

15コマ集中講義 ITブートキャンプ おまけ1

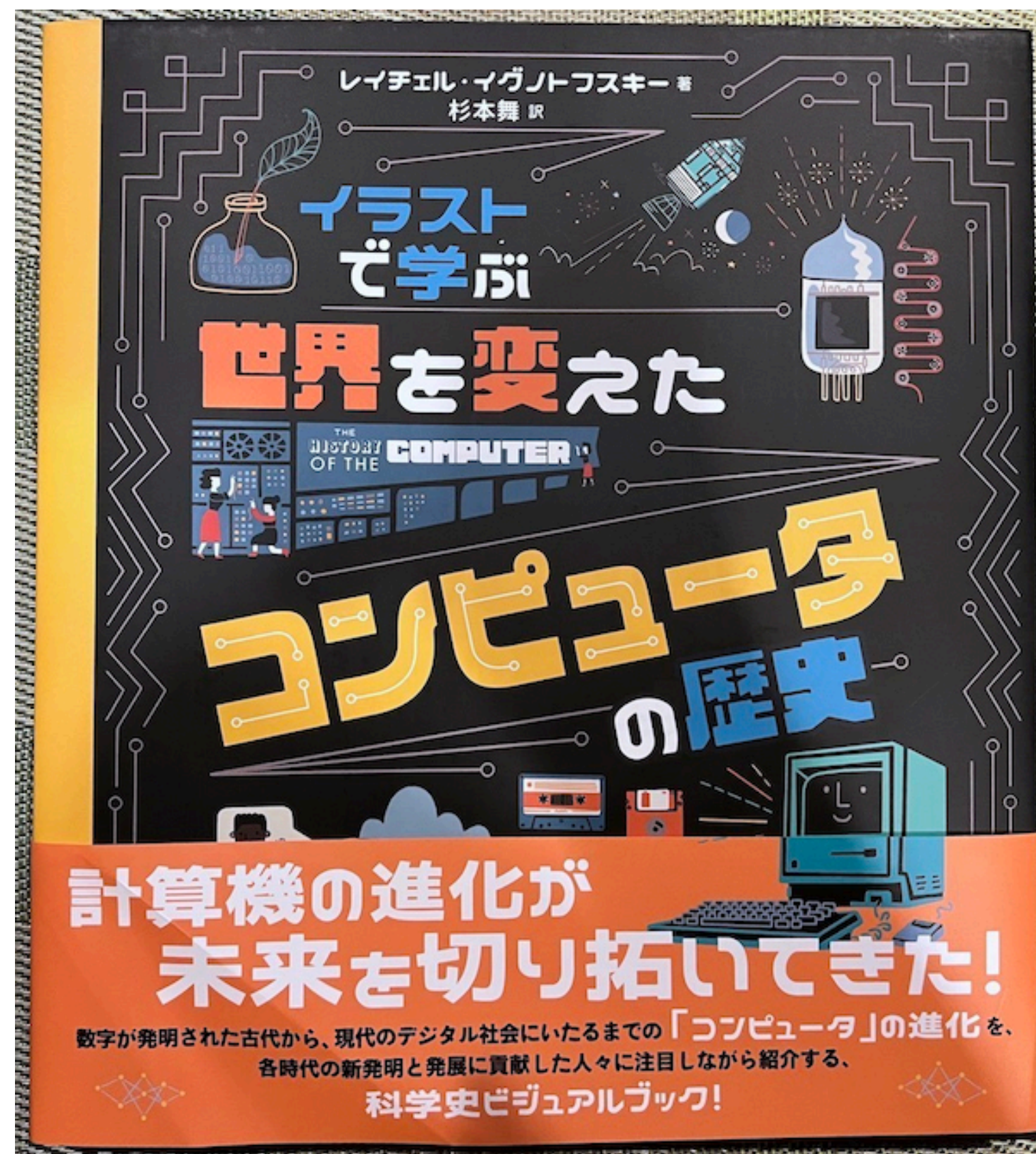
GPUアート



神山まると高専 技術教育統括ディレクター
福野泰介 @taisukef

一日一創

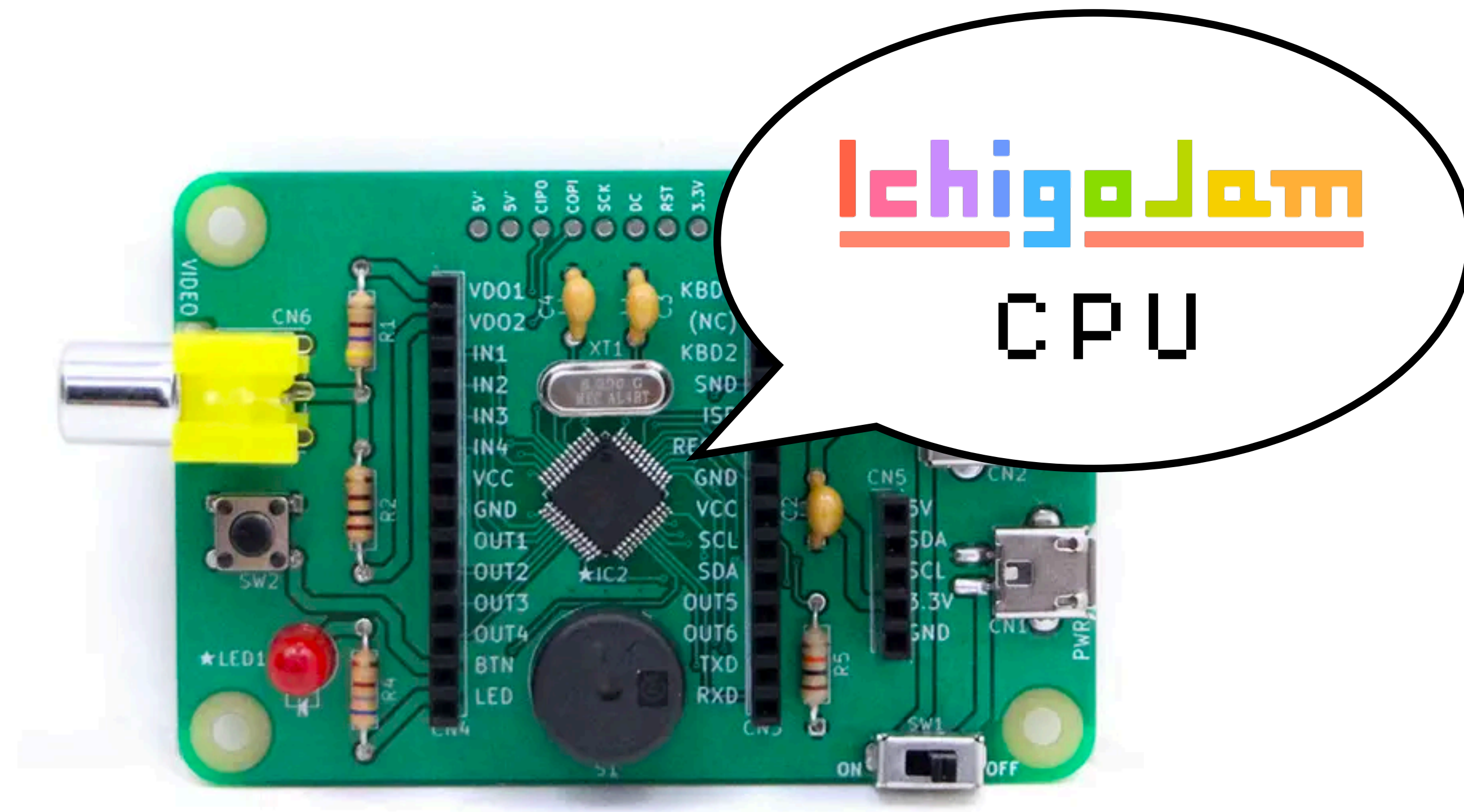
コンピューター＝計算する人



イラストで学ぶ
世界を変えたコンピュータの歴史



300円のコンピューター
1秒間に何回計算できる？



1 秒に1億回！



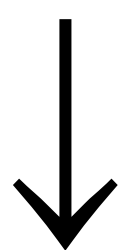
MacBook M1 Pro
1 秒間に何回計算できる？

MacBook M1 Pro

5.2兆回/秒

(5.2TFLOPS)

1 CPU
24億回/秒
2000倍違う？



GPU

【表1】 M1、M1 Pro、M1 Maxのスペック(?は筆者の推定)

| | M1 | M1 Pro | M1 Max |
|-----------|----------------|---------------|---------------|
| 高性能コア | 4コア | 8コア | 8コア |
| 高効率コア | 4コア | 2コア | 2コア |
| GPU | 8コア | 16コア | 32コア |
| 演算性能 | 2.6TFLOPS | 5.2TFLOPS | 10.4TFLOPS |
| ニューラルエンジン | 16コア | 16コア | 16コア |
| 最大メモリ容量 | 16GB | 32GB | 64GB |
| メモリ帯域幅 | 68.2GB/s ? | 200GB/s | 400GB/s |
| メモリ帯域幅 | 128bit | 256bit ? | 512bit ? |
| DRAM | LPDDR4x-4266 ? | LPDDR5-6400 ? | LPDDR5-6400 ? |
| トランジスタ数 | 160億 | 337億 | 570億 |
| 製造プロセスルール | 5nm | 5nm | 5nm |

