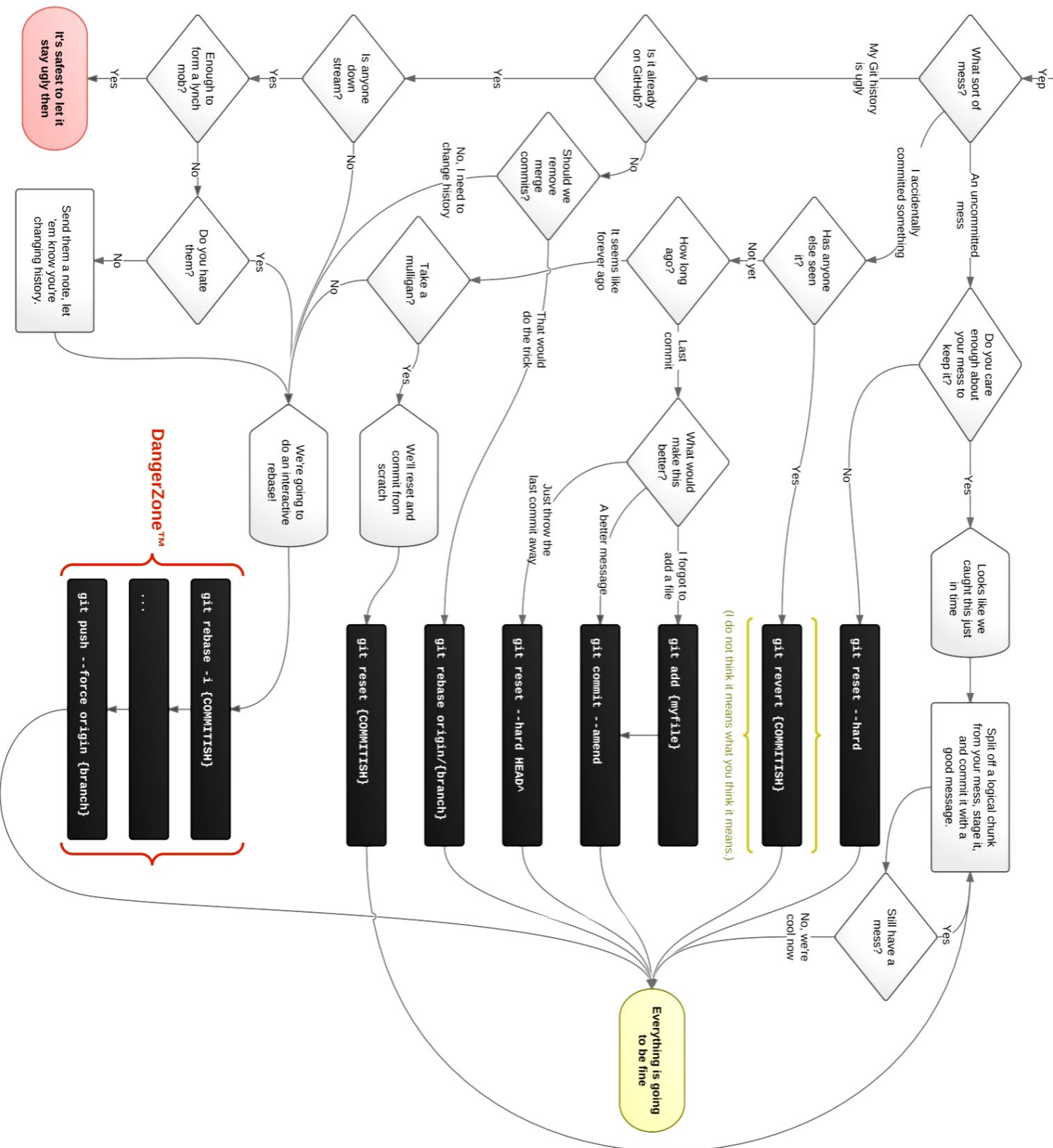


So you have a mess on your hands

justin hileman
<http://justinhileman.info/git-prety/>



[illegible][illegible]

