyeon:Script/Reference/Applications/Fuse/Creating Fuses

< Eyeon:Script | Reference | Applications | Fuse

Creating Fuses

Contents [hide]

1 Creating Fuses
1.1 Structure of a Fuse
1.2 FuRegisterClass Function
1.3 Create Event
1.4 Process Event

Structure of a Fuse

Fuse scripts are generally composed of a call to the FuRegisterClass function, to register our plugin, and several standard functions, called event functions. These are the Create event, the Process event, and the NotifyChanged event.

When Fusion is first launched, it will load and run all the Fuse files it can find, typically executing the FuRegisterClass call to tell Fusion what to call the new tool, what category to put it in, and other information.

When the Fuse tool is added to the composition the Create event function is triggered. This is used to setup the various inputs and outputs presented by the tool.

When Fusion requests an image from the tool, it will pass an object called a 'request' to the Process event function. This contains the meat of the tool - all of the code which performs the actual operation on the image(s).

The above three events are required in all Fuse tools. In addition, an optional DoNotifyChanged event function will be executed everytime one of the tools controls changes. This is used in cases where a change to one control might disable or automatically set another control in the tool. If present, the NotifyChanged event will be triggered before the Process event.

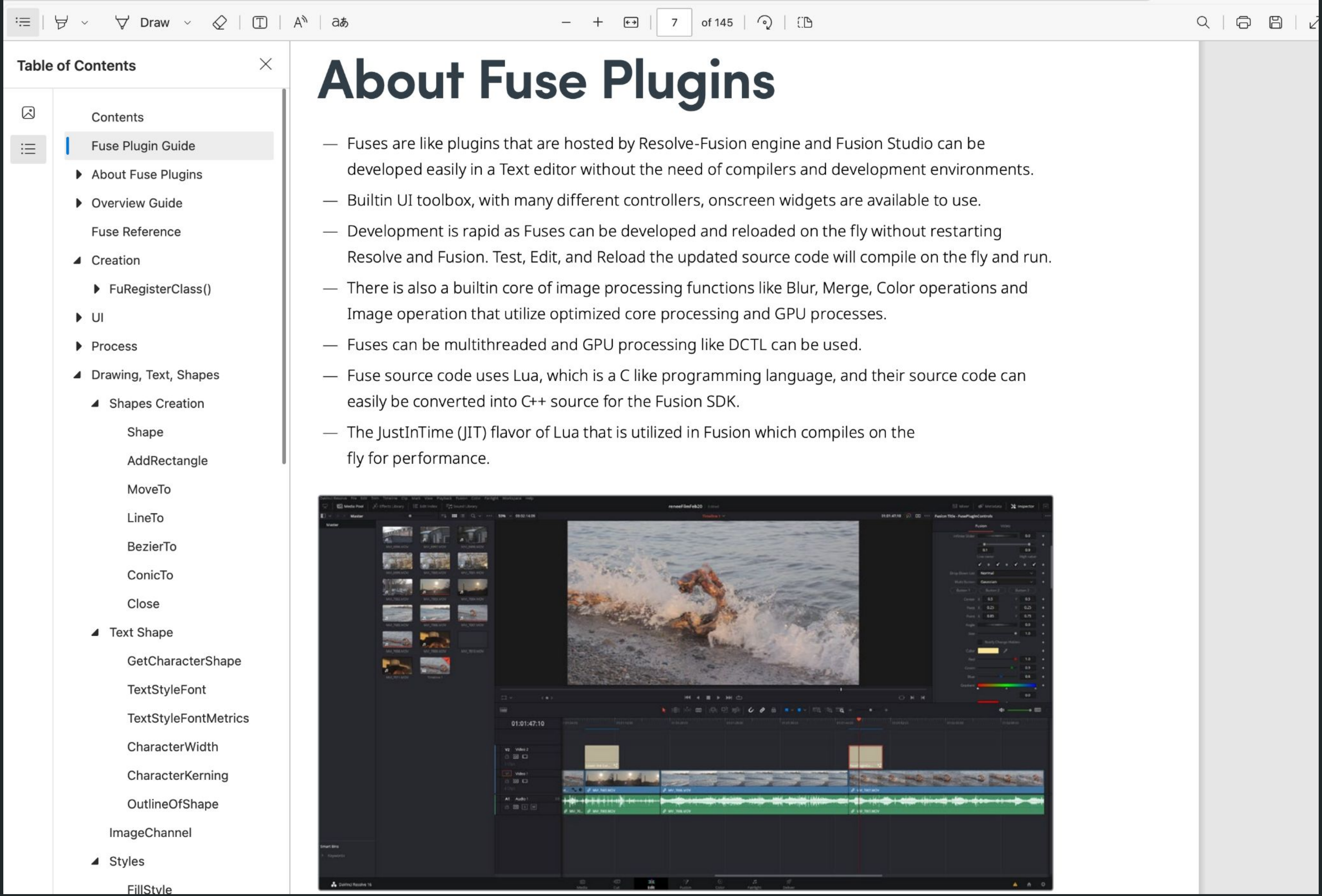
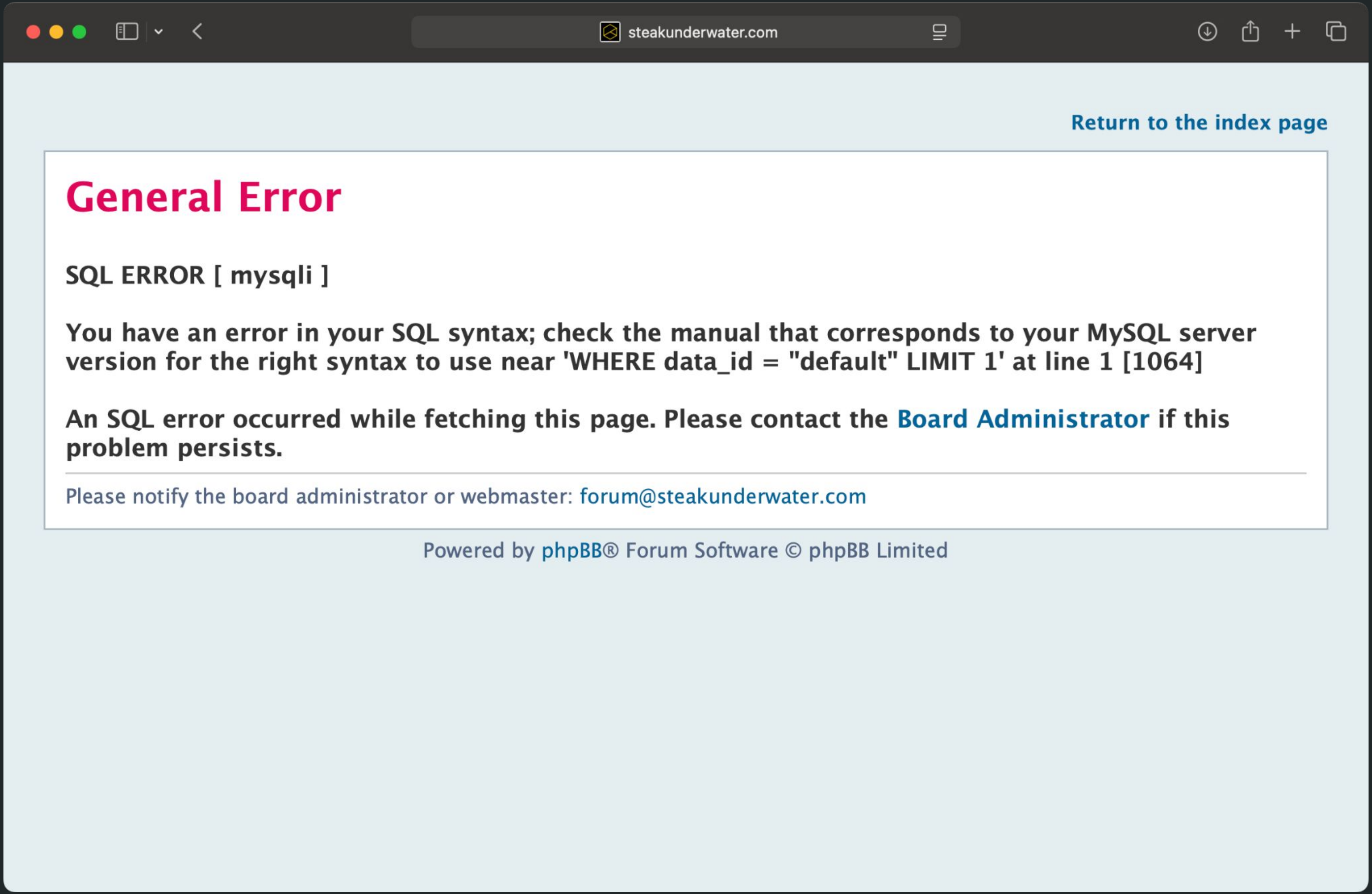


Table of Contents

Contents
Fuse Plugin Guide
About Fuse Plugins
Overview Guide
Fuse Reference
Creation
FuRegisterClass()
UI
Process
Drawing, Text, Shapes
Shapes Creation
Shape
AddRectangle
MoveTo
LineTo
BezierTo
ConicTo
Close
Text Shape
GetCharacterShape
TextStyleFont
TextStyleFontMetrics
CharacterWidth
CharacterKerning
OutlineOfShape
ImageChannel
Styles
FillStyle

About Fuse Plugins

- Fuses are like plugins that are hosted by Resolve-Fusion engine and Fusion Studio can be developed easily in a Text editor without the need of compilers and development environments.
- Builtin UI toolbox, with many different controllers, onscreen widgets are available to use.
- Development is rapid as Fuses can be developed and reloaded on the fly without restarting Resolve and Fusion. Test, Edit, and Reload the updated source code will compile on the fly and run.
- There is also a builtin core of image processing functions like Blur, Merge, Color operations and Image operation that utilize optimized core processing and GPU processes.
- Fuses can be multithreaded and GPU processing like DCTL can be used.
- Fuse source code uses Lua, which is a C like programming language, and their source code can easily be converted into C++ source for the Fusion SDK.
- The JustInTime (JIT) flavor of Lua that is utilized in Fusion which compiles on the fly for performance.



Return to the index page

General Error

SQL ERROR [mysqli]

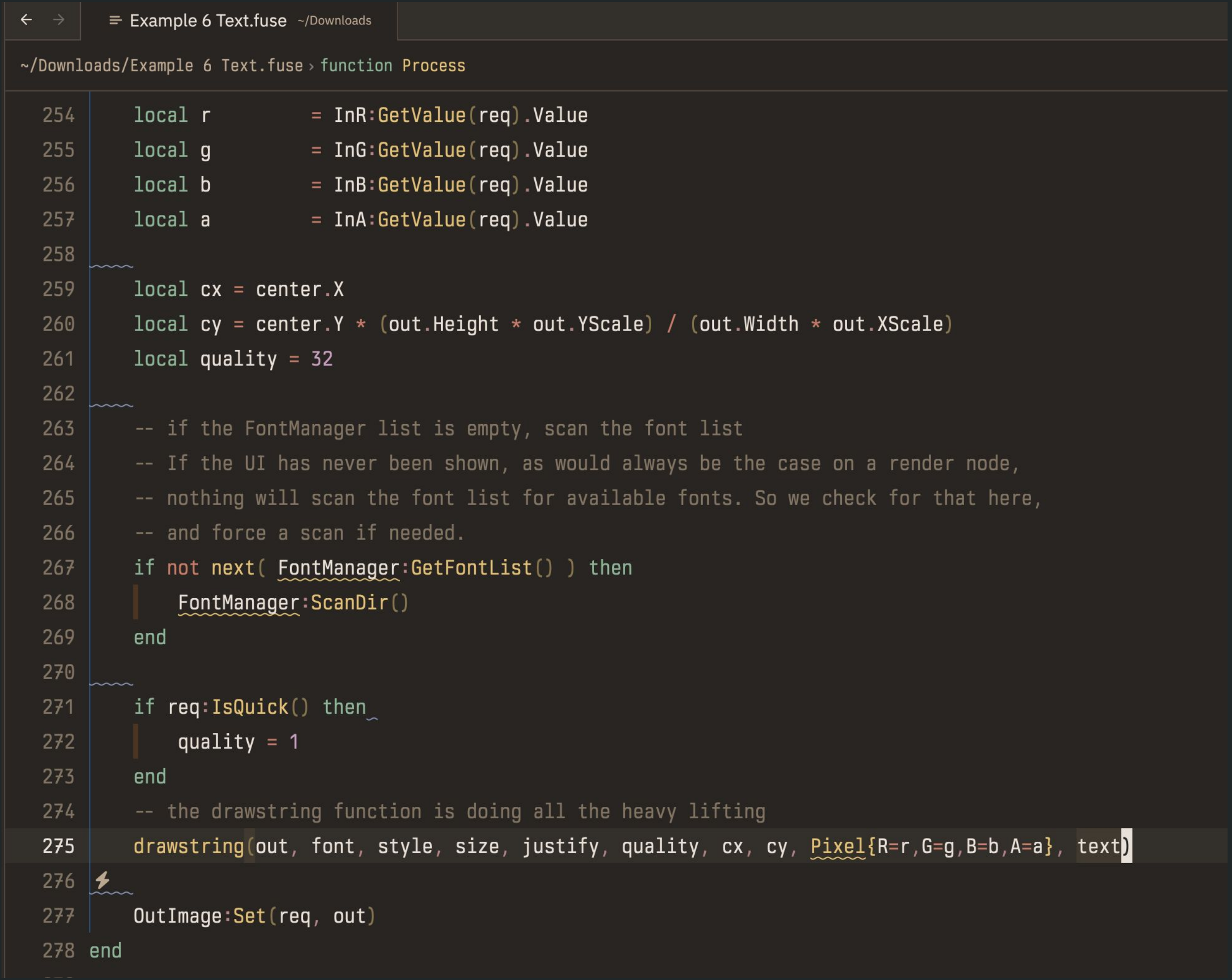
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'WHERE data_id = "default" LIMIT 1' at line 1 [1064]

An SQL error occurred while fetching this page. Please contact the Board Administrator if this problem persists.