GPUを活用したくなった？

GPUを活用したくなった？

→ C言語の親戚、GLSL

GLSL = OpenGL Shading Language

<https://ss.sabae.cc/#297>

ES-Jam ~Webプログラミング道場~

見る (^S)

SAVEする

LOADする

UPする

```
1 <script type="module" src="https://code4fukui.github.io/eggl/fragment
2 <fragment-shader hidden>#version 300 es
3 precision highp float;
4
5 uniform float time;
6 uniform vec2 mouse;
7 uniform vec2 resolution;
8
9 out vec4 outColor;
10
11 void main(void) {
12     // マウスカーソルの座標を(0.0,0.0)-(1.0,1.0)に変換した2次元ベクトル m を計算
13     vec2 m = vec2(mouse.x * 2.0 - 1.0, -mouse.y * 2.0 + 1.0);
14     // 画面内の点の座標を(0.0,0.0)-(1.0,1.0)に変換した2次元ベクトル p を計算
15     vec2 p = (gl_FragCoord.xy * 2.0 - resolution) / min(resolution.x, r
16     // マウスカーソルと画面内の点との距離 l を計算
17     float l = length(m - p);
18
19     float size = 0.2;
20     size = size + (sin(time * 10.0) + 1.0) * .1;
21     float t = size / l;
22     //t = floor(t);
23
24     outColor = vec4(t, 0.0, 0.0, 1.0);
```



23

24

25

```
outColor = vec4(t, 0.0, 0.0, 1.0);
```



赤



緑



青



不透明度

```
outColor = vec4(0.0, 0.0, t, 1.0); // 青色
```

```
outColor = vec4(t, t, 0.0, 1.0); // 黄色
```

```
outColor = vec4(t, t, t, 1.0); // 白色
```

```
outColor = vec4(t * 0.2, 0.0, t, 1.0); // ??
```


C言語、GLSL、JavaScriptなどで使えるコメントだよ

// 一行コメント

/*

複数行コメント

*/

```
18
19 float size = 0.2;
20 size = size + (sin(time * 10.0) + 1.0) * .1;
21 float t = size / l;
22 //t = floor(t);
23
```



スラッシュスラッシュで始まる行は実行しない (コメント)

```
18
19 float size = 0.2;
20 size = size + (sin(time * 10.0) + 1.0) * .1;
21 float t = size / l;
22 t = floor(t);
23
```



トル

スマホでも動くよ
ウェブサイトの背景にもできるよ
レッツ、GPUアート！

<https://ss.sabae.cc/#502>

ES-Jam ~Webプログラミング道場~

ss.sabae.cc/#502

ES-Jam ~Webプログラミング道場~

見る (^S) SAVEする LOADする UPする

```
1 <!DOCTYPE html><html><head><meta charset="utf-8"><meta name="viewport" content="width=device-width, height=device-height"></head><body><script type="module" src="https://code4fukui.github.io/eggl/fragment-shader.js"></script><fragment-shader hidden>#version 300 es
2
3 precision highp float;
4
5
6 uniform float time;
7 uniform vec2 mouse;
8 uniform vec2 resolution;
9
10 out vec4 outColor;
11
12 void main(void) {
13     // マウスの座標を(0.0,0.0)-(1.0,1.0)に変換した2次元ベクトル m を計算
14     //vec2 m = vec2(mouse.x * 2.0 - 1.0, -mouse.y * 2.0 + 1.0);
15     // 自動的に回転させる2次元ベクトル m を計算
16     vec2 m = vec2(cos(time) * 0.5, sin(time) * 0.5);
17
18     // 画面内の点の座標を(0.0,0.0)-(1.0,1.0)に変換した2次元ベクトル p を計算
19     vec2 p = (gl_FragCoord.xy * 2.0 - resolution) / min(resolution.x, resolution.y);
20     // マウスの座標と画面内の点との距離 l を計算
21     vec2 d = m - p;
22     float th = atan(d.y, d.x);
23     float l = length(m - p) + .5 + sin(th * 3.0 + time * 5.0) + cos(th * 7.0) * 0.5;
24
25     float size = 0.2;
26     size = size + (sin(time * 1.0) + 5.0) * .1;
27     float t = size / l;
28     t = floor(t);
29
30     if (mod(l, 0.1) < 0.05) {
31         outColor = vec4(1.0 - t, 0.0, 0.0, .1);
32     } else {
33         outColor = vec4(t, sin(t), t * .5, 0.3);
34     }
35 }
```

GSLSアート

HTMLを拡張するHTMLElementという仕様を使って、GPUを活用する言語GLSLを直接書ける拡張タグ「fragment-shader」タグを作りました。

fragment-shaderタグのCSSに、position: fixed; width: 100vw; height: 100vh; top: 0; z-index: -1; を設定すると背景にできます。

[src on GitHub](#)