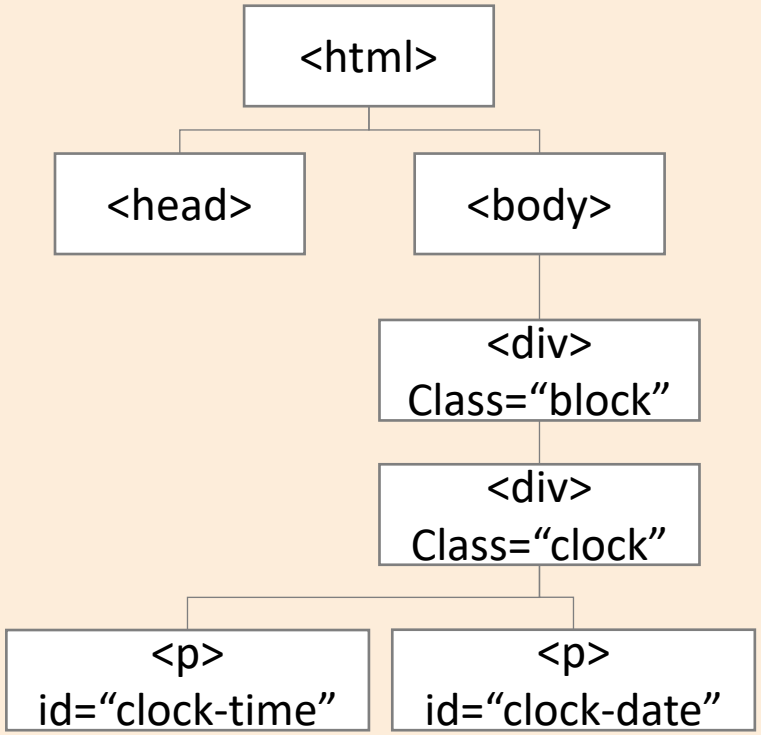
デジタル時計

リアルタイムが表示されるwebページ

プログラミングの構造

index.html
時計のタグを用意



Script.js
時計の機能を実装

関数 clock　現在の日時取得

関数 zero　1桁の場合0をつける
例：1:25→01:25

getFullYear()
getMonth()
getDate()…　情報取得

Document.getElementById("")
.innerText
HTMLに文字列を出力

setInterval(clock, 1000)
1秒ごとにclock関数を呼び出す

style.css
時計の見た目を装飾

body　Background-color:black;