Manipulation

Copying	
close()	
DOM Insertion, Around	
wrap()	..after()
wrapAll()	..before()
wrapInner()	..insertAfter()
	..insertBefore()
DOM Insertion, Inside	
append()	DOM Removal
appendTo()	..detach()
html()	..empty()
prepend()	..remove()
prependTo()	..unwrap()
text()	DOM Replacement
	..replaceAll()
	..replaceAllText()

```

10 Filtering
11 add()
12 filter()
13 first()
14 last()
15 lastIndexOf()
16 map()
17 sort()
18 slice()
19
20 Miscellaneous Traversing
21 add()
22 addClass()
23 animate()
24 animate()
25 attr()
26 attr()
27 attr()
28 attr()
29 attr()
30 attr()
31 attr()
32 attr()
33 attr()
34 attr()
35 attr()
36 attr()
37 attr()
38 attr()
39 attr()
40 attr()
41 attr()
42 attr()
43 attr()
44 attr()
45 attr()
46 attr()
47 attr()
48 attr()
49 attr()
50 attr()
51 attr()
52 attr()
53 attr()
54 attr()
55 attr()
56 attr()
57 attr()
58 attr()
59 attr()
60 attr()
61 attr()
62 attr()
63 attr()
64 attr()
65 attr()
66 attr()
67 attr()
68 attr()
69 attr()
70 attr()
71 attr()
72 attr()
73 attr()
74 attr()
75 attr()
76 attr()
77 attr()
78 attr()
79 attr()
80 attr()
81 attr()
82 attr()
83 attr()
84 attr()
85 attr()
86 attr()
87 attr()
88 attr()
89 attr()
90 attr()
91 attr()
92 attr()
93 attr()
94 attr()
95 attr()
96 attr()
97 attr()
98 attr()
99 attr()
100 attr()

```

Events

```
Browsers Events
blur()
error()
load()
mousemove()
mouseout()
mouseover()
scroll()

Document Loading
load()
ready()
readystatechange()

Event Handler Attachment
bind()
attachEvent()
detach()
dispatch()
listen()
off()
on()
removeEventListener()
useCapture()
addEventListener()
attachEvent()
detachEvent()

Form Events
change()
click()
dblclick()
focus()
keypress()
paste()
submit()
upload()
unload()
```

Effects

```

1 Basics
2 hide()
3 show()
4 toggle()
5
6 Custom
7 animate()
8 clearQueue()
9 delay()
10 dequeue()
11 jQuery.dequeue()
12 finish()
13 jQuery.fx.interval
14 jQuery.fx.off
15 jQuery.suspend
16 queue()
17 jQuery.queue()

```

AJAX

Global Ajax Event Handlers	Low-Level Interface
<code>ajaxComplete()</code>	<code>jQuery.ajax()</code>
<code>ajaxError()</code>	<code>jQuery.ajaxFilter()</code>
<code>ajaxStart()</code>	<code>jQuery.ajaxSetup()</code>
<code>ajaxStop()</code>	<code>jQuery.ajaxTransport()</code>
<code>ajaxSuccess()</code>	
Helper Functions	Shortcut Methods
<code>jQuery.param()</code>	<code>jQuery.get()</code>
<code>serialize()</code>	<code>jQuery.getJSON()</code>
<code>serializeArray()</code>	<code>jQuery.getScript()</code>
	<code>jQuery.get()</code>
	<code>load()</code>

Core

[illegible]

CANVAS

CANVAS IS A COMPLEX ELEMENT THAT BASICALLY

CONSISTS OF 2 PARTS:

HTML <canvas id="crvs" width="600" height="300"></canvas>

JS

```
var c = document.getElementById("crv");
var ctx = c.getContext("2d");
//check types of content.....;
```

REQUIRED ATTRIBUTES FOR <CANVAS>

id

Width

Height

TYPES OF CONTENT:

Failback content

```
<canvas id="crvs" width="600" height="300">Your
browser doesn't support canvas</canvas>
```

2D content

```
var ctx = canvas.getContext("2d");
```

3D content

```
var ctx = canvas.getContext("webgl");
```

SHAPES

RECTANGLE

To draw a rectangle:

```
rect(x, y, width, height)
fillRect(x, y, width, height)
strokeRect(x, y, width, height)
clearRect(x, y, width, height)
```

PATH

To draw a path:

```
beginPath()
Use Path methods
closePath()
stroke()/fill()
```

Path methods

```
moveTo()
lineTo()
bezierCurveTo()
quadraticCurveTo()
arc()
arcTo()
ellipse()
rect()
```

STYLES & COLORS

COLORS:

```
fillStyle = color
strokeStyle = color
```

Transparency:

```
globalAlpha = transparencyValue
```

Line styles:

```
lineWidth = value
lineCap = type
```



lineJoin = type



miterLimit = value

```
getLineDash()
setLineDash(segments)
lineDashOffset = value
```