Please notify the board administrator or webmaster: forum@steakunderwater.com

Powered by phpBB® Forum Software © phpBB Limited

self-directed research

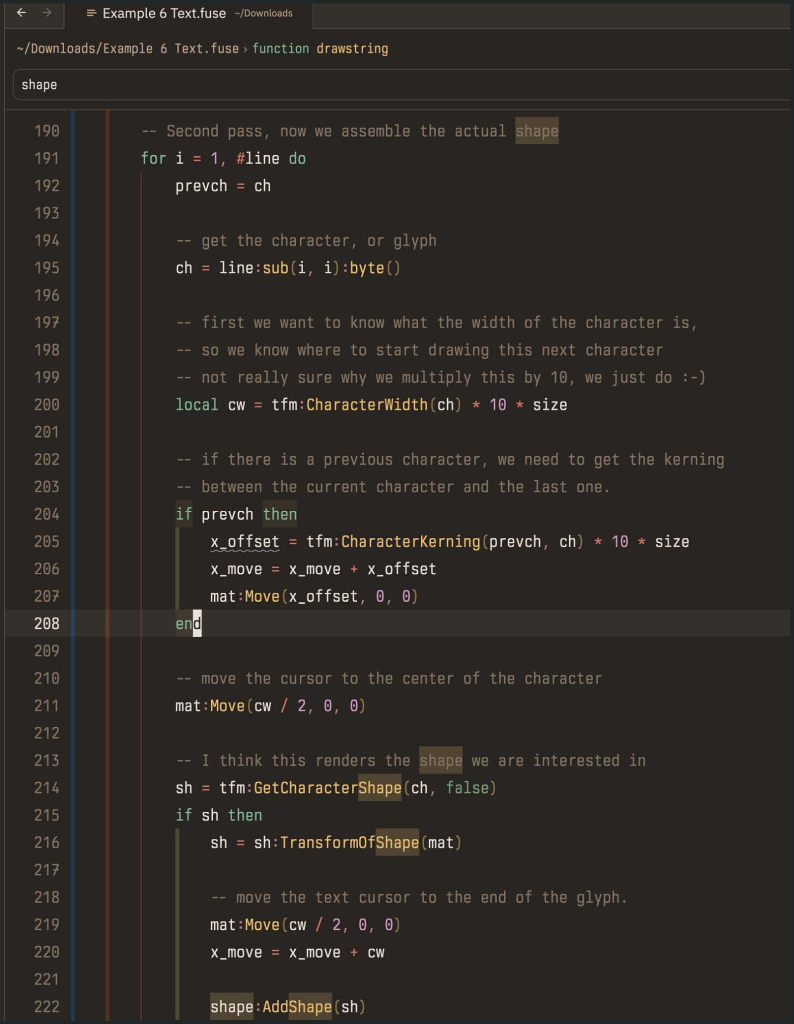


Example 6 Text.fuse ~/Downloads

~/Downloads/Example 6 Text.fuse › function Process

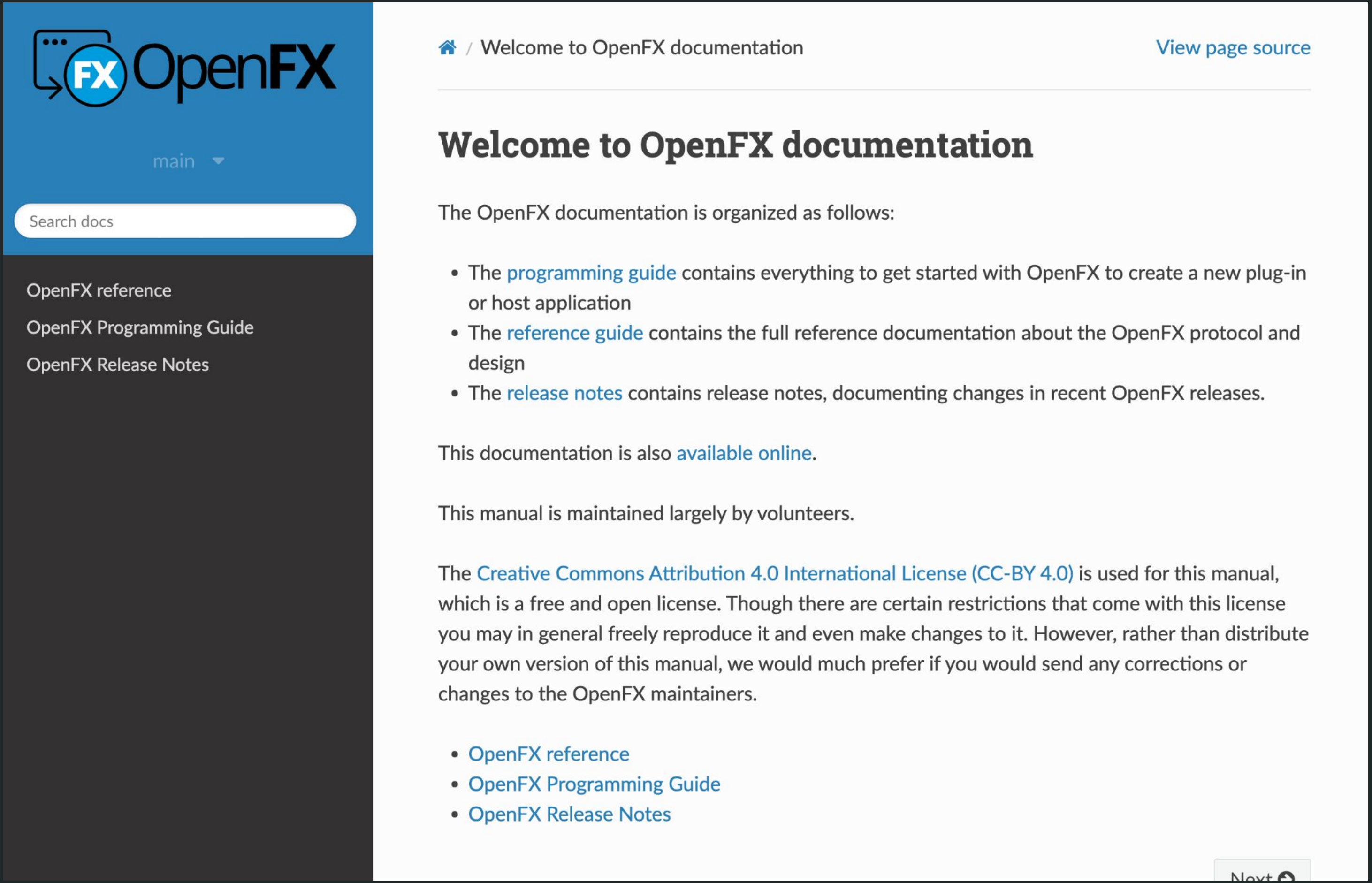
```
    local r         = InR:GetValue(req).Value
    local g         = InG:GetValue(req).Value
    local b         = InB:GetValue(req).Value
    local a         = InA:GetValue(req).Value

    local cx = center.X
    local cy = center.Y * (out.Height * out.YScale) / (out.Width * out.XScale)
    local quality = 32

    -- if the FontManager list is empty, scan the font list
    -- If the UI has never been shown, as would always be the case on a render node,
    -- nothing will scan the font list for available fonts. So we check for that here,
    -- and force a scan if needed.
    if not next( FontManager:GetFontList() ) then
        FontManager:ScanDir()
    end

    if req:IsQuick() then
        quality = 1
    end
    -- the drawstring function is doing all the heavy lifting
    drawstring(out, font, style, size, justify, quality, cx, cy, Pixel{R=r,G=g,B=b,A=a}, text)

    OutImage:Set(req, out)
end
```

self-directed research


Example 6 Text.fuse ~/Downloads
~/Downloads/Example 6 Text.fuse › function drawstring
shape

```
    -- Second pass, now we assemble the actual shape
    for i = 1, #line do
        prevch = ch

        -- get the character, or glyph
        ch = line:sub(i, i):byte()

        -- first we want to know what the width of the character is,
        -- so we know where to start drawing this next character
        -- not really sure why we multiply this by 10, we just do :-)
        local cw = tfm:CharacterWidth(ch) * 10 * size

        -- if there is a previous character, we need to get the kerning
        -- between the current character and the last one.
        if prevch then
            x_offset = tfm:CharacterKerning(prevch, ch) * 10 * size
            x_move = x_move + x_offset
            mat:Move(x_offset, 0, 0)
        end

        -- move the cursor to the center of the character
        mat:Move(cw / 2, 0, 0)

        -- I think this renders the shape we are interested in
        sh = tfm:GetCharacterShape(ch, false)
        if sh then
            sh = sh:TransformOfShape(mat)

            -- move the text cursor to the end of the glyph.
            mat:Move(cw / 2, 0, 0)
            x_move = x_move + cw

            shape:AddShape(sh)
```

OpenFX filters



Welcome to OpenFX documentation

The OpenFX documentation is organized as follows:

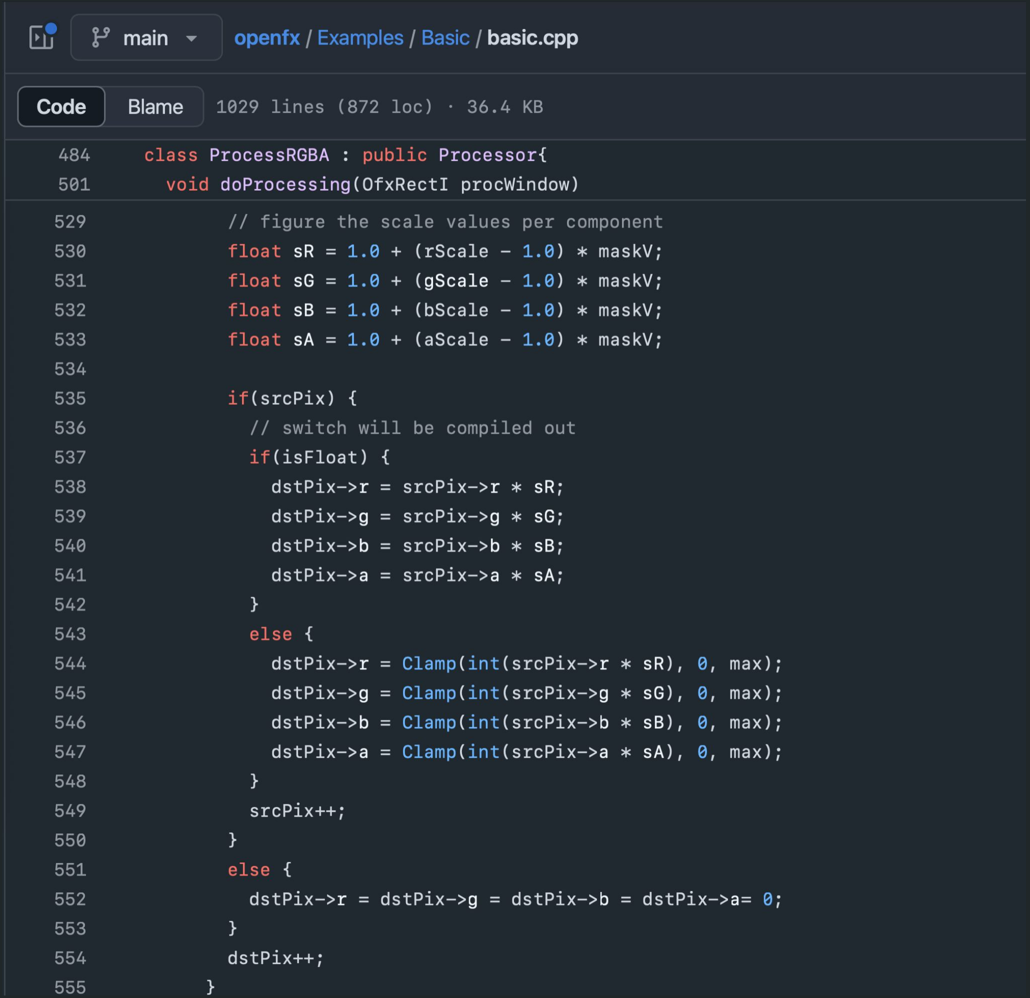
- The programming guide contains everything to get started with OpenFX to create a new plug-in or host application
- The reference guide contains the full reference documentation about the OpenFX protocol and design
- The release notes contains release notes, documenting changes in recent OpenFX releases.

This documentation is also available online.

This manual is maintained largely by volunteers.

The Creative Commons Attribution 4.0 International License (CC-BY 4.0) is used for this manual, which is a free and open license. Though there are certain restrictions that come with this license you may in general freely reproduce it and even make changes to it. However, rather than distribute your own version of this manual, we would much prefer if you would send any corrections or changes to the OpenFX maintainers.

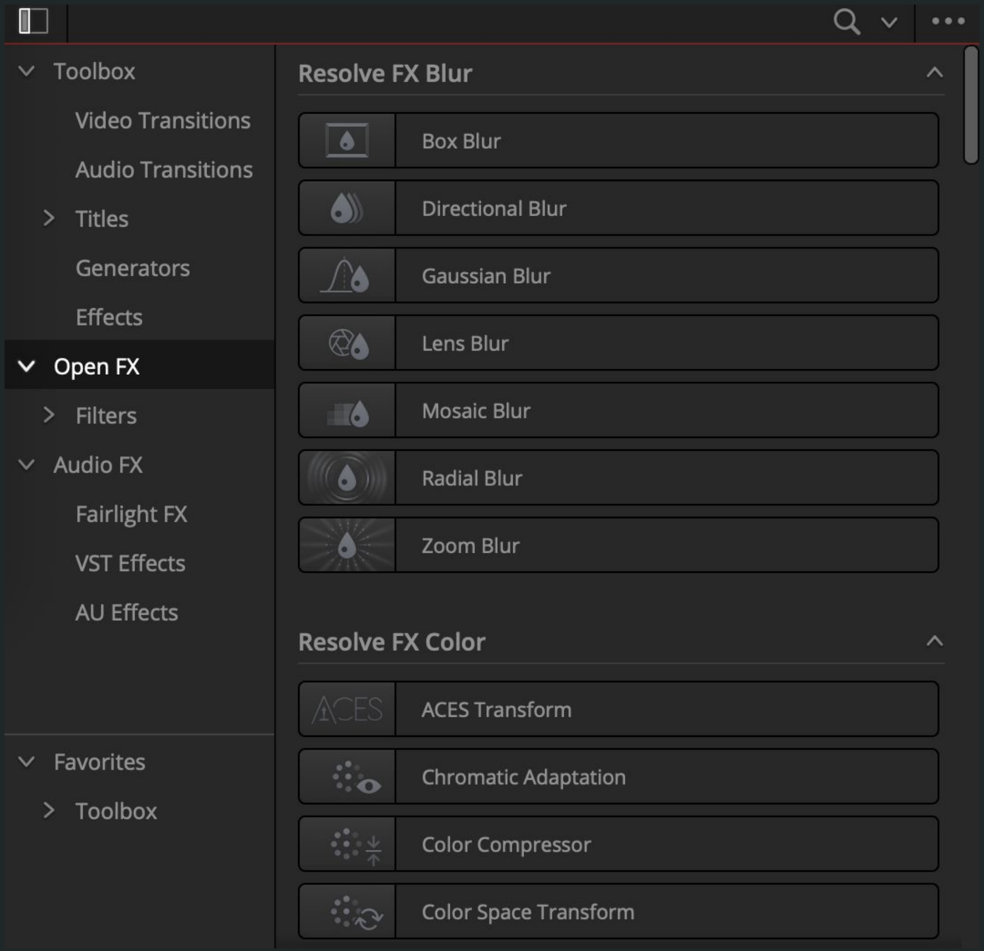
- OpenFX reference
- OpenFX Programming Guide
- OpenFX Release Notes



1029 lines (872 loc) · 36.4 KB

```
class ProcessRGBA : public Processor{
  void doProcessing(OfxRectI procWindow)
```

```
        // figure the scale values per component
        float sR = 1.0 + (rScale - 1.0) * maskV;
        float sG = 1.0 + (gScale - 1.0) * maskV;
        float sB = 1.0 + (bScale - 1.0) * maskV;
        float sA = 1.0 + (aScale - 1.0) * maskV;

        if(srcPix) {
          // switch will be compiled out
          if(isFloat) {
            dstPix->r = srcPix->r * sR;
            dstPix->g = srcPix->g * sG;
            dstPix->b = srcPix->b * sB;
            dstPix->a = srcPix->a * sA;
          }
          else {
            dstPix->r = Clamp(int(srcPix->r * sR), 0, max);
            dstPix->g = Clamp(int(srcPix->g * sG), 0, max);
            dstPix->b = Clamp(int(srcPix->b * sB), 0, max);
            dstPix->a = Clamp(int(srcPix->a * sA), 0, max);
          }
          srcPix++;
        }
        else {
          dstPix->r = dstPix->g = dstPix->b = dstPix->a= 0;
        }
        dstPix++;
      }
```