.block　全体を囲っているブロック
・左右上下中央配置
・ウィンドウサイズに合わせた大きさ

.clock　時計のスタイル

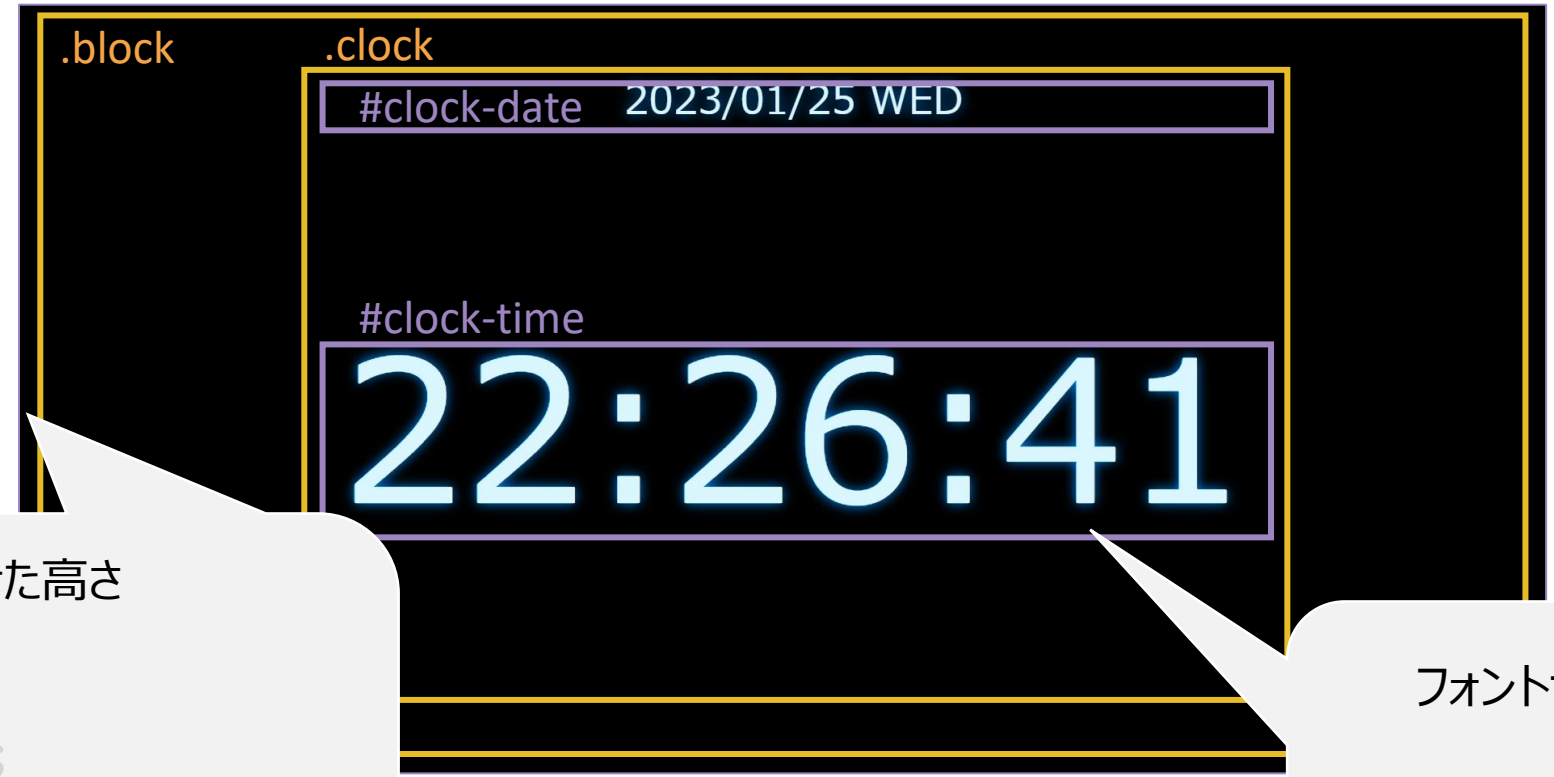
#clock-date　日付のスタイル

#clock-time　時間のスタイル

工夫したところ

工夫１：見栄えの良さ

画面サイズが変化しても配置やサイズが大きく変化しないようにブラウザウィンドウサイズを基準。



表示領域に合わせた高さ

```
width: 100vw;
height: 100vh;
```

Vw：ブラウザウィンドウの横幅の割合
Vh：ブラウザウィンドウの高さの割合

フォントサイズの単位：vmin

・・・ブラウザウィンドウの縦横サイズで小さいほうを基準とする単位

工夫したところ

工夫2：処理時間の考慮

時間を表示するプログラムのため、処理時間を計測して表示に影響があるかを確認。

⇒0.0~0.1の誤差であったため未対応。（NICTのページで目視でも確認）

国立研究開発法人 情報通信研究機構 National Institute of Information and Communications Technology

https://www.nict.go.jp/JST/JST5.html

```
const startTime = performance.now(); // 開始時間(処理時間計測用)

 実際の処理

const endTime = performance.now(); // 終了時間(処理時間計測用)
console.log(endTime - startTime); // 何ミリ秒かかったかを表示する
```

メソッド	精度
Date.now()	1ミリ秒
Performance.now()	1/1000ミリ秒

※課題：ブラウザ処理速度はユーザーによって違うため実際は考慮が必要。

関連URL

- Webページ
  - https://mo30ka.github.io/digitalClock/

- ソースコード
  - https://codepen.io/mo30ka/pen/poZVvdm