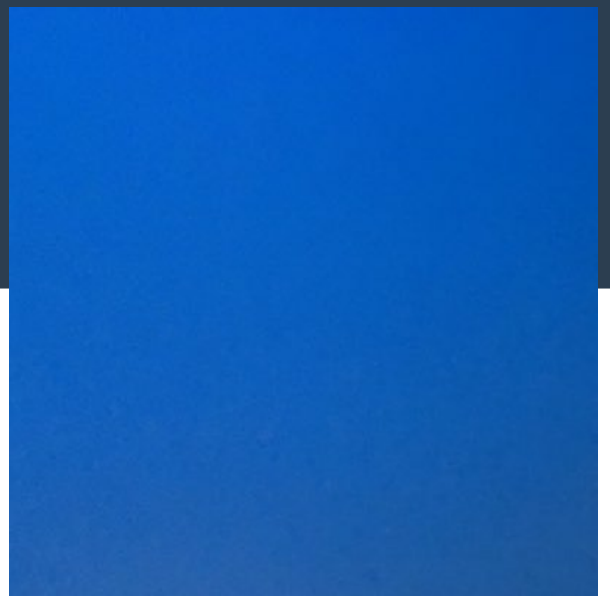
Kokkai
Speed Battle

sei0o @ #nitaclt 第 9 回

誰？



- sei0o inoue (@sei0o)
- 明石高専 2E
- CTF とかなんかいろいろ広く浅く

ネタに困る

- 広く浅くなのでネタがたくさんありすぎる
- 「1 年生でも楽しめる」の壁

ネタに困る

- LLVM, コンパイラ最適化
- Bluetooth パケットを読む
- CSS で論理回路
- iOS アプリの中身を見る
  - 自己満足 && 時間足りない

ネタに困る

- 環境構築
  - Git(Hub), Linux, editor etc…
  - 戦争
  - 他の人と被りそう

ネタに困る

- 「もっと、こう、わかりやすいの」
- そうだ!

ん?


日本共産党
Japanese Communist Party
お問い合わせ
入党の申し込み
文字サイズ 小 中 大
ENGLISH PAGE
キーワードを入力して検索
政策
議員
党紹介
ダウンロード
エントリー
安倍９条改憲 NO!
３千万人署名に
あなたも。
憲法を生かす全国統一署名
LOVE 9条
あなたも
3千万人
署名を。
共産党のキホンのキホン
日本共産党の
基本情報

政治的主張

- 左翼の Web サイトは読み込みが速そう
  - 左翼はプロパガンダが得意
- いっぱい集めて速度を比べてみたい
  - プログラミングの出番

Kokkai Speed Battle

- 衆議院に議席がある政党
- 自宅 PC から測定
  - VPS からではダメだった（メモリ不足？）
- Node から puppeteer 経由で Headless Chrome を操作
- Navigation Timing APIで計測
- スクレイピングに近い感覚

雑な説明

- Headless Chrome
  - 顔(画面)がない Chrome
  - キーボードの代わりにプログラムで操作
    - 今回は JavaScript を使用
- Node
  - JavaScript をブラウザの外でも使えるように
- puppeteer
  - Node から Headless Chrome を利用するためのライブラリ

雑な説明

- API: Application Programming Interface
  - プログラムから他のプログラムの機能を使えるように作られた橋
- ライブラリ
  - だいたい似たようなもん
  - #include <stdio.h> すると、printf が使える

もうすこし「双方向」「共通語」の意味合いが強くなると「プロトコル」と呼ばれます

Navigation Timing API

- Web サイトの読み込みにかかる時間を測ってくれる API
- 自分の書いたプログラムからブラウザ (Headless Chrome) の機能を使う

コードを見てみよう

- 百聞は一見にしかず
- Gistに置いてあります

コード

```
const puppeteer = require('puppeteer')
const URLs = [
  'https://www.jcp.or.jp/'
  'https://kibounotou.jp/', …
];
```

コード

```
async function measureTime (browser, url)  {
  let results = []
  const page = await browser.newPage()
  for (let i = 0; i < 10; i++) {
    await page.goto(url, ...)
    results.push(await page.evaluate(() => {
      const tim = window.performance.timing
      return tim.loadEventEnd - tim.navigationStart
    }))
  }
  await page.close()
  return results.reduce((prev, cur) => prev + cur) / 10.0
}
```

コード

```
(async () => {
  const browser = await puppeteer.launch(...)
  const results = await Promise.all(
    URLs.map(url => measureTime(browser, url))
  )
  console.log(results)
  await browser.close()
})()
```

実行

```
$ node index.js
[ [ 'https://cdp-japan.jp/', 2329.2 ],
  [ 'https://kibounotou.jp/', 2646.8 ],
  [ 'http://www.liberalparty.jp/', 2713.5 ],
  [ 'https://www.jimin.jp/', 3583.5 ],
  [ 'https://www.jcp.or.jp/', 3599.3 ],
  [ 'https://www.komei.or.jp/', 3830.1 ],
  [ 'https://o-ishin.jp/', 4031.7 ],
  [ 'http://www5.sdp.or.jp/', 5344 ] ]
```

測定結果（生データ）

政党	load 10回平均	load
立憲民主党（中道左派）	2329.2ms	13774ms
希望の党（?）	2646.8ms	30565ms
自由党（?）	2713.5ms	13895ms
自由民主党（中道右派）	3583.5ms	35684ms
日本共産党（左派）	3599.3ms	18083ms
公明党（中道）	3830.1ms	33801ms
日本維新の会（?）	4031.7ms	30431ms
社会民主党（中道左派）	5344.0ms	29584ms
阿部寛のホームページ	54.9ms	293ms

キャッシュが効くので２回目以降はかなり速くなる

まとめ

- 左翼ではなく、新しい政党の Web サイトが速い
- Headless Chrome かんたん

- 世界中の政党ではどうなるか？
- スマホ版では？
- 立憲民主党と社民党サイトの差はどこから？
  - LT ネタにどうぞ