Toolbox

Video Transitions

Audio Transitions

Titles

Generators

Effects

Open FX

Filters

Audio FX

Fairlight FX

VST Effects

AU Effects

Favorites

Toolbox

Resolve FX Blur

Box Blur

Directional Blur

Gaussian Blur

Lens Blur

Mosaic Blur

Radial Blur

Zoom Blur

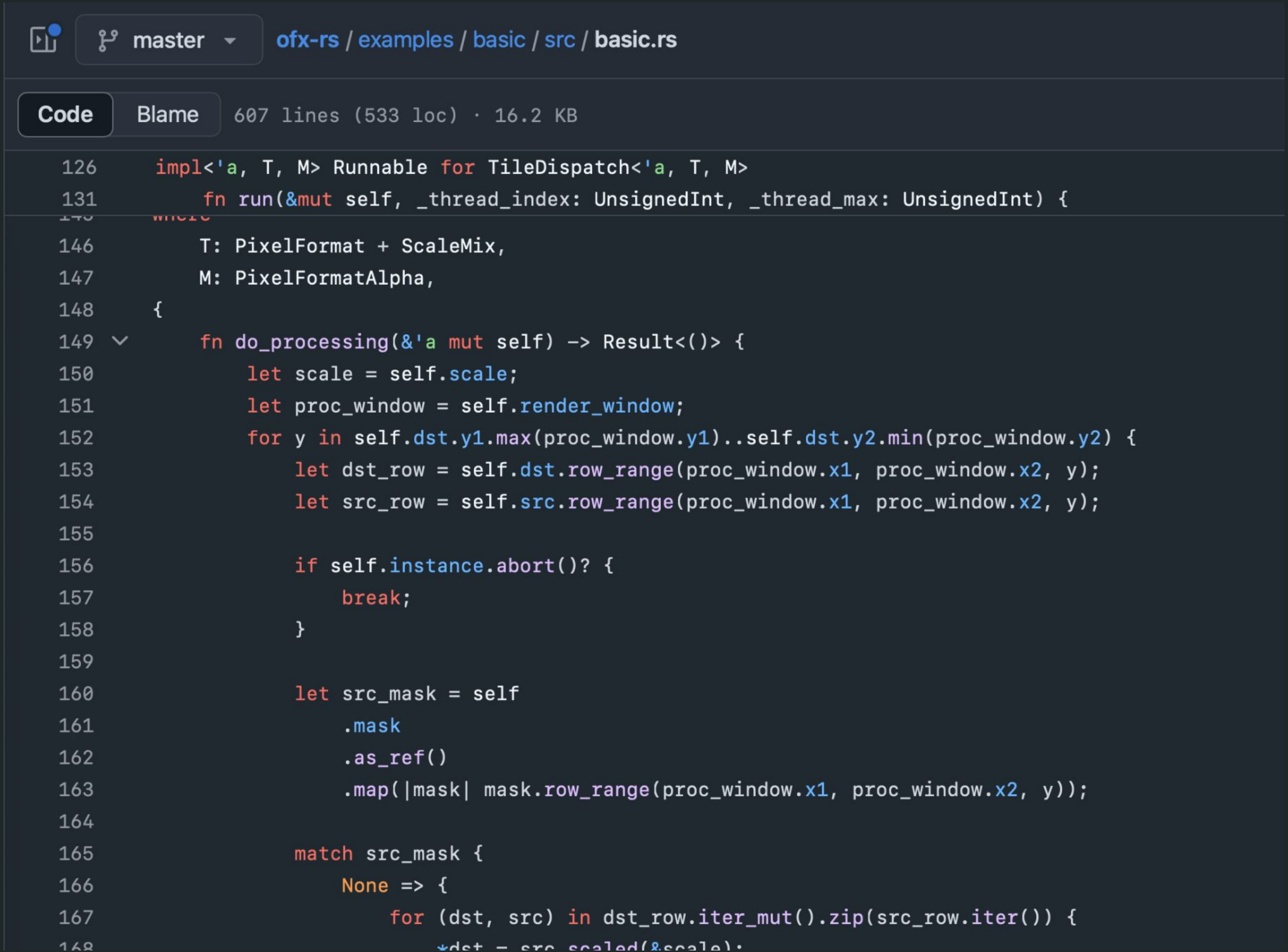
Resolve FX Color

ACES Transform

Chromatic Adaptation

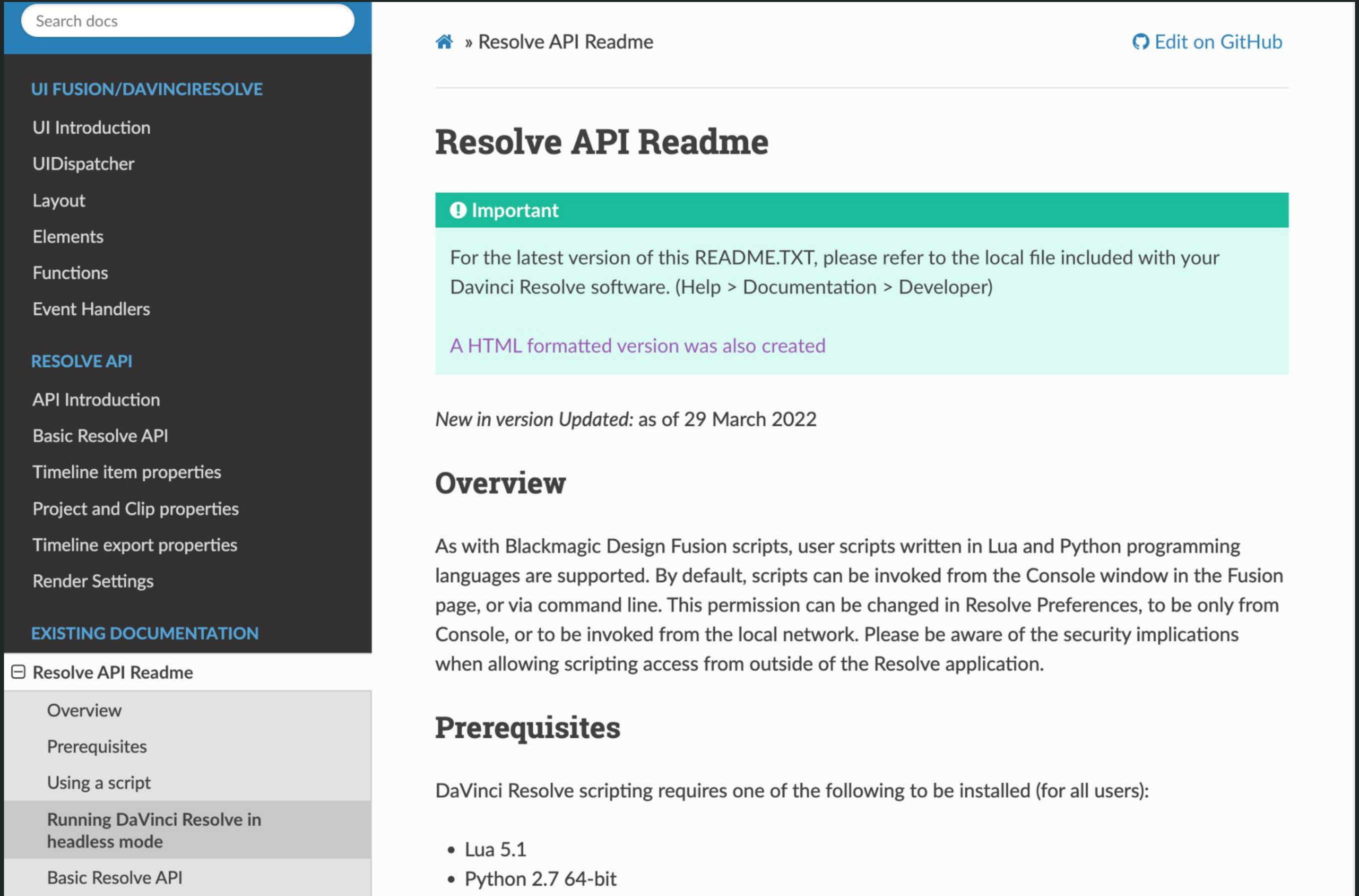
Color Compressor

Color Space Transform


master
ofx-rs / examples / basic / src / basic.rs
Code
Blame
607 lines (533 loc) · 16.2 KB

```
impl<'a, T, M> Runnable for TileDispatch<'a, T, M>
    fn run(&mut self, _thread_index: UnsignedInt, _thread_max: UnsignedInt) {
where
    T: PixelFormat + ScaleMix,
    M: PixelFormatAlpha,
{
    fn do_processing(&'a mut self) -> Result<()> {
        let scale = self.scale;
        let proc_window = self.render_window;
        for y in self.dst.y1.max(proc_window.y1)..self.dst.y2.min(proc_window.y2) {
            let dst_row = self.dst.row_range(proc_window.x1, proc_window.x2, y);
            let src_row = self.src.row_range(proc_window.x1, proc_window.x2, y);

            if self.instance.abort()? {
                break;
            }

            let src_mask = self
                .mask
                .as_ref()
                .map(|mask| mask.row_range(proc_window.x1, proc_window.x2, y));

            match src_mask {
                None => {
                    for (dst, src) in dst_row.iter_mut().zip(src_row.iter()) {
                        *dst = src.scaled(&scale);
```

Resolve scripting API



Search docs

UI FUSION/DAVINCIRESOLVE

- UI Introduction
- UIDispatcher
- Layout
- Elements
- Functions
- Event Handlers

RESOLVE API

- API Introduction
- Basic Resolve API
- Timeline item properties
- Project and Clip properties
- Timeline export properties
- Render Settings

EXISTING DOCUMENTATION

- ⊟ Resolve API Readme
  - Overview
  - Prerequisites
  - Using a script
  - Running DaVinci Resolve in headless mode
  - Basic Resolve API

» Resolve API Readme

Edit on GitHub

Resolve API Readme

Important

For the latest version of this README.TXT, please refer to the local file included with your Davinci Resolve software. (Help > Documentation > Developer)

A HTML formatted version was also created

New in version Updated: as of 29 March 2022

Overview

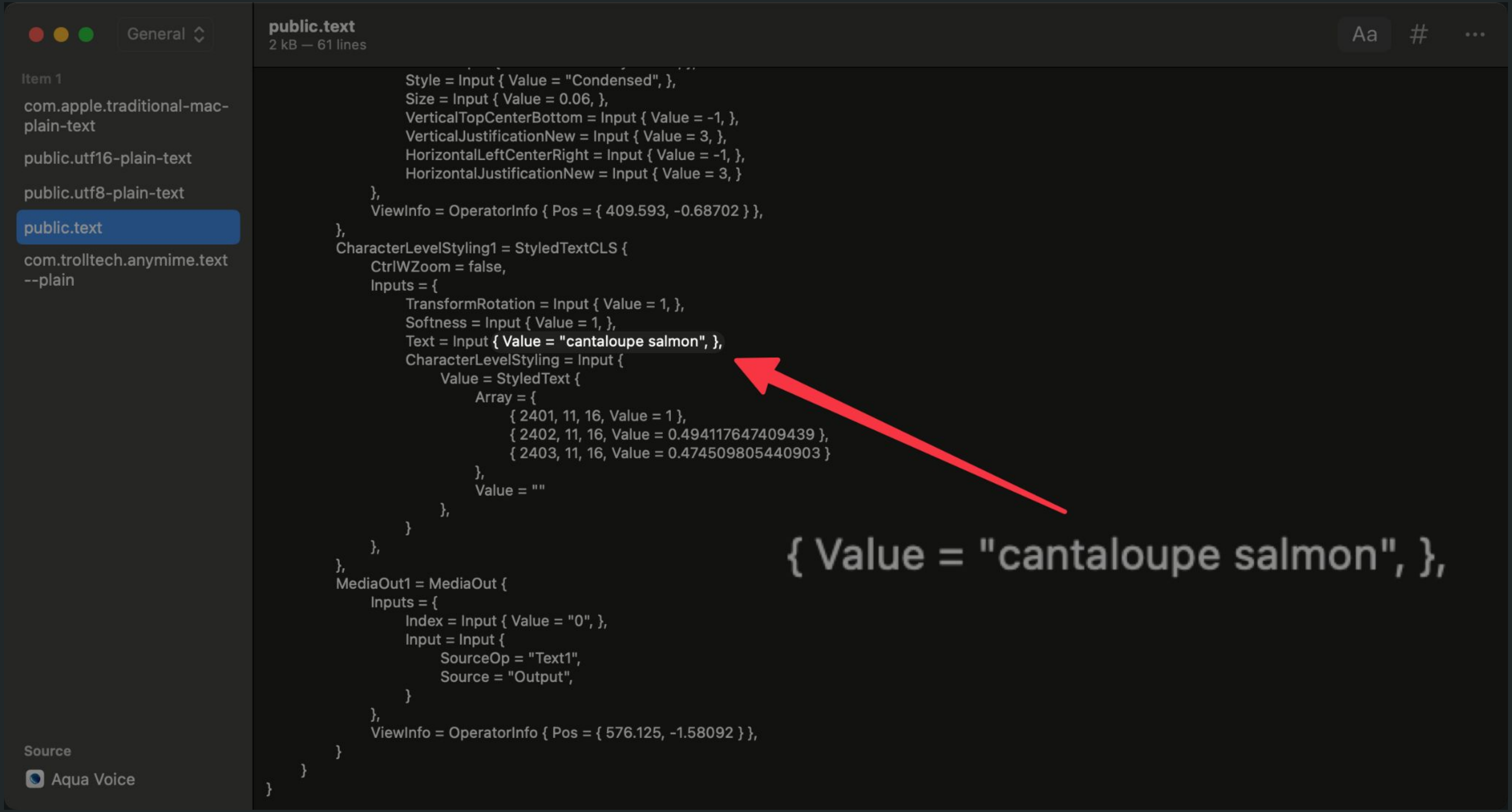
As with Blackmagic Design Fusion scripts, user scripts written in Lua and Python programming languages are supported. By default, scripts can be invoked from the Console window in the Fusion page, or via command line. This permission can be changed in Resolve Preferences, to be only from Console, or to be invoked from the local network. Please be aware of the security implications when allowing scripting access from outside of the Resolve application.

Prerequisites

DaVinci Resolve scripting requires one of the following to be installed (for all users):

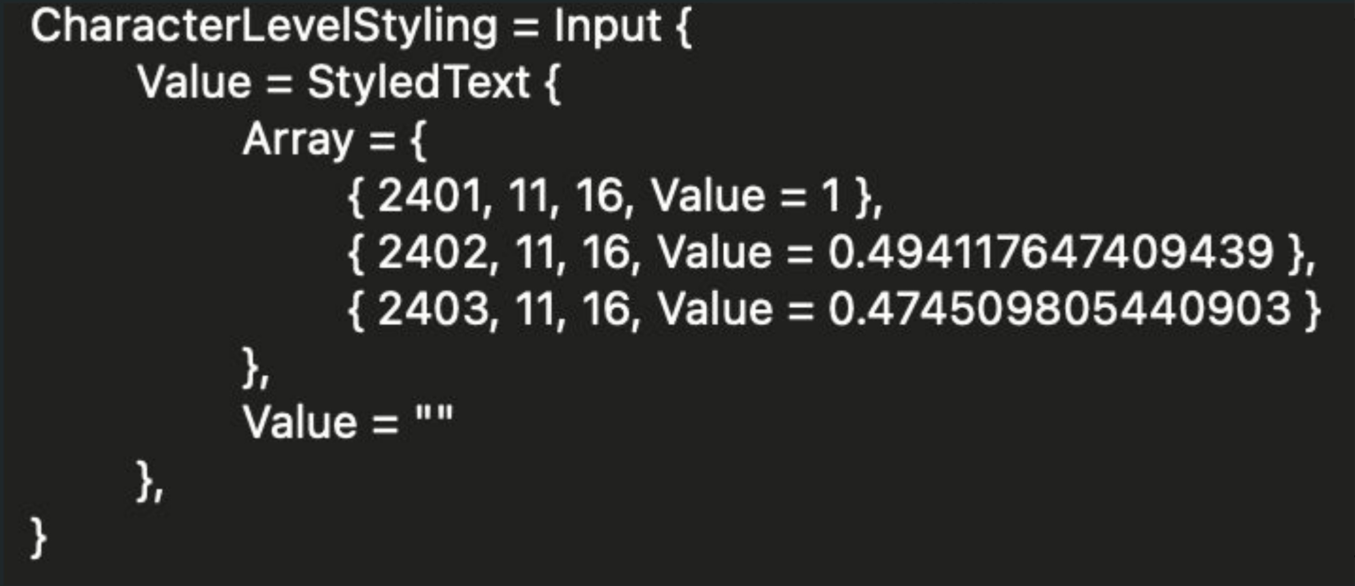
- Lua 5.1
- Python 2.7 64-bit

It's Lua all the way down!

