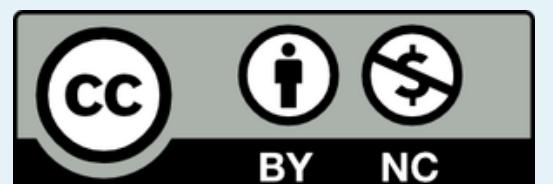


メディアアート・プログラミング2

東京藝術大学 芸術情報センター開設科目 後期金曜4限 第5週

2023.10.31 松浦知也 (matsura.tomoya@noc.geidai.ac.jp teach@matsuuratomo.ya.com)



配列としてのテキストデータ

青空文庫

メイン お知らせ 別館 資料 運営
www.aozora.gr.jp 内を検索

インターネットの電子図書館、青空文庫へようこそ。

[「青空文庫、充電中」](#)

初めての方はまず「[青空文庫早わかり](#)」をご覧ください。

ファイル利用をお考えの方は、[こちら](#)をご一読ください。

[「青空文庫収録ファイルを用いた朗読配信をお考えのみなさまへ」](#)

メインエリア	
青空文庫早わかり	青空文庫の使い方と約束事を紹介しています。初めての方、ファイルやキャプチャーの取り扱いについて知りたい方も、こちらへどうぞ。
総合インデックス	作家名、作品名の50音別に、公開作品と入力・校正作業中の作品を一覧できるインデックスです。公開中の作品を探すときは、下の近道もご利用ください。
公開中 作家別：	あ行 か行 さ行 た行 な行 は行 ま行 や行 ら行 わ行 他
公開中 作品別：	あ か さ た な は ま や ら わ い き し ち に ひ み り を う く す つ ぬ ふ む ゆ る を え け せ て ね へ め ゆ れ ん お こ そ と の ほ も よ ろ 他

<https://www.aozora.gr.jp/>

Project Gutenberg

Project Gutenberg

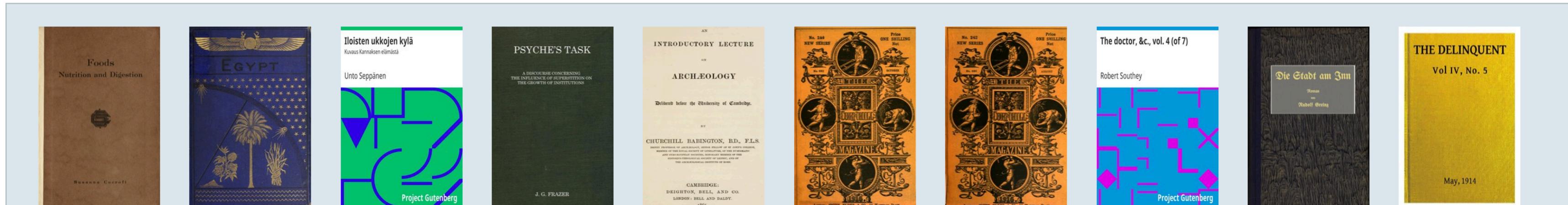
About ▾ Search and Browse ▾ Help ▾

Quick search **Go!** [Donation](#) [PayPal](#)

Welcome to Project Gutenberg

Project Gutenberg is a library of over 70,000 free eBooks

Choose among free epub and Kindle eBooks, download them or read them online. You will find the world's great literature here, with focus on older works for which U.S. copyright has expired. Thousands of volunteers digitized and diligently proofread the eBooks, for you to enjoy.



Some of our latest eBooks [Click Here for more latest books!](#)

Trouble downloading? In late September 2023 we began getting reports from people using Google Chrome who could no longer download EPUB or other formats. This seems to be due to a change in Chrome. Instead, try right-clicking then use the "Save as..." or similar menu to save to your computer. Or, try using a different web browser.

