

GS1標準バーコードの処理を簡単に！

GS1 Barcode Syntax Resource

GS1標準バーコードやGS1アプリケーション識別子（AI）を適切に処理するのは面倒...

GS1本部提供の GS1 Barcode Syntax Resourceが代行します！

GS1 element stringシンタックスも、GS1 Digital Link URIシンタックスも、これひとつ！

- GS1 Barcode Syntax Dictionary
GS1アプリケーション識別子の機械可読な定義
- GS1 Barcode Syntax Tests
GS1標準のデータに対するリンター
- GS1 Barcode Syntax Engine
上記二点を組み合わせ、GS1標準データの解釈を行うソフトウェアライブラリ



詳細はQRコードから！
本部紹介ページへリンクします（英文）

```
import { GS1encoder } from './gs1-syntax-engine/gs1encoder.mjs';
```

GS1 Barcode Syntax Engineの読み込み

元はC言語。このデモではWebAssemblyビルドを使用。
.NET/JVM/Android/iOSなど、他の環境でも使えます！

```
gs1encoder=new GS1encoder();
gs1encoder.init();
```

GS1 Barcode Syntax Engineの初期化

```
if(scannerText.startsWith("[E0") || scannerText.startsWith("[E4")){
    var gtin=scannerText.substring(3);
    while(gtin.length<14){
        gtin="0"+gtin;
    }
    aiData["01"]=gtin;
}else{
    gs1encoder.scanData=scannerText;
    var parsedHRIArray=gs1encoder.hri;
    parsedHRIArray.forEach(elm=>aiData[elm.split(" ")[0].substring(1)]=elm.split(" ")[1]);
}
```

GS1標準シンボルの読み込み結果の処理

プレーンシンタックスのシンボルについてはGTINを切り出すだけ。
GS1アプリケーション識別子の解析はGS1 Barcode Syntax Engineに丸投げ！