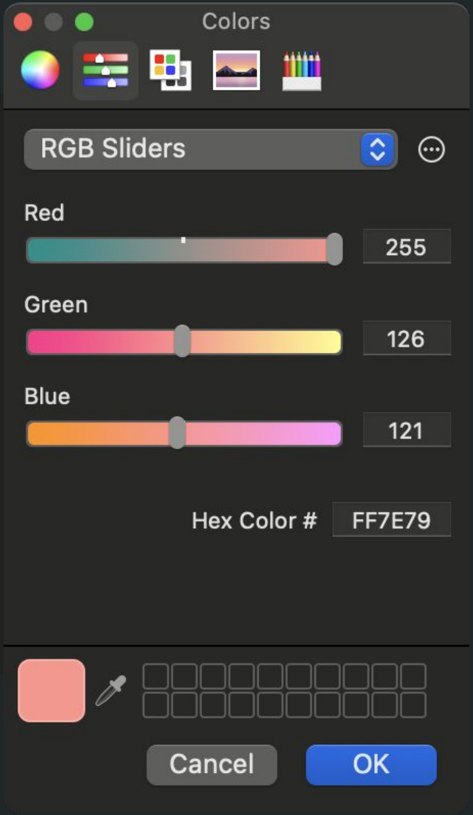


```
                Style = Input { Value = "Condensed", },
                Size = Input { Value = 0.06, },
                VerticalTopCenterBottom = Input { Value = -1, },
                VerticalJustificationNew = Input { Value = 3, },
                HorizontalLeftCenterRight = Input { Value = -1, },
                HorizontalJustificationNew = Input { Value = 3, }
            },
            ViewInfo = OperatorInfo { Pos = { 409.593, -0.68702 } },
        },
        CharacterLevelStyling1 = StyledTextCLS {
            CtrlWZoom = false,
            Inputs = {
                TransformRotation = Input { Value = 1, },
                Softness = Input { Value = 1, },
                Text = Input { Value = "cantaloupe salmon", },
                CharacterLevelStyling = Input {
                    Value = StyledText {
                        Array = {
                            { 2401, 11, 16, Value = 1 },
                            { 2402, 11, 16, Value = 0.494117647409439 },
                            { 2403, 11, 16, Value = 0.474509805440903 }
                        },
                        Value = ""
                    },
                }
            },
        },
        MediaOut1 = MediaOut {
            Inputs = {
                Index = Input { Value = "0", },
                Input = Input {
                    SourceOp = "Text1",
                    Source = "Output",
                }
            },
            ViewInfo = OperatorInfo { Pos = { 576.125, -1.58092 } },
        }
    }
}
```

{ Value = "cantaloupe salmon", },



```
CharacterLevelStyling = Input {
        Value = StyledText {
                Array = {
                        { 2401, 11, 16, Value = 1 },
                        { 2402, 11, 16, Value = 0.494117647409439 },
                        { 2403, 11, 16, Value = 0.474509805440903 }
                },
                Value = ""
        },
}
```



RGB Sliders

Red 255

Green 126

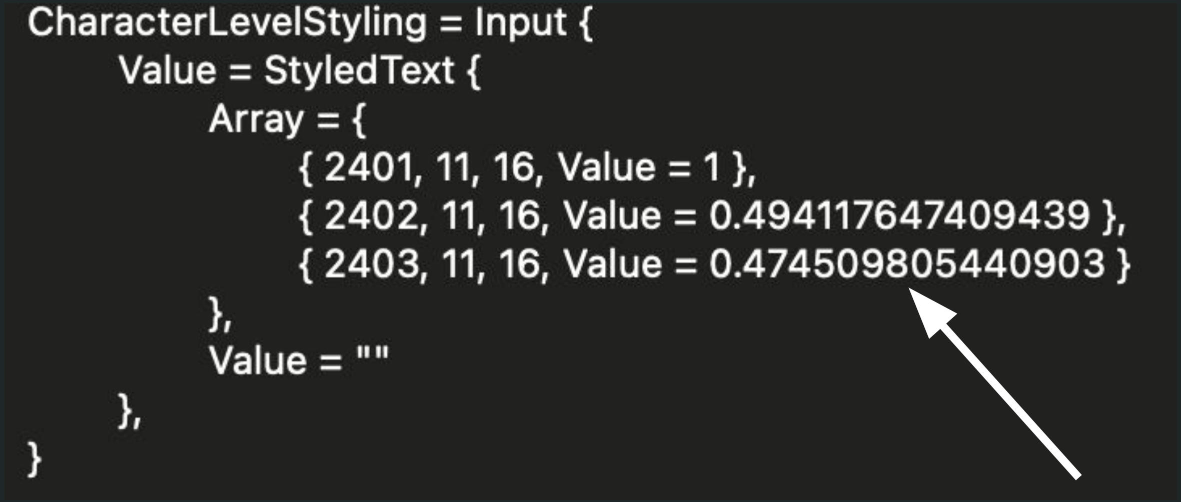
Blue 121

Hex Color # FF7E79

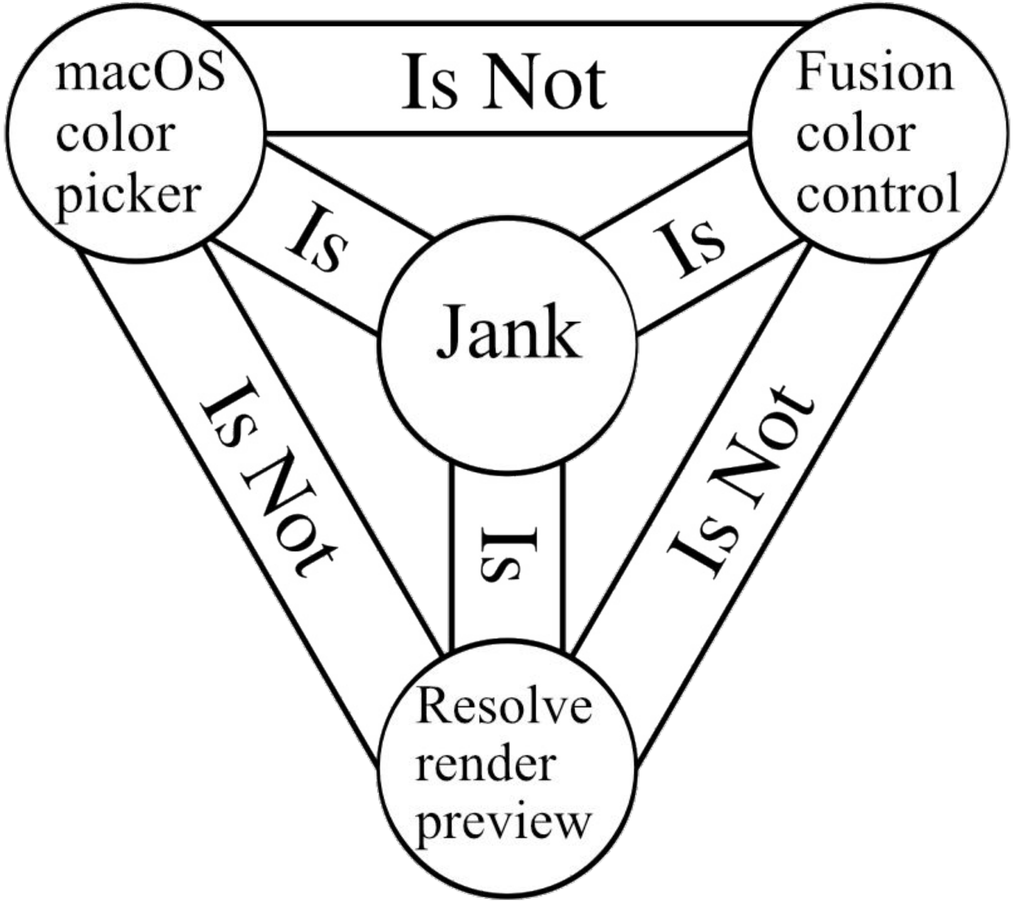
1) Generate lua into clipboard
2) Paste into Fusion
3) Profit!

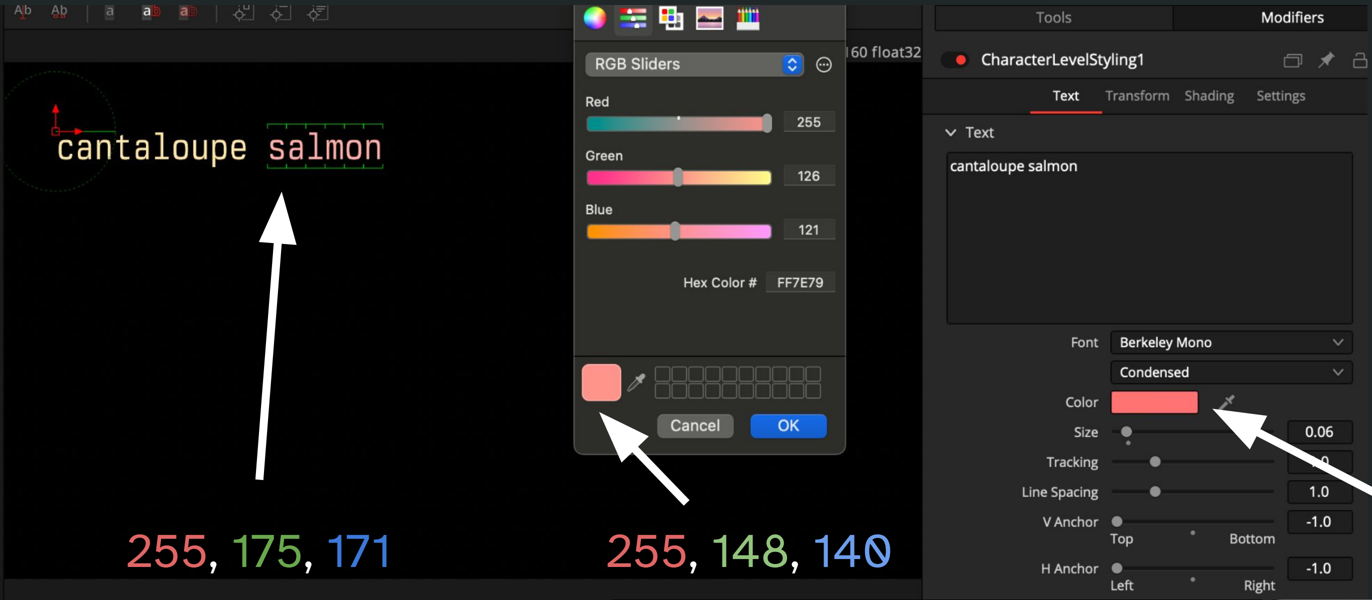
Detail 1

what does RGB 0-255 mean?



```
CharacterLevelStyling = Input {
        Value = StyledText {
                Array = {
                        { 2401, 11, 16, Value = 1 },
                        { 2402, 11, 16, Value = 0.494117647409439 },
                        { 2403, 11, 16, Value = 0.474509805440903 }
                },
                Value = ""
        },
}
```


macOS color picker
Is Not
Fusion color control
Is
Is
Jank
Is Not
Is
Is Not
Resolve render preview



cantaloupe salmon

RGB Sliders

Red 255

Green 126

Blue 121

Hex Color # FF7E79

CharacterLevelStyling1

Text

cantaloupe salmon

Font: Berkeley Mono

Condensed

Color

Size 0.06

Tracking

Line Spacing 1.0

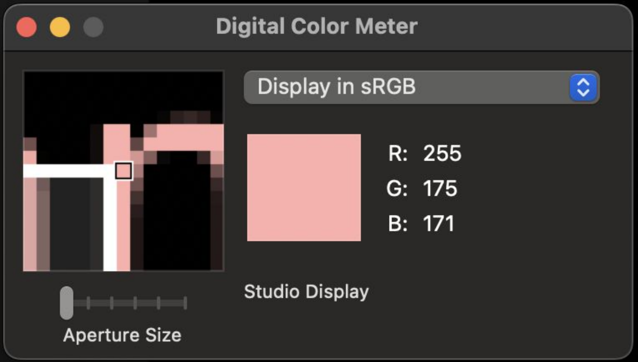
V Anchor -1.0 (Top – Bottom)

H Anchor -1.0 (Left – Right)