<https://www.gutenberg.org/>

●[トップ](#) ●[インデックス／全](#) ●[作家別作品リスト](#) ●[図書カード](#)

図書カード：No.49866

作品名： 変身
作品名読み： へんしん
原題： DIE VERWANDLUNG
著者名： [カフカ フランツ](#)

[[ファイルのダウンロード](#) | [いますぐ XHTML版で読む](#)]

作品データ

分類： NDC 943

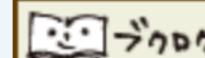
作品について： [「変身\(カフカ\)」](#)

文字遣い種別： 新字新仮名

備考： この作品には、今日からみれば、不適切と受け取られる可能性のある表現がみられます。その旨をここに記載した上で、そのままの形で作品を公開します。（青空文庫）

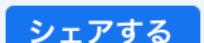
 [「青空 in Browsersで縦書き表示。PC、スマホ、タブレット対応」](#)

 [「えあ草紙・青空図書館」](#) 「スマホ＆タブレットで動く青空文庫縦書き対応汎用電子書籍リーダー」

 [「ブクログでレビューを読む、書く。」](#)

 [この作品の朗読](#) を Google で検索する

 [ツイート](#) この作品を twitter でつぶやく

 [いいね！](#)  [シェアする](#) 23人が「いいね！」しました。Facebookに登録して、友達の「いいね！」を見てみましょう。

<https://www.aozora.gr.jp/cards/001235/card49866.html>

工作員データ

入力 : kompass

校正 : 青空文庫

ファイルのダウンロード

ファイル種別	圧縮	ファイル名（リンク）	文字集合／符号化方式	サイズ	初登録日	最終更新日
テキストファイル(ルビあり)	zip	49866_ruby_41853.zip	JIS X 0208／ShiftJIS	47691	2011-01-01	2016-02-22
XHTMLファイル	なし	49866_41897.html	JIS X 0208／ShiftJIS	119433	2011-01-01	2016-02-22

● [ファイルのダウンロード方法・解凍方法](#)

関連サイトデータ



●作家リスト：公開中 [\[あ\]](#) [\[か\]](#) [\[さ\]](#) [\[た\]](#) [\[な\]](#) [\[は\]](#) [\[ま\]](#) [\[や\]](#) [\[ら\]](#) [\[わ\]](#) [\[他\]](#)

●作家リスト：全 [\[あ\]](#) [\[か\]](#) [\[さ\]](#) [\[た\]](#) [\[な\]](#) [\[は\]](#) [\[ま\]](#) [\[や\]](#) [\[ら\]](#) [\[わ\]](#) [\[他\]](#)

● [トップ](#) ● [インデックス／全](#) ● [作家別作品リスト](#)

```
henshin.txt — map2_text
```

エクスプローラー ...

MAP2_TEXT

henshin.txt

≡ henshin.txt ×

≡ henshin.txt

1 ♀Bg

2 DIE VERWANDLUNG

3 ♀t♀?????c♀E♀J♀t♀J♀@Franz Kafka

4 ♀??c♀`♀l♀?