GRADIENTS:

```
createLinearGradient(x1, y1, x2, y2)
createRadialGradient(x1, y1, r1, x2, y2, r2)
gradient.addColorStop(position, color)
```

PATTERNS:

```
createPattern(image, type)
```

SHADOWS:

```
shadowOffsetX = float
shadowOffsetY = float
shadowBlur = float
shadowColor = color
```

CANVAS FIL RULES:

```
Nonzero-rule
Even-odd rule
```

TEXT

DRAWING TEXT:

```
fillText(text, x, y [, maxWidth])
strokeText(text, x, y [, maxWidth])
```

STYLING TEXT:

```
font = value
textAlign = value
textBaseline = value
direction = value
```

ADVANCED TEXT MEASUREMENT

```
measureText()
```

IMAGES

CANVAS API CAN USE ANY OF THE FOLLOWING DATA

TYPES

```
HTMLImageElement
HTMLVideoElement
HTMLCanvasElement
```

GET AN IMAGE:

from the same page:
use another canvas element:

Create images from scratch:
Embedding an image via data: url

Using frames from a video <video></video>

DRAW AN IMAGE:

```
drawImage(image, x, y)
```

SCALE AN IMAGE:

```
drawImage(image, x, y, width, height)
```

SLICE AN IMAGE:

```
drawImage(image, sx, sy, sWidth, sHeight, dx, dy, dWidth,
dHeight)
```

CONTROL IMAGE SCALING BEHAVIOR

```
ctx.mozImageSmoothingEnabled = false;
ctx.webkitImageSmoothingEnabled = false;
ctx.msImageSmoothingEnabled = false;
ctx.imageSmoothingEnabled = false;
```

TRANSFORMATIONS

```
save()
restore()
translate(x, y)
rotate(angle)
scale(x, y)
transform(a, b, c, d, e, f)
setTransform(a, b, c, d, e, f)
resetTransform()
```

COMPOSITING AND CLIPPING

globalCompositeOperation = typeclip()



source-over



source-in

source-atop

destination-over

destination-in



copy

xor

destination-out



destination-atop

lighter

ANIMATION

BASIC ANIMATION STEPS:

Clear the canvas
Save the canvas state
Draw animated shapes
Restore the canvas state

SCHEDULE UPDATES:

```
setInterval(function, delay)
setTimeout(function, delay)
requestAnimationFrame(callback)
```

PIXEL MANIPULATION

```
createImageData()
getImageData()
putImageData()
drawImage()
```

Saving images

Creates a PNG image canvas.toDataURL('image/png')

Creates a JPG image canvas.toDataURL('image/jpeg', quality)

Creates a Blob object canvas.toBlob(callback, type, encodingsOptions)

HIT REGIONS AND ACCESSIBILITY

Hit regions (experimental)

```
CanvasRenderingContext2D.addHitRegion()
CanvasRenderingContext2D.removeHitRegion()
CanvasRenderingContext2D.clearHitRegions()
Focus ring (experimental):
CanvasRenderingContext2D.drawFocusIfNeeded()
CanvasRenderingContext2D.scrollPathIntoView()
```

USEFUL TIPS

Pre-render similar primitives or repeating objects on an off-screen canvas

Avoid floating-point coordinates and use integers

Don't scale images in drawImage

Use multiple layered canvases for complex scenes

CSS for large background images

Scaling canvas using CSS transforms

Use the moz-opacity attribute (Gecko only)

Batch canvas calls together

Avoid unnecessary canvas state changes.

Render screen differences only, not the whole new state.

Avoid the shadowBlur property whenever possible.

Try different ways to clear the canvas (clearRect() vs. fillRect() vs. resizing the canvas)

With animations, use window.requestAnimationFrame() instead of window.setInterval().

Be careful with heavy physics libraries

Source
https://developer.mozilla.org/en/docs/Web/API/Canvas_API/Tutorial
http://www.w3schools.com/tags/tryit.asp?filename=try_html5_canvas_api
http://www.html5canvastutorials.com/
https://www.airpoint.com/html5-canvas-tutorial-introduction/
https://www.kdowling.com/html5/html5-canvas.htm
http://davehax.com/html/canvas-element/
https://en.wikipedia.org/wiki/Canvas_element