255, 175, 171

255, 148, 140

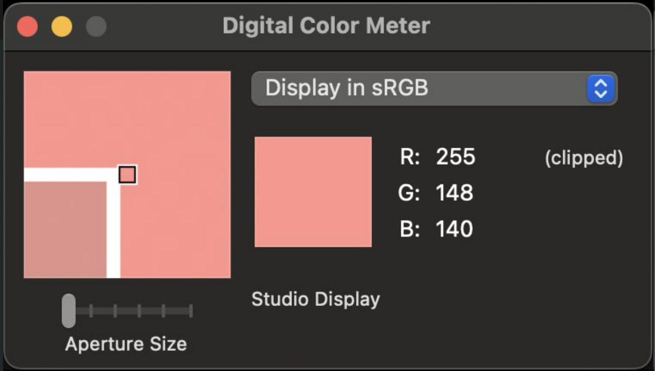
255, 115, 115



Display in sRGB

R: 255
G: 175
B: 171

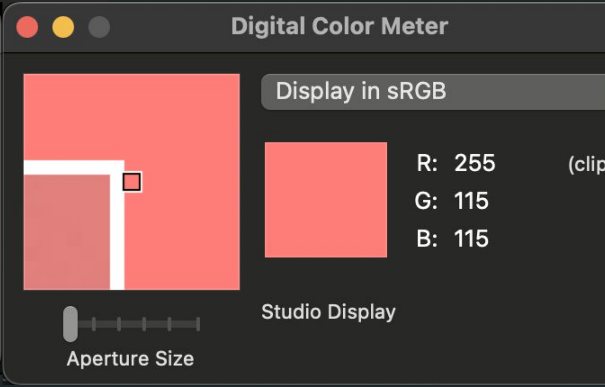
Studio Display



Display in sRGB

R: 255 (clipped)
G: 148
B: 140

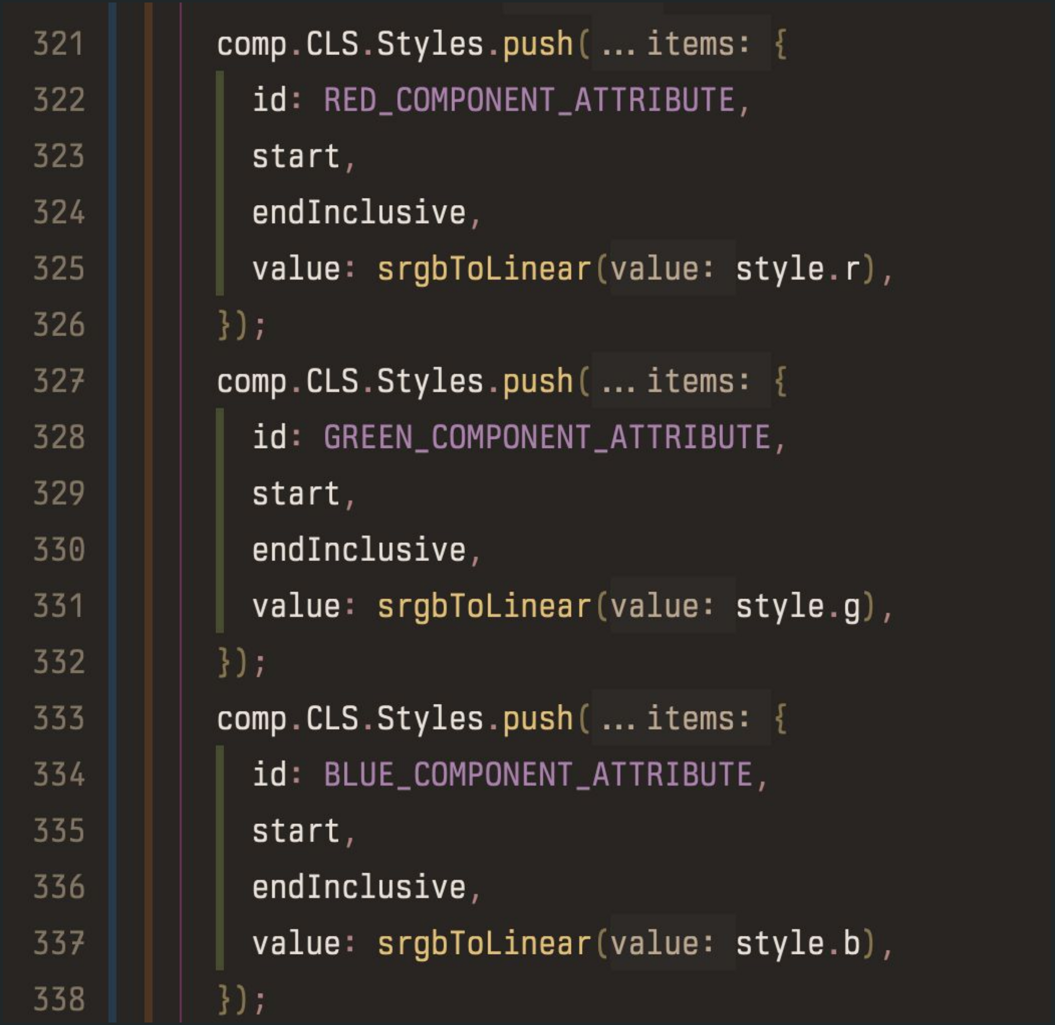
Studio Display



Display in sRGB

R: 255 (clip
G: 115
B: 115

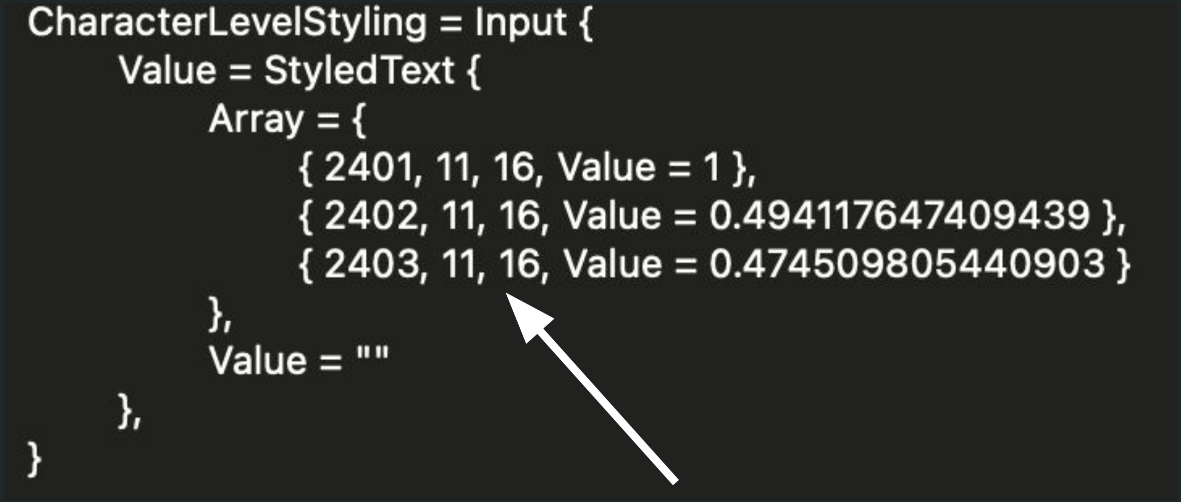
Studio Display



```
comp.CLS.Styles.push(...items: {
    id: RED_COMPONENT_ATTRIBUTE,
    start,
    endInclusive,
    value: srgbToLinear(value: style.r),
});
comp.CLS.Styles.push(...items: {
    id: GREEN_COMPONENT_ATTRIBUTE,
    start,
    endInclusive,
    value: srgbToLinear(value: style.g),
});
comp.CLS.Styles.push(...items: {
    id: BLUE_COMPONENT_ATTRIBUTE,
    start,
    endInclusive,
    value: srgbToLinear(value: style.b),
});
```

Detail 2

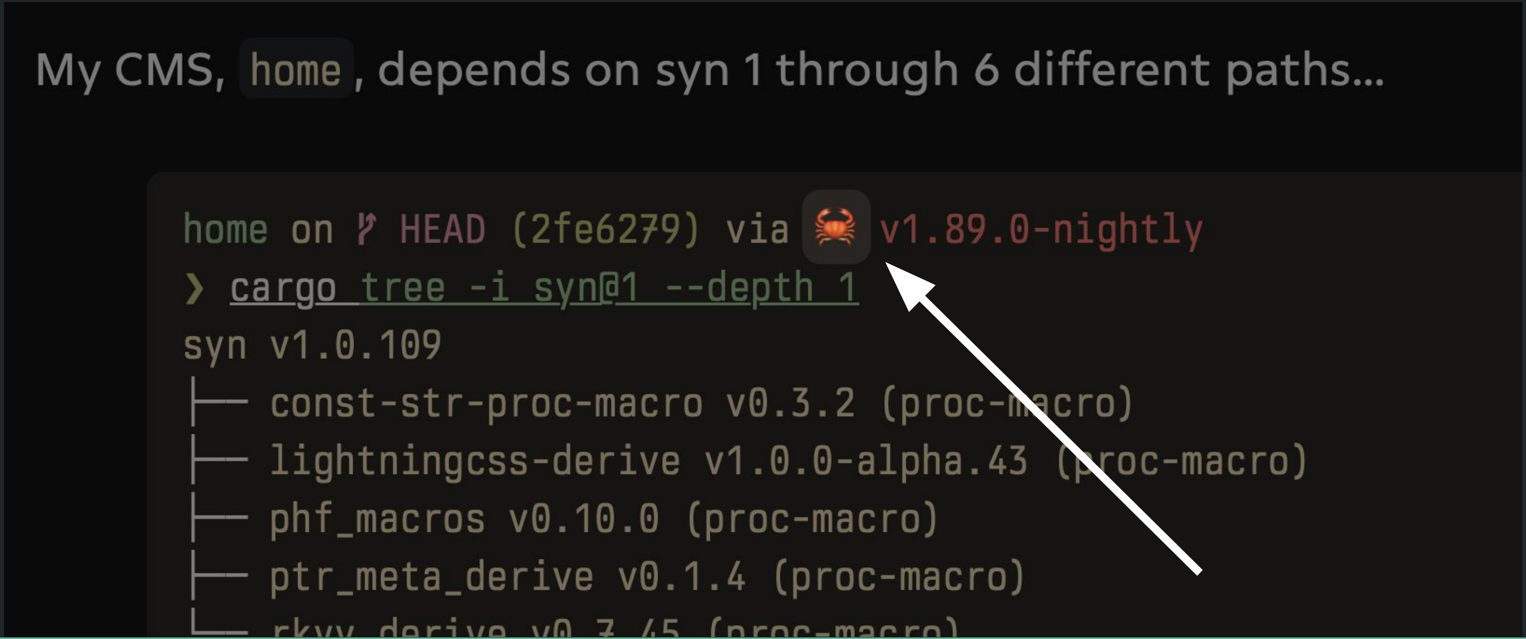
how do we define text ranges?



```
CharacterLevelStyling = Input {
        Value = StyledText {
                Array = {
                        { 2401, 11, 16, Value = 1 },
                        { 2402, 11, 16, Value = 0.494117647409439 },
                        { 2403, 11, 16, Value = 0.474509805440903 }
                },
                Value = ""
        },
}
```

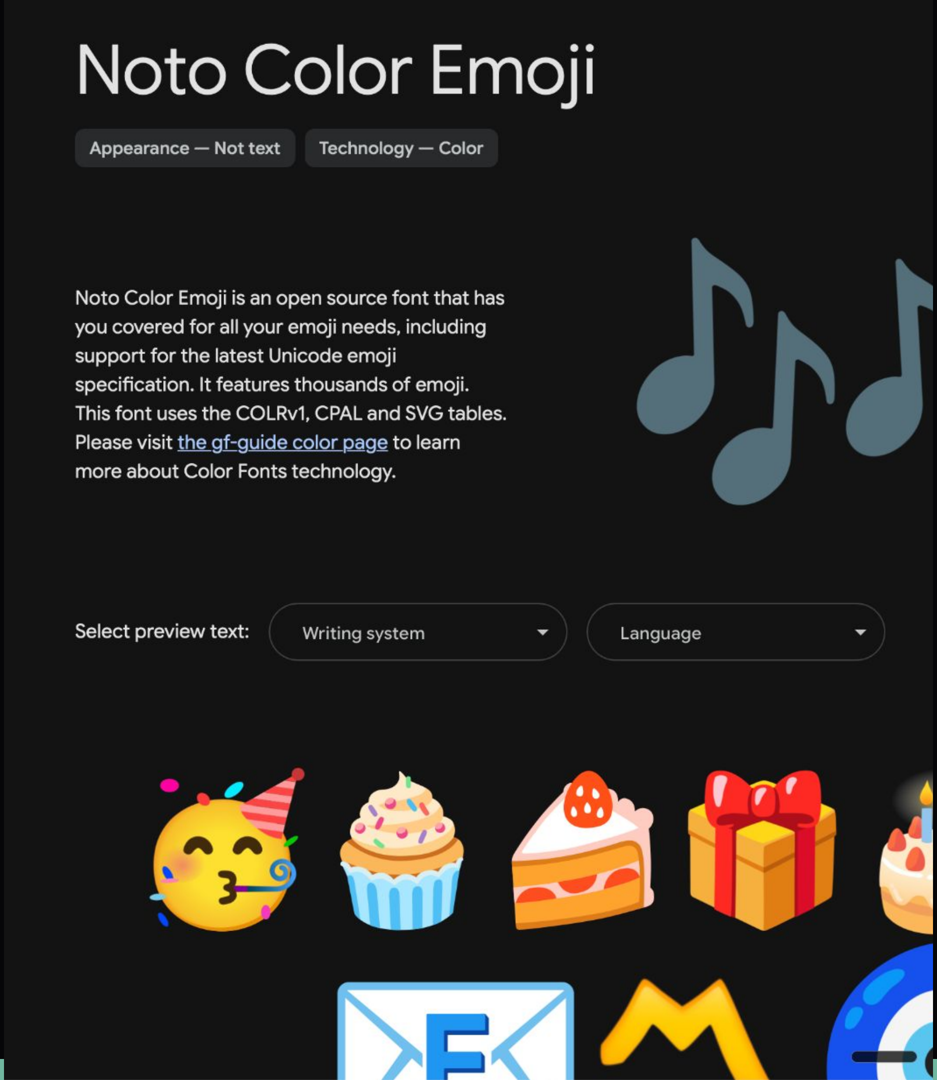
Detail 2b

how do we deal with emojis?



My CMS, `home`, depends on syn 1 through 6 different paths...

```
home on  HEAD (2fe6279) via 🦀 v1.89.0-nightly
❯ cargo tree -i syn@1 --depth 1
syn v1.0.109
├── const-str-proc-macro v0.3.2 (proc-macro)
├── lightningcss-derive v1.0.0-alpha.43 (proc-macro)
├── phf_macros v0.10.0 (proc-macro)
├── ptr_meta_derive v0.1.4 (proc-macro)
└── rkyv_derive v0.7.45 (proc-macro)
```

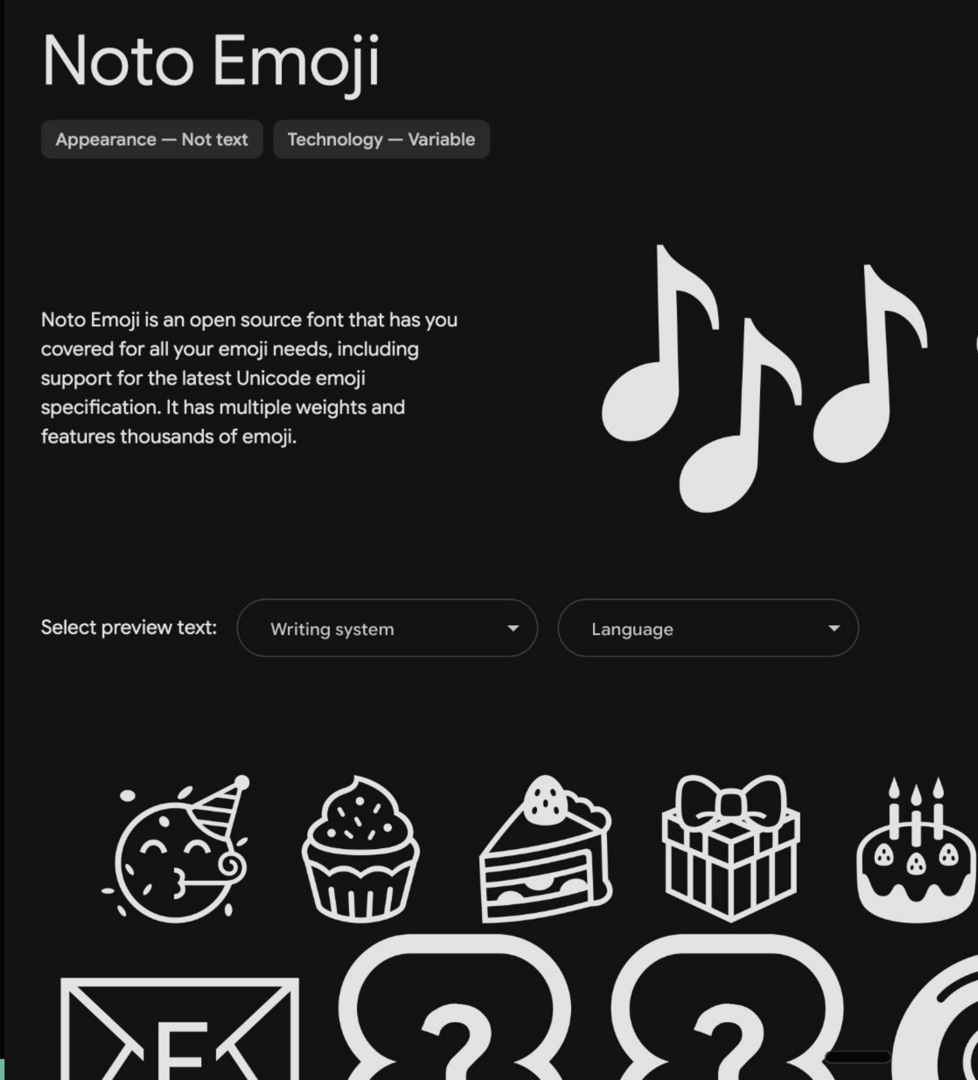


Noto Color Emoji

Appearance — Not text | Technology — Color

Noto Color Emoji is an open source font that has you covered for all your emoji needs, including support for the latest Unicode emoji specification. It features thousands of emoji. This font uses the COLRv1, CPAL and SVG tables. Please visit the gf-guide color page to learn more about Color Fonts technology.

Select preview text: Writing system ▾ Language ▾



Noto Emoji

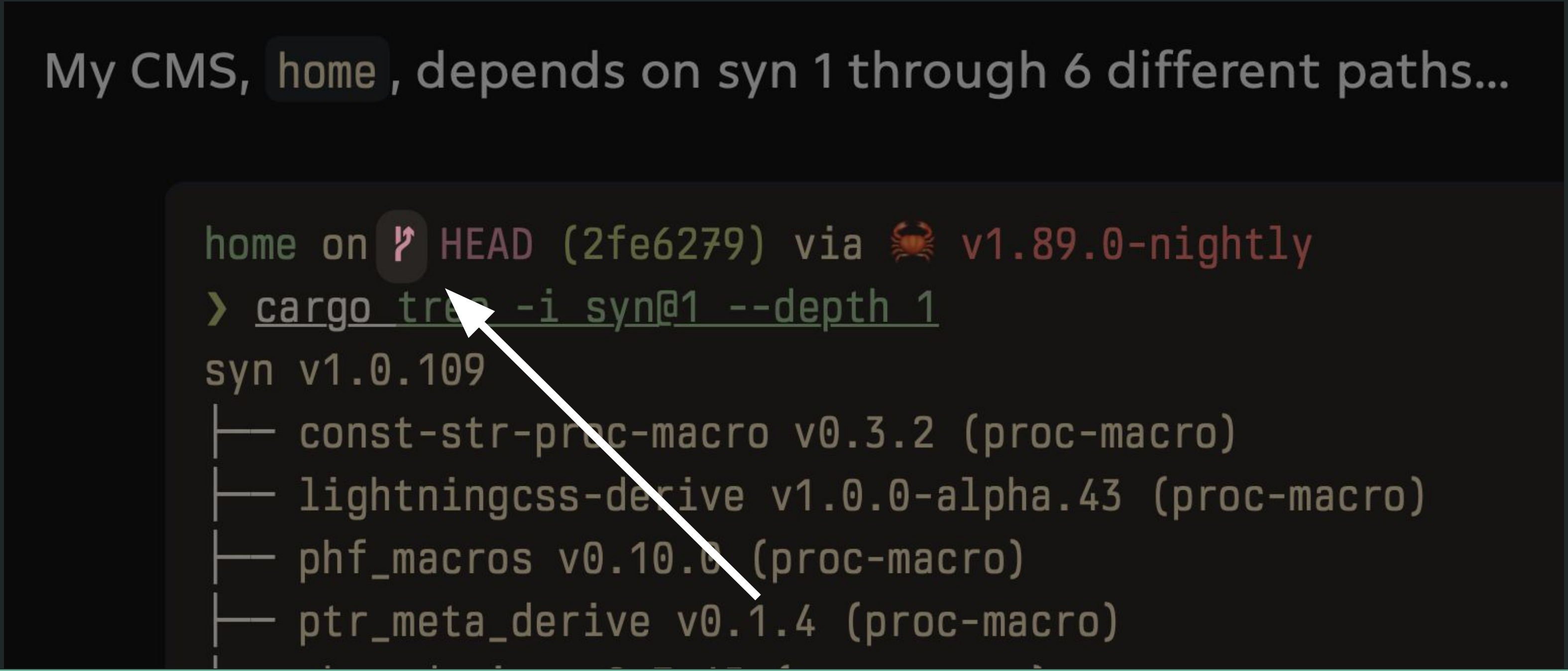
Appearance — Not text | Technology — Variable

Noto Emoji is an open source font that has you covered for all your emoji needs, including support for the latest Unicode emoji specification. It has multiple weights and features thousands of emoji.

Select preview text: Writing system ▾ Language ▾

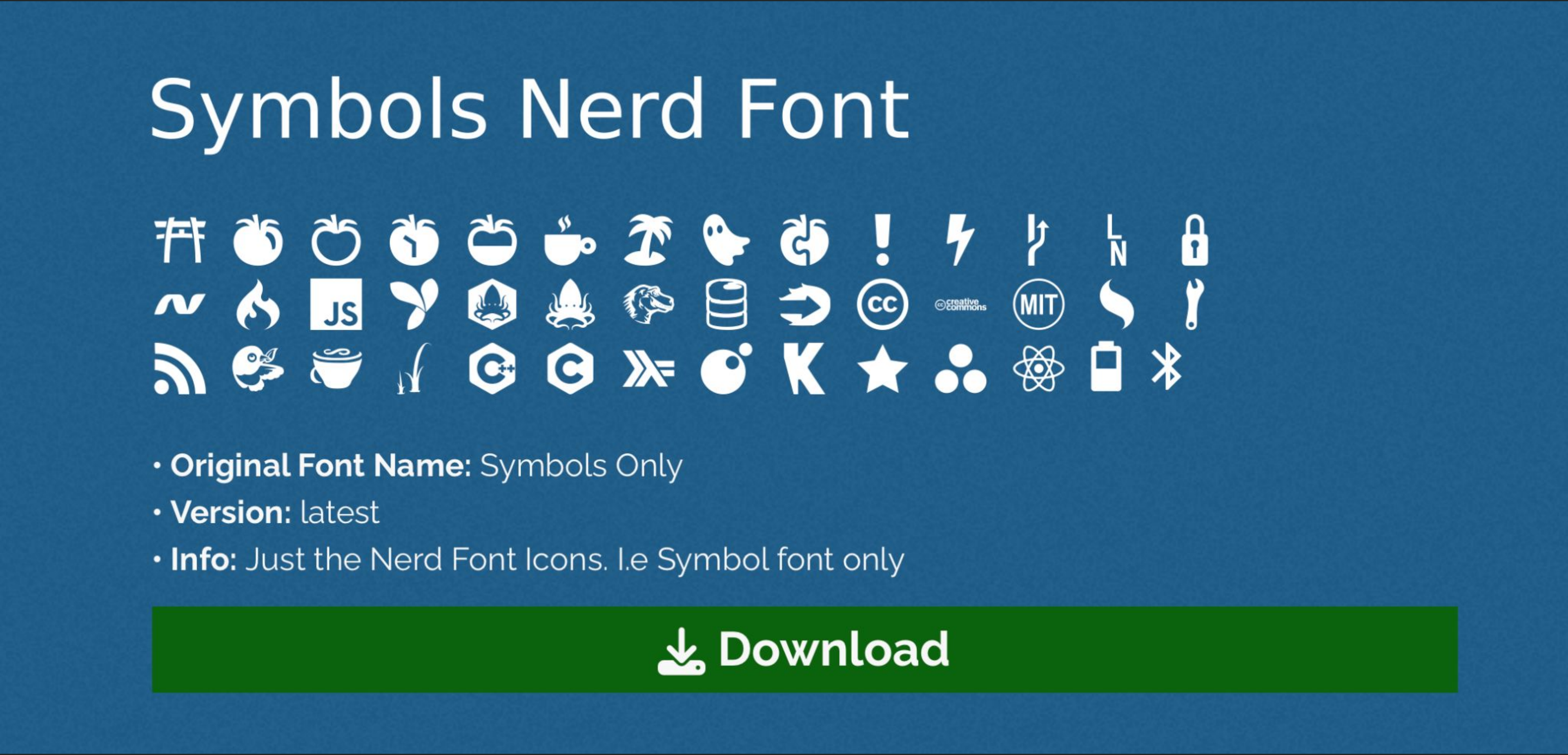
Detail 2c

how do we deal with nerdfonts?



My CMS, `home`, depends on syn 1 through 6 different paths...

```
home on  HEAD (2fe6279) via 🦀 v1.89.0-nightly
❯ cargo tree -i syn@1 --depth 1
syn v1.0.109
├── const-str-proc-macro v0.3.2 (proc-macro)
├── lightningcss-derive v1.0.0-alpha.43 (proc-macro)
├── phf_macros v0.10.0 (proc-macro)
├── ptr_meta_derive v0.1.4 (proc-macro)
```



Symbols Nerd Font

- **Original Font Name:** Symbols Only
- **Version:** latest
- **Info:** Just the Nerd Font Icons. I.e Symbol font only

Let's play:
what's the length of this string?

"$"

Grapheme clusters: ?

Unicode code points: ?

UTF-16 code units: ?

UTF-8 bytes: ?

"$"

Grapheme clusters: 1

Unicode code points: 1 [U+0024]

UTF-16 code units: 1 [0024]

UTF-8 bytes: 1 [24]

"🦀"

Grapheme clusters: ?

Unicode code points: ?

UTF-16 code units: ?

UTF-8 bytes: ?

"🦀"

Grapheme clusters: 1

Unicode code points: 1 [U+1F980]

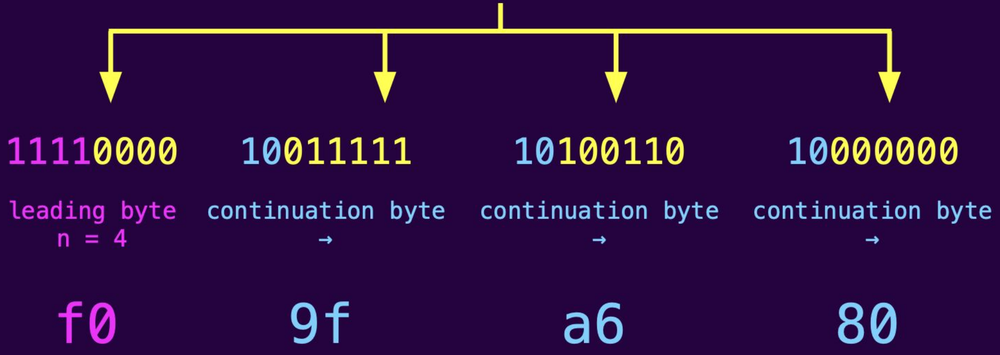
UTF-16 code units: 2 [D83E, DD80]

UTF-8 bytes: 4 [F0, 9F, A6, 80]

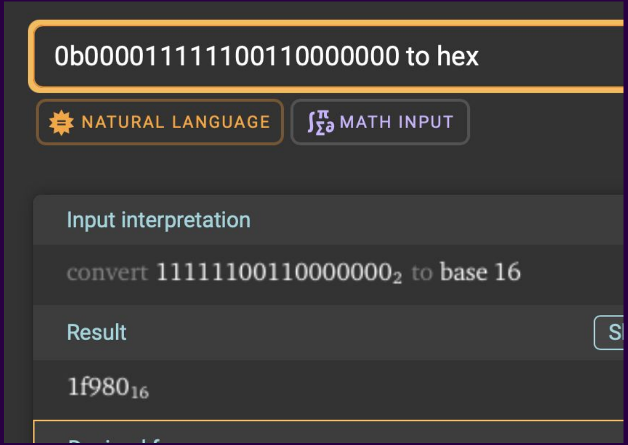
https://sonarsource.github.io/utf8-visualizer/
https://utf-8-visualizer.ardis.lu/

UTF-8 Visualizer

U+1F980



11110000	10011111	10100110	10000000
leading byte n = 4	continuation byte →	continuation byte →	continuation byte →
f0	9f	a6	80



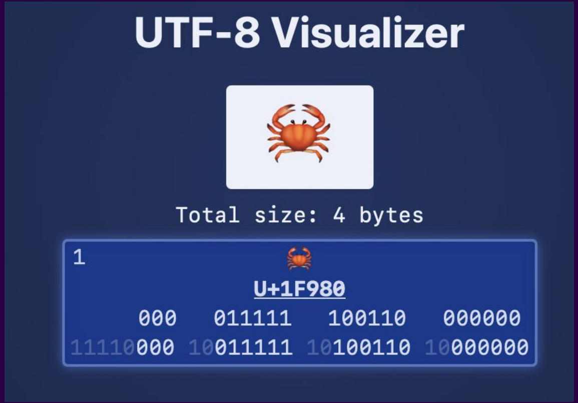
0b000011111100110000000 to hex

Input interpretation

convert 11111100110000000_2 to base 16

Result

$1f980_{16}$



UTF-8 Visualizer

Total size: 4 bytes

```
1
           U+1F980
    000     011111    100110    000000
11110000  10011111  10100110  10000000
```

"✂️"

Grapheme clusters: ?

Unicode code points: ?

UTF-16 code units: ?

UTF-8 bytes: ?

"✂️"

Grapheme clusters: 1

Unicode code points: 2 [U+2702, U+FE0F]

UTF-16 code units: 2 [2702, FE0F]

UTF-8 bytes: 6 [E2, 9C, 82, EF, B8, 8F]

U+2702 "✂"

+

U+FE0F (image variation selector)

=

"✂️"

"👨‍👩‍👧‍👦"

Grapheme clusters: ?

Unicode code points: ?

UTF-16 code units: ?

UTF-8 bytes: ?

"👨‍👩‍👧‍👦️"

Grapheme clusters: 1

Unicode code points: 8 [U+1F468, U+200D, U+1F469, U+200D, U+1F467, U+200D, U+1F466, U+FE0F]

UTF-16 code units: 12 [D83D, DC68, 200D, D83D, DC69, 200D, D83D, DC67, 200D, D83D, DC66, FE0F]

UTF-8 bytes: 28 [F0, 9F, 91, A8, E2, 80, 8D, F0, 9F, 91, A9, E2, 80, 8D, F0, 9F, 91, A7, E2, 80, 8D, F0, 9F, 91, A6, EF, B8, 8F]

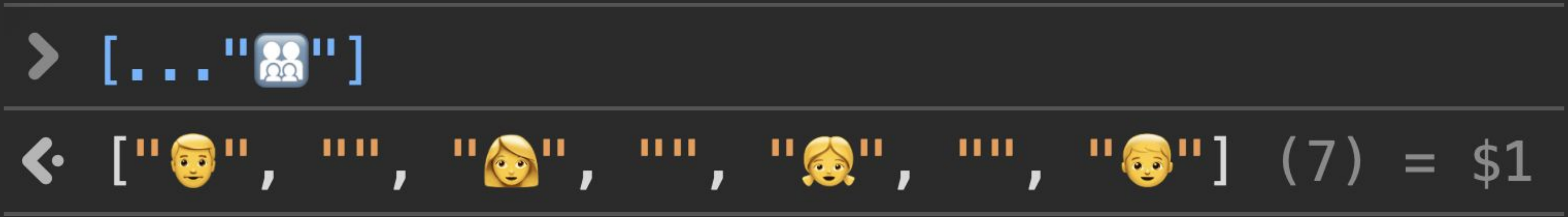
"👨‍👩‍👧‍👦️"

Grapheme clusters: 1

Unicode code points: 8 [U+1F468, U+200D, U+1F469, U+200D, U+1F467, U+200D, U+1F466, U+FE0F]

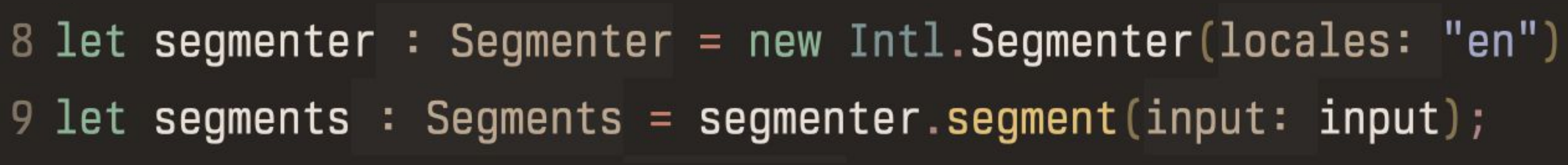
UTF-16 code units: 12 [D83D, DC68, 200D, D83D, DC69, 200D, D83D, DC67, 200D, D83D, DC66, FE0F]

UTF-8 bytes: 28 [F0, 9F, 91, A8, E2, 80, 8D, F0, 9F, 91, A9, E2, 80, 8D, F0, 9F, 91, A7, E2, 80, 8D, F0, 9F, 91, A6, EF, B8, 8F]

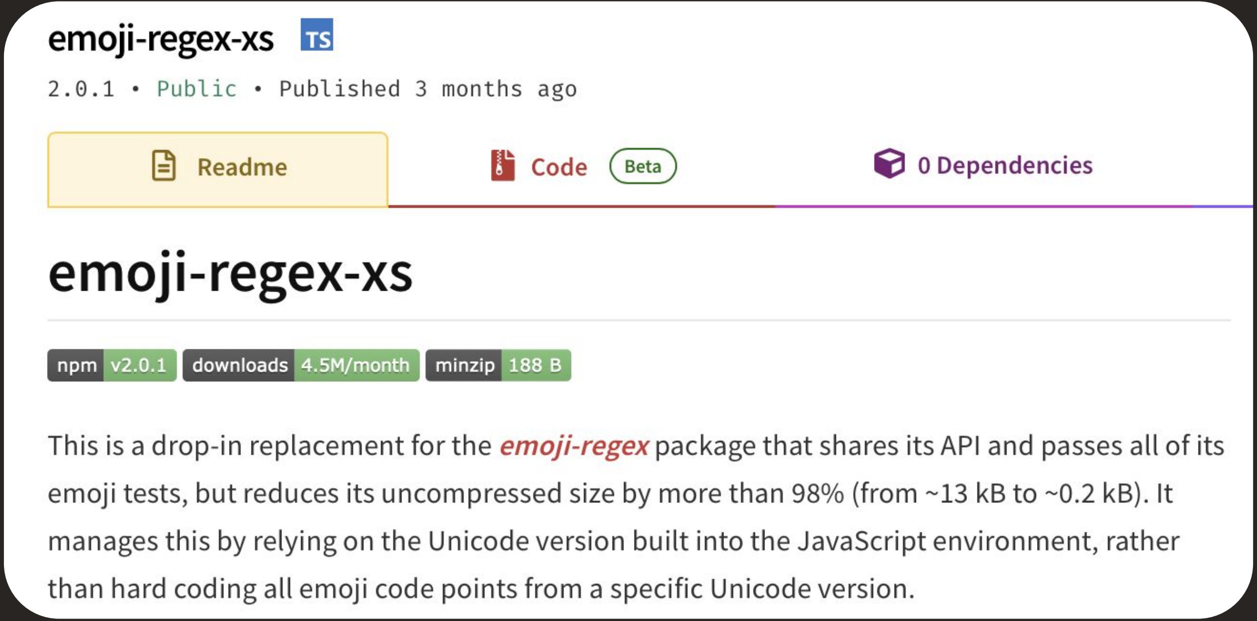


```
> [..."👨‍👩‍👧‍👦"]
<· ["👨", "‍", "👩", "‍", "👧", "‍", "👦"] (7) = $1
```