5

6 -----

7 ♀y♀e♀L♀X♀g♀R♀L♀-♀az

8

9 ♀s♀t♀F♀r

10 ♀i♀j♀I♀s♀t

11

12 ♀b♀F♀r♀t♀n♀.♀欣♀L♀|

13 ♀i♀j♀C♀b♀E♀s♀ät

14

15 ♀m♀n♀F♀F♀@♀Q♀A♀T♀_♀3♀w♀

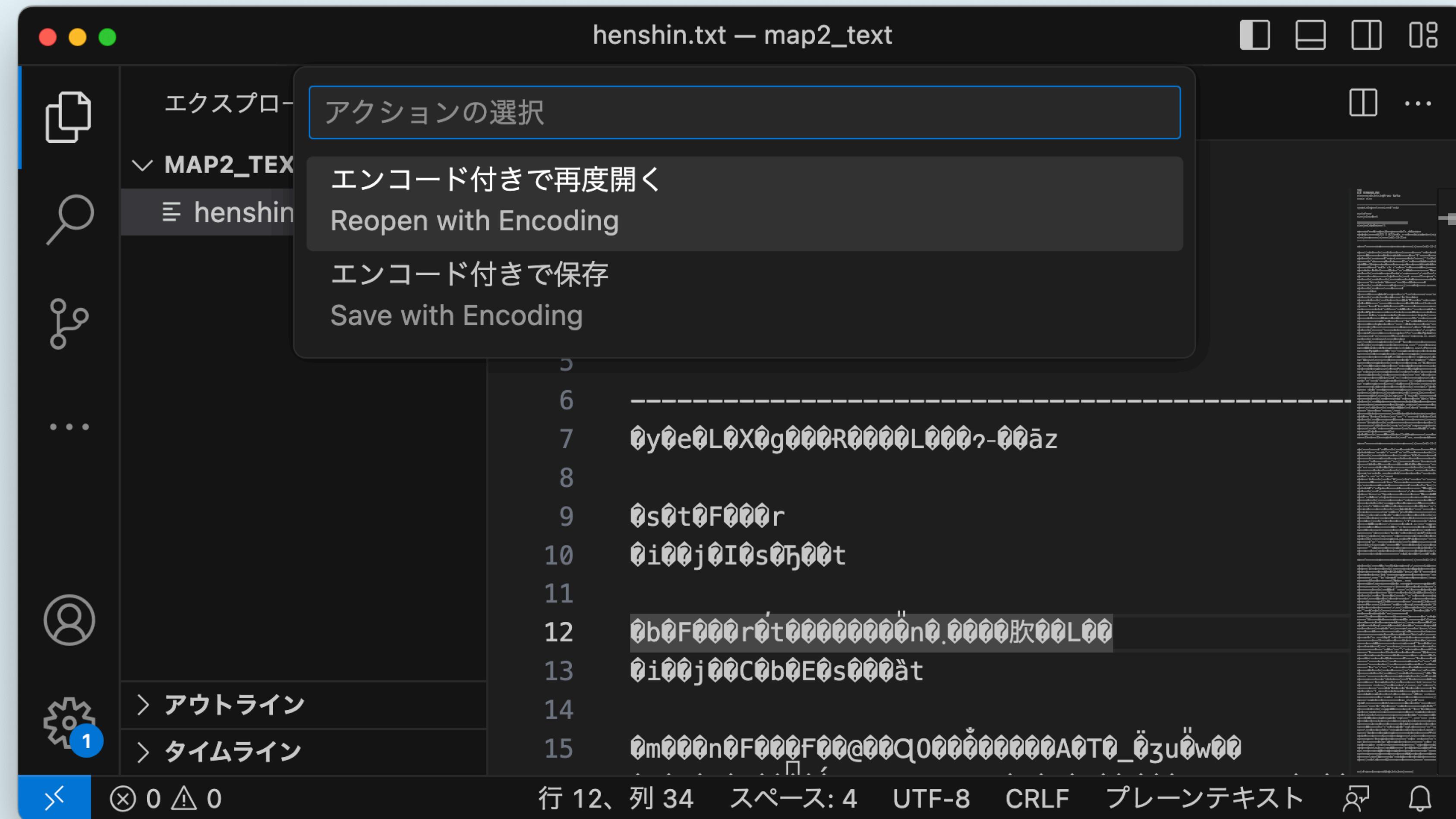
適当なフォルダに入れてVSCodeで開く

TextEdit window showing the file 'henshin.txt' with the following content:

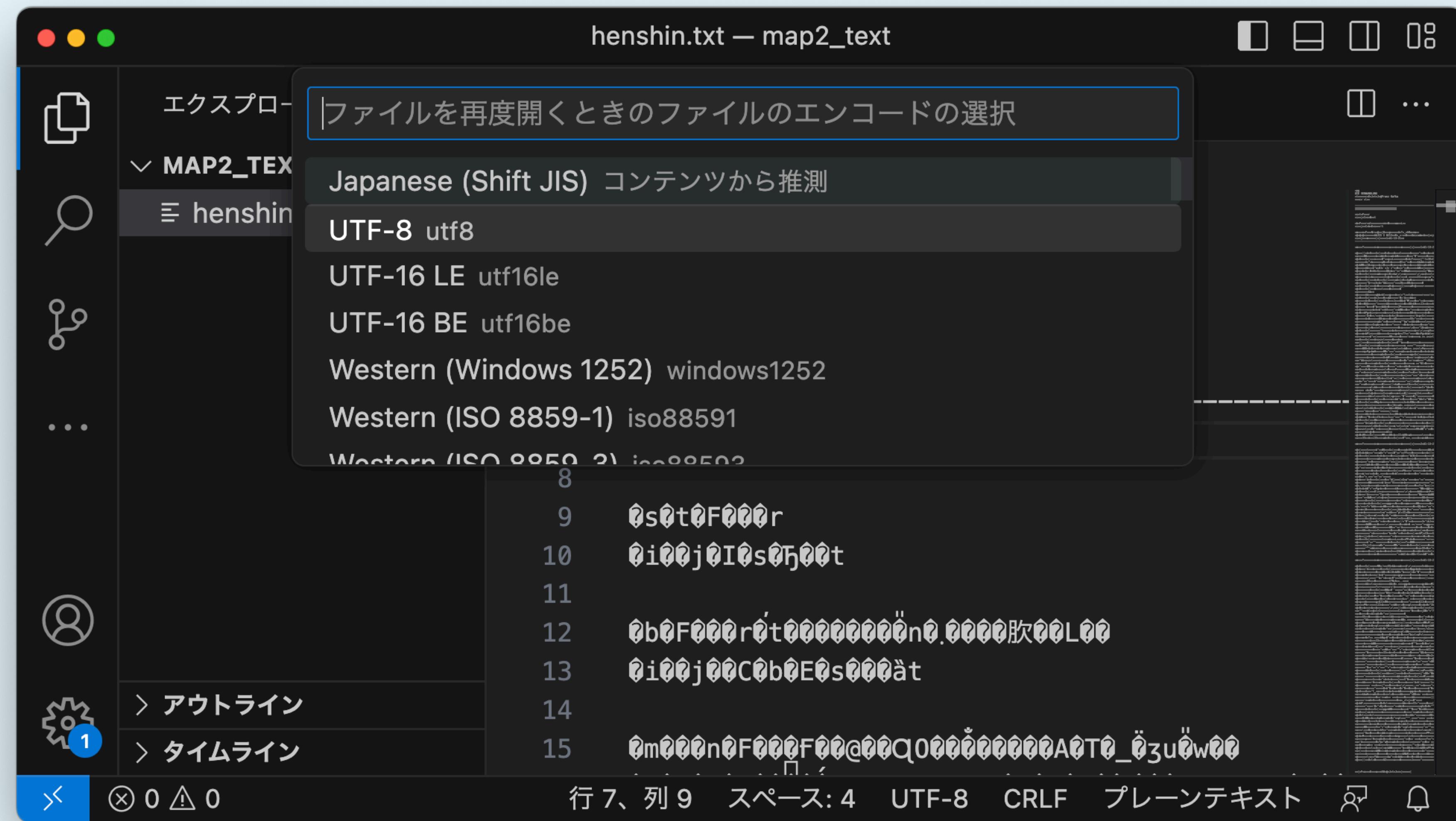
```
1  ♀Bg
2  DIE VERWANDLUNG
3  ♀t♀@♀@♀c♀E♀J♀t♀J♀@Franz Kafka
4  ♀@♀c♀`♀l♀@
5
6 -----
7  ♀y♀e♀L♀X♀g♀R♀L♀-♀az
8
9  ♀s♀t♀F♀r
10 ♀i♀j♀I♀s♀h♀t
11
12 ♀b♀F♀r♀t♀n♀.♀欣♀L♀
13 ♀i♀j♀C♀b♀E♀s♀at
14
15 ♀m♀n♀F♀F♀@♀Q♀A♀T♀_♀3♀w♀
```

The status bar at the bottom indicates the encoding is UTF-8.

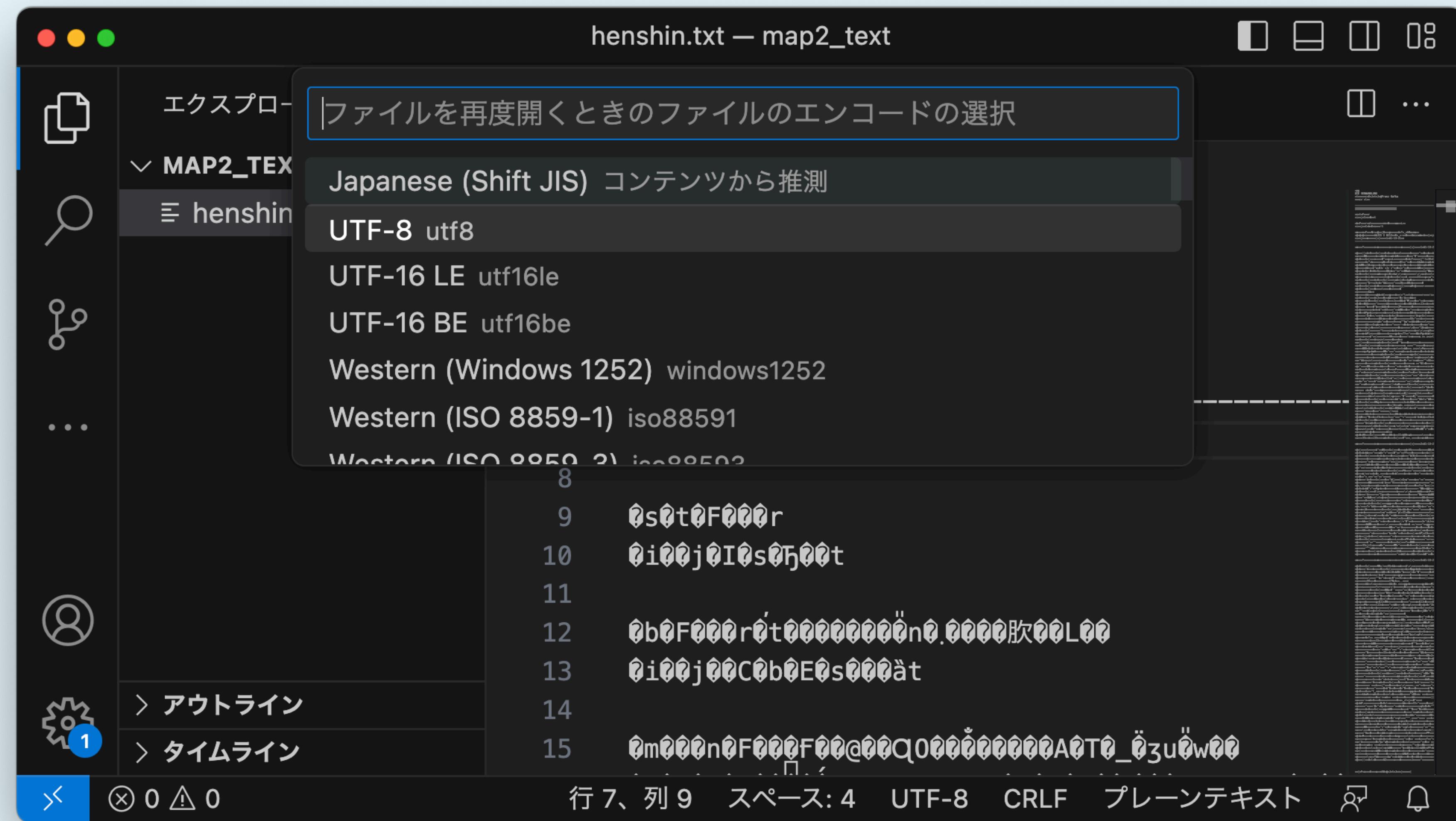
青空文庫は文字コードがShift-JISなので一度UTF-8に変換する



「エンコード付きで再度開く」を選択



Shift-JISを推測してくれるので選択



Shift-JISを推測してくれるので選択

henshin.txt — map2_text

エクスプローラー ...