How do you identify emojis?



```
let segmenter : Segmenter = new Intl.Segmenter(locales: "en")
let segments : Segments = segmenter.segment(input: input);
```



emoji-regex-xs TS

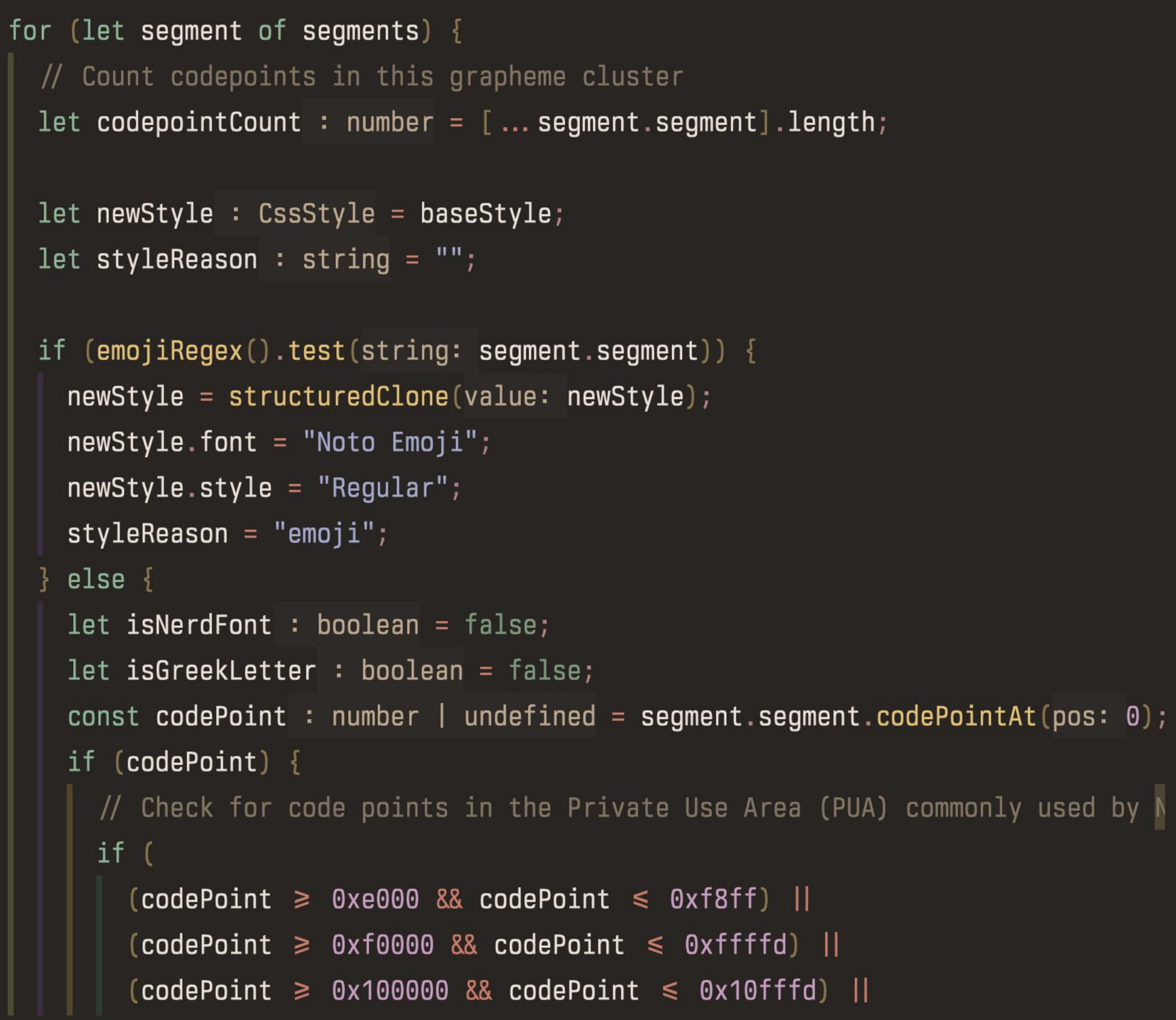
2.0.1 • Public • Published 3 months ago

Readme | Code Beta | 0 Dependencies

emoji-regex-xs

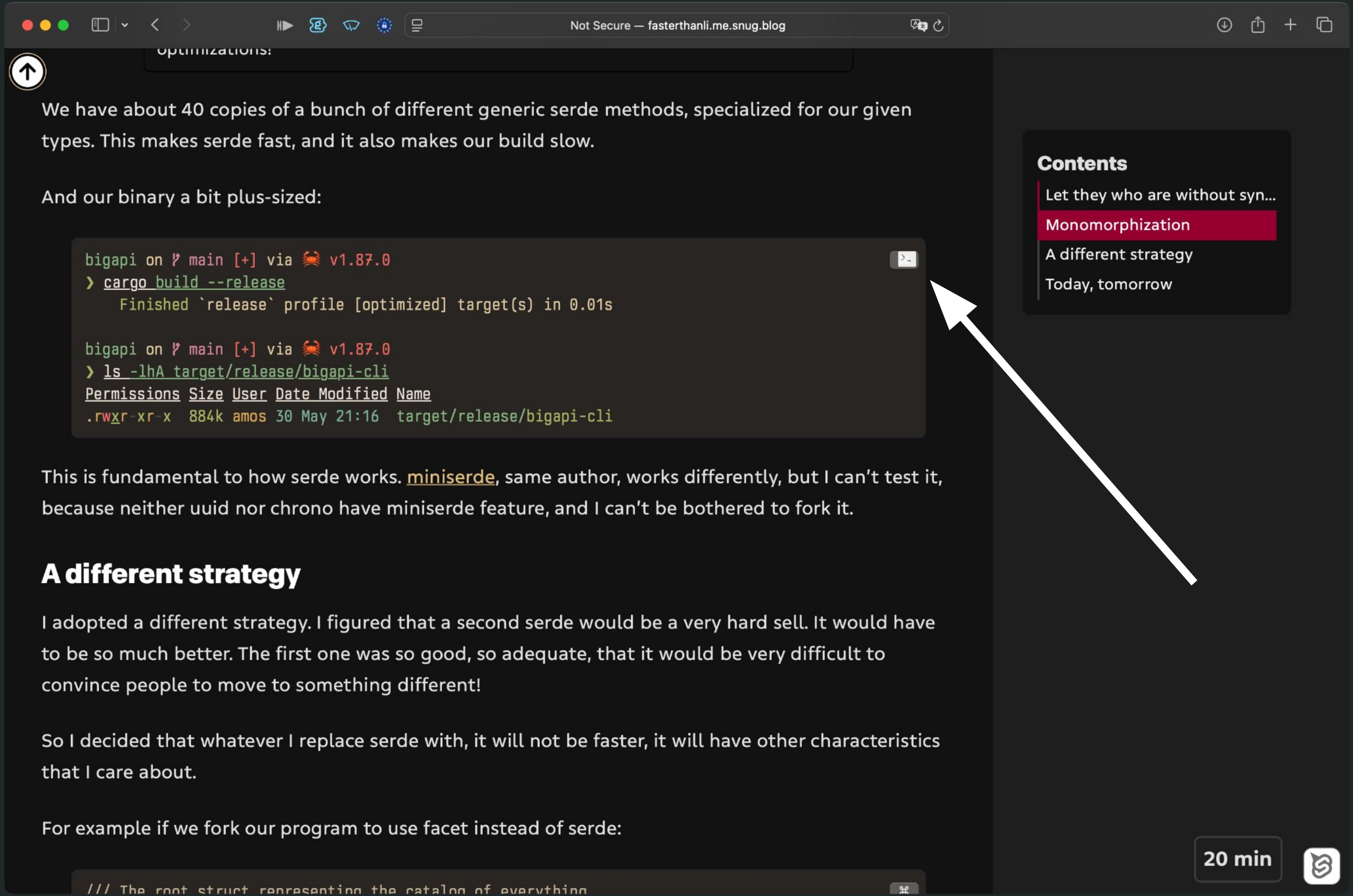
npm v2.0.1 | downloads 4.5M/month | minzip 188 B

This is a drop-in replacement for the *emoji-regex* package that shares its API and passes all of its emoji tests, but reduces its uncompressed size by more than 98% (from ~13 kB to ~0.2 kB). It manages this by relying on the Unicode version built into the JavaScript environment, rather than hard coding all emoji code points from a specific Unicode version.



```
for (let segment of segments) {
  // Count codepoints in this grapheme cluster
  let codepointCount : number = [...segment.segment].length;

  let newStyle : CssStyle = baseStyle;
  let styleReason : string = "";

  if (emojiRegex().test(string: segment.segment)) {
    newStyle = structuredClone(value: newStyle);
    newStyle.font = "Noto Emoji";
    newStyle.style = "Regular";
    styleReason = "emoji";
  } else {
    let isNerdFont : boolean = false;
    let isGreekLetter : boolean = false;
    const codePoint : number | undefined = segment.segment.codePointAt(pos: 0);
    if (codePoint) {
      // Check for code points in the Private Use Area (PUA) commonly used by N
      if (
        (codePoint ≥ 0xe000 && codePoint ≤ 0xf8ff) ||
        (codePoint ≥ 0xf0000 && codePoint ≤ 0xffffd) ||
        (codePoint ≥ 0x100000 && codePoint ≤ 0x10fffd) ||
```

The final workflow



optimizations!

We have about 40 copies of a bunch of different generic serde methods, specialized for our given types. This makes serde fast, and it also makes our build slow.

And our binary a bit plus-sized:

```
bigapi on  main [+] via 🦀 v1.87.0
❯ cargo build --release
    Finished `release` profile [optimized] target(s) in 0.01s

bigapi on  main [+] via 🦀 v1.87.0
❯ ls -lhA target/release/bigapi-cli
Permissions Size User Date Modified Name
.rwxr-xr-x  884k amos 30 May 21:16  target/release/bigapi-cli
```

This is fundamental to how serde works. miniserde, same author, works differently, but I can't test it, because neither uuid nor chrono have miniserde feature, and I can't be bothered to fork it.

A different strategy

I adopted a different strategy. I figured that a second serde would be a very hard sell. It would have to be so much better. The first one was so good, so adequate, that it would be very difficult to convince people to move to something different!

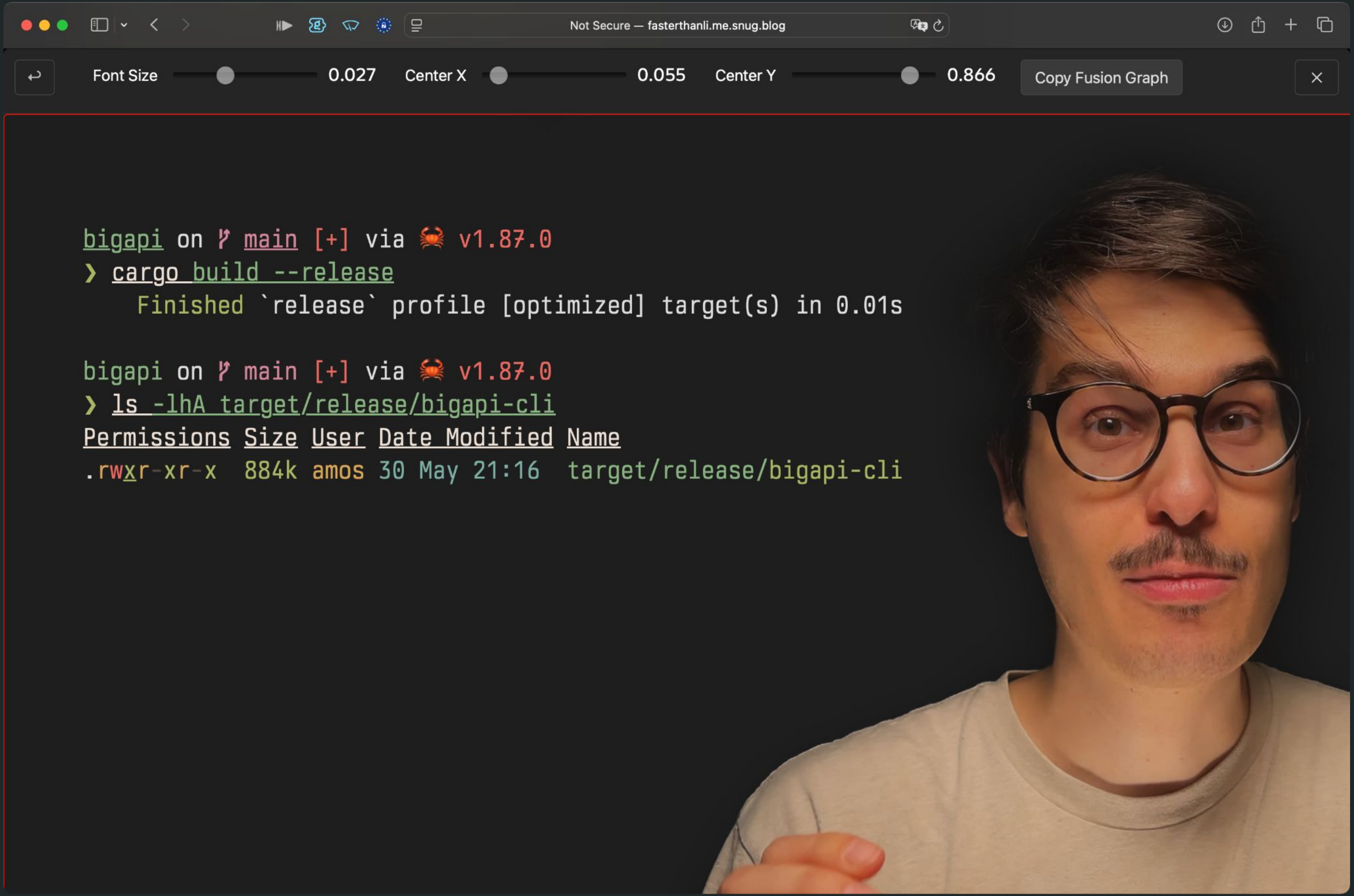
So I decided that whatever I replace serde with, it will not be faster, it will have other characteristics that I care about.

For example if we fork our program to use facet instead of serde:

```
/// The root struct representing the catalog of everything
```

Contents

Let they who are without syn...
Monomorphization
A different strategy
Today, tomorrow

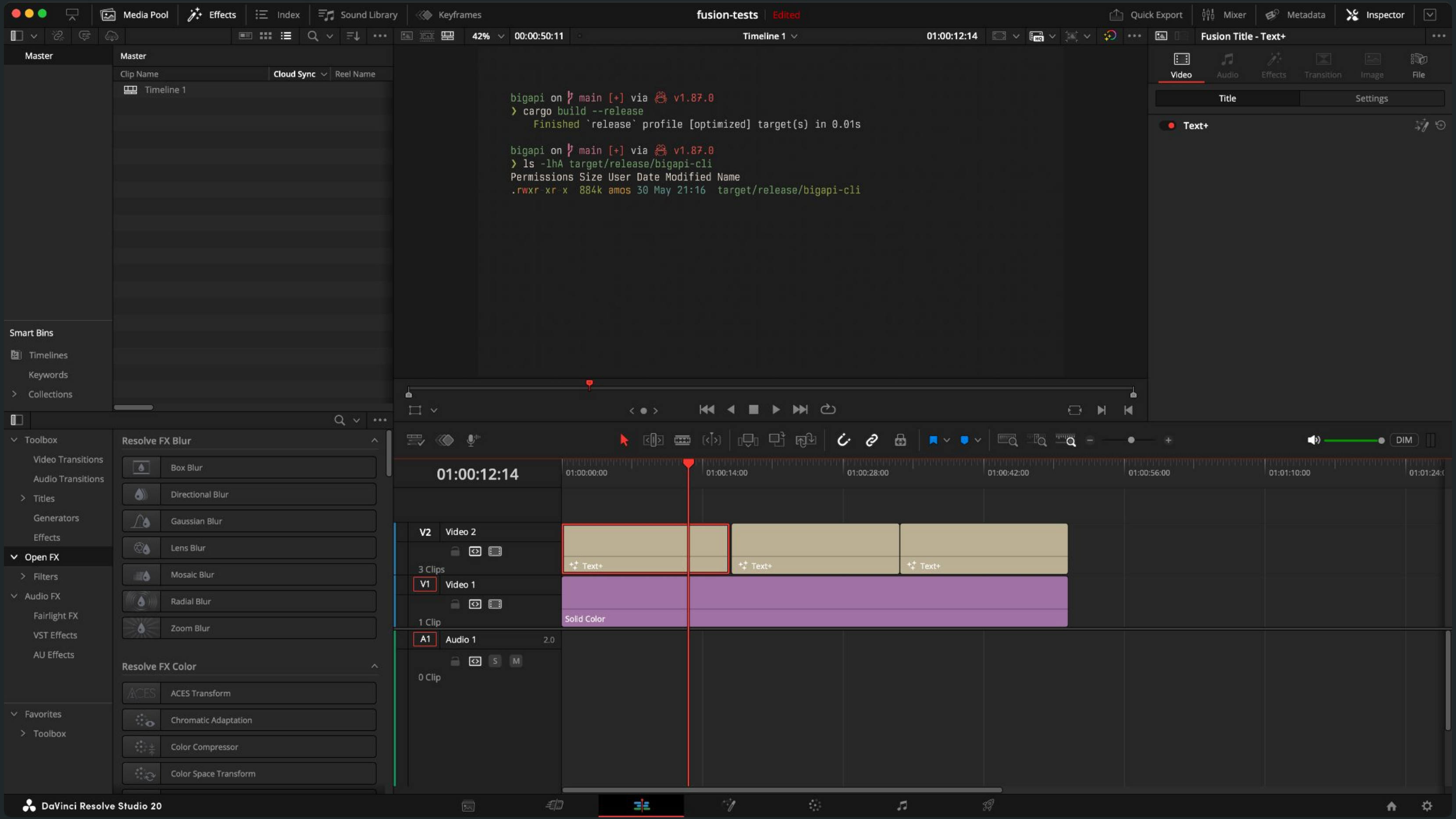


Not Secure — fasterthanli.me.snug.blog

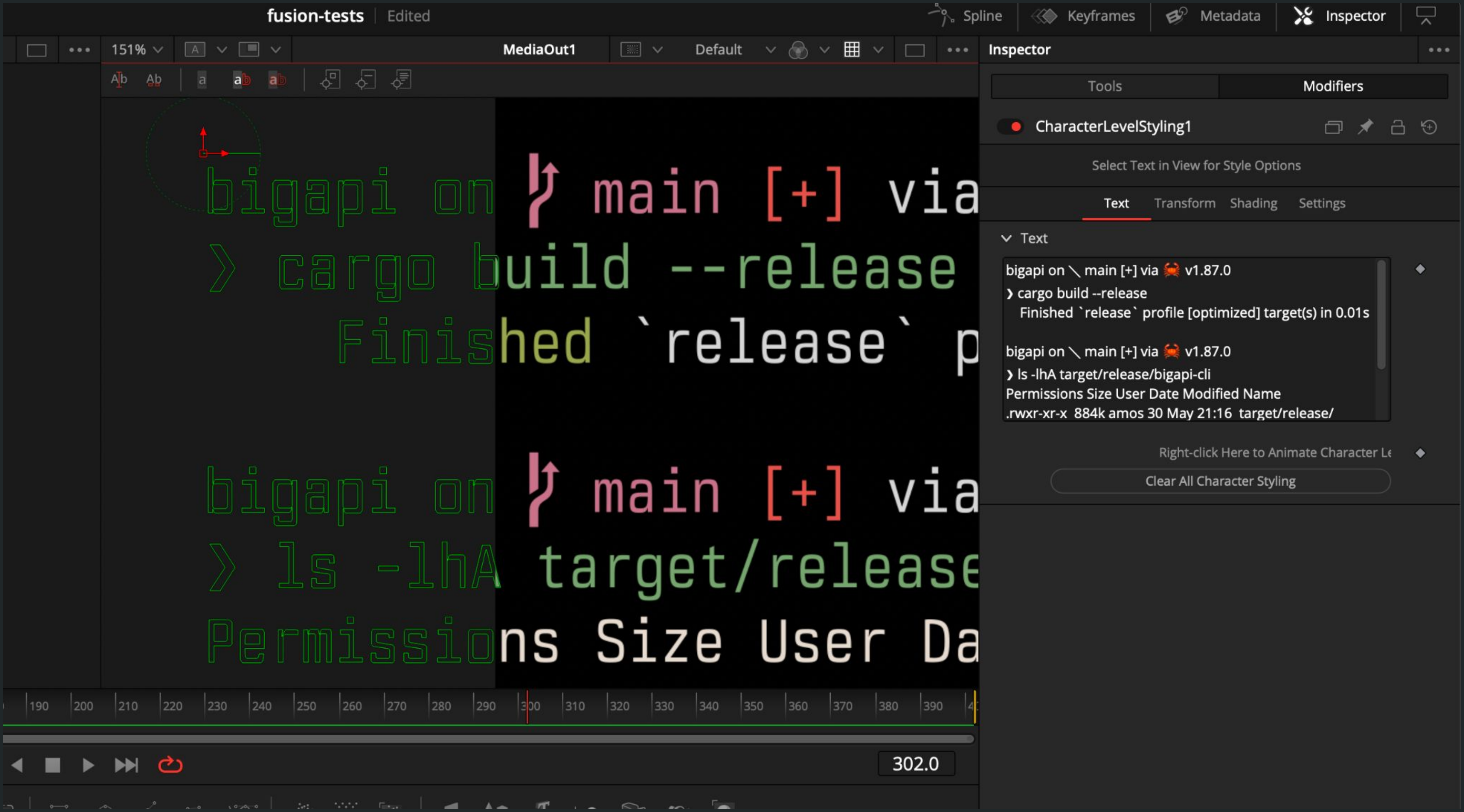
Font Size 0.027 Center X 0.055 Center Y 0.866 Copy Fusion Graph

```
bigapi on  main [+] via 🦀 v1.87.0
❯ cargo build --release
    Finished `release` profile [optimized] target(s) in 0.01s

bigapi on  main [+] via 🦀 v1.87.0
❯ ls -lhA target/release/bigapi-cli
Permissions Size User Date Modified Name
.rwxr-xr-x  884k amos 30 May 21:16  target/release/bigapi-cli
```

self-directed research

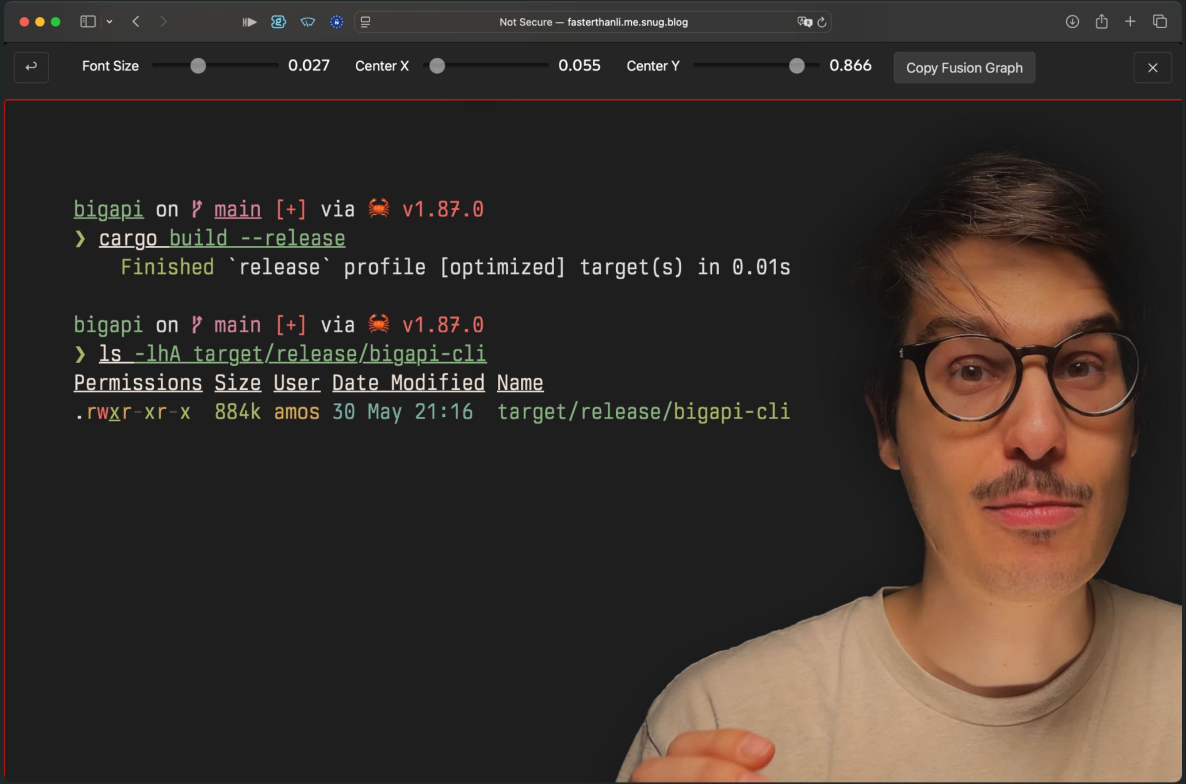


```
bigapi on  main [+] via 🦀 v1.87.0
❯ cargo build --release
   Finished `release` profile [optimized] target(s) in 0.01s

bigapi on  main [+] via 🦀 v1.87.0
❯ ls -lhA target/release/bigapi-cli
Permissions Size User Date Modified Name
.rwxr-xr-x  884k amos 30 May 21:16  target/release/bigapi-cli
```

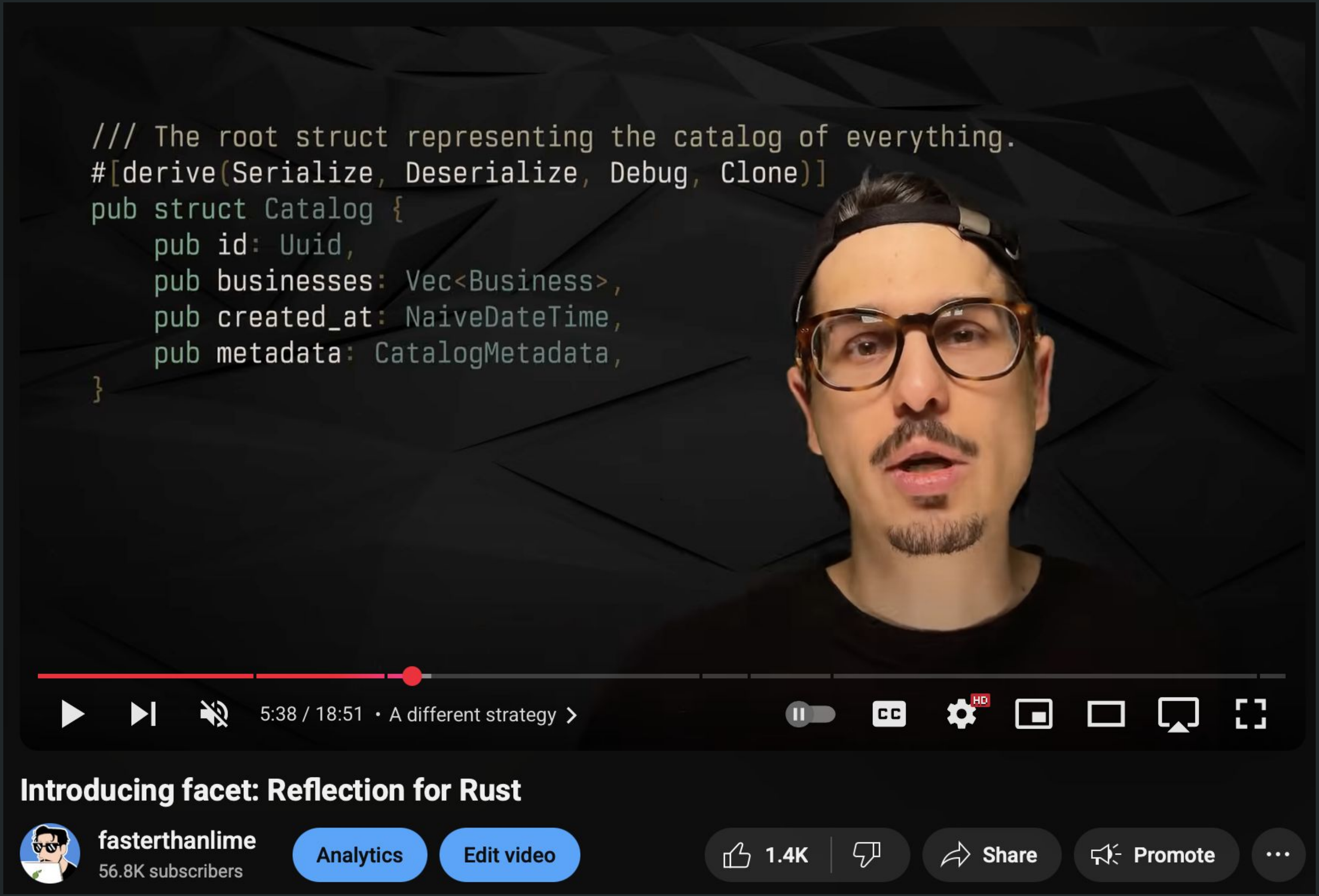


```
bigapi on  main [+] via 🦀 v1.87.0
❯ cargo build --release
  Finished `release` profile [optimized] target(s) in 0.01s

bigapi on  main [+] via 🦀 v1.87.0
❯ ls -lhA target/release/bigapi-cli
Permissions Size User Date Modified Name
.rwxr-xr-x  884k amos 30 May 21:16  target/release/
```

Other QoL stuff



```
bigapi on  main [+] via 🦀 v1.87.0
❯ cargo build --release
    Finished `release` profile [optimized] target(s) in 0.01s

bigapi on  main [+] via 🦀 v1.87.0
❯ ls -lhA target/release/bigapi-cli
Permissions Size User Date Modified Name
.rwxr-xr-x  884k amos 30 May 21:16  target/release/bigapi-cli
```



```rust
/// The root struct representing the catalog of everything.
#[derive(Serialize, Deserialize, Debug, Clone)]
pub struct Catalog {
    pub id: Uuid,
    pub businesses: Vec<Business>,
    pub created_at: NaiveDateTime,
    pub metadata: CatalogMetadata,
}
```

5:38 / 18:51 • A different strategy ›
CC
HD
Introducing facet: Reflection for Rust
fasterthanlime
56.8K subscribers
Analytics
Edit video
1.4K
Share
Promote
···