MAP2_TEXT

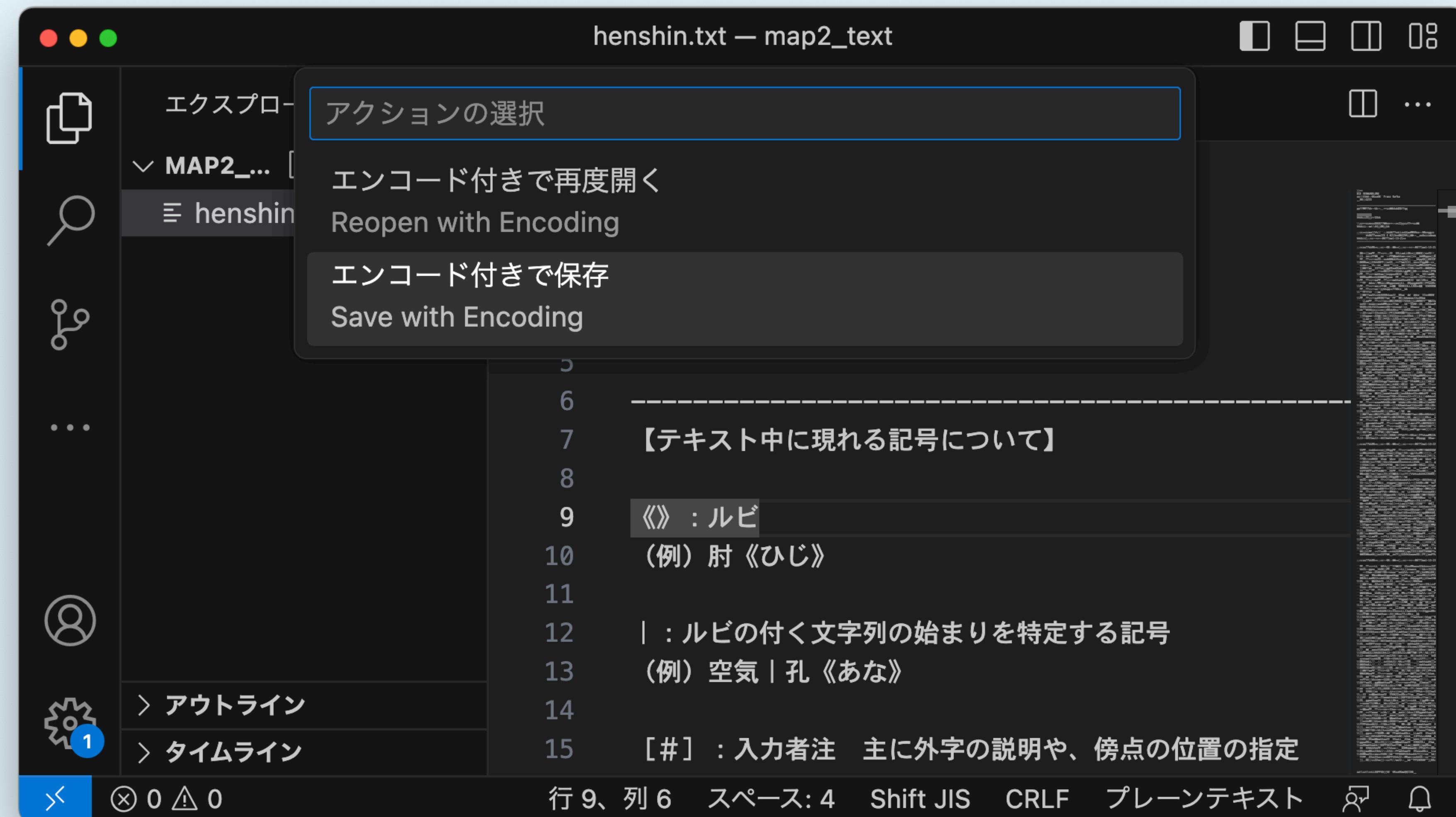
henshin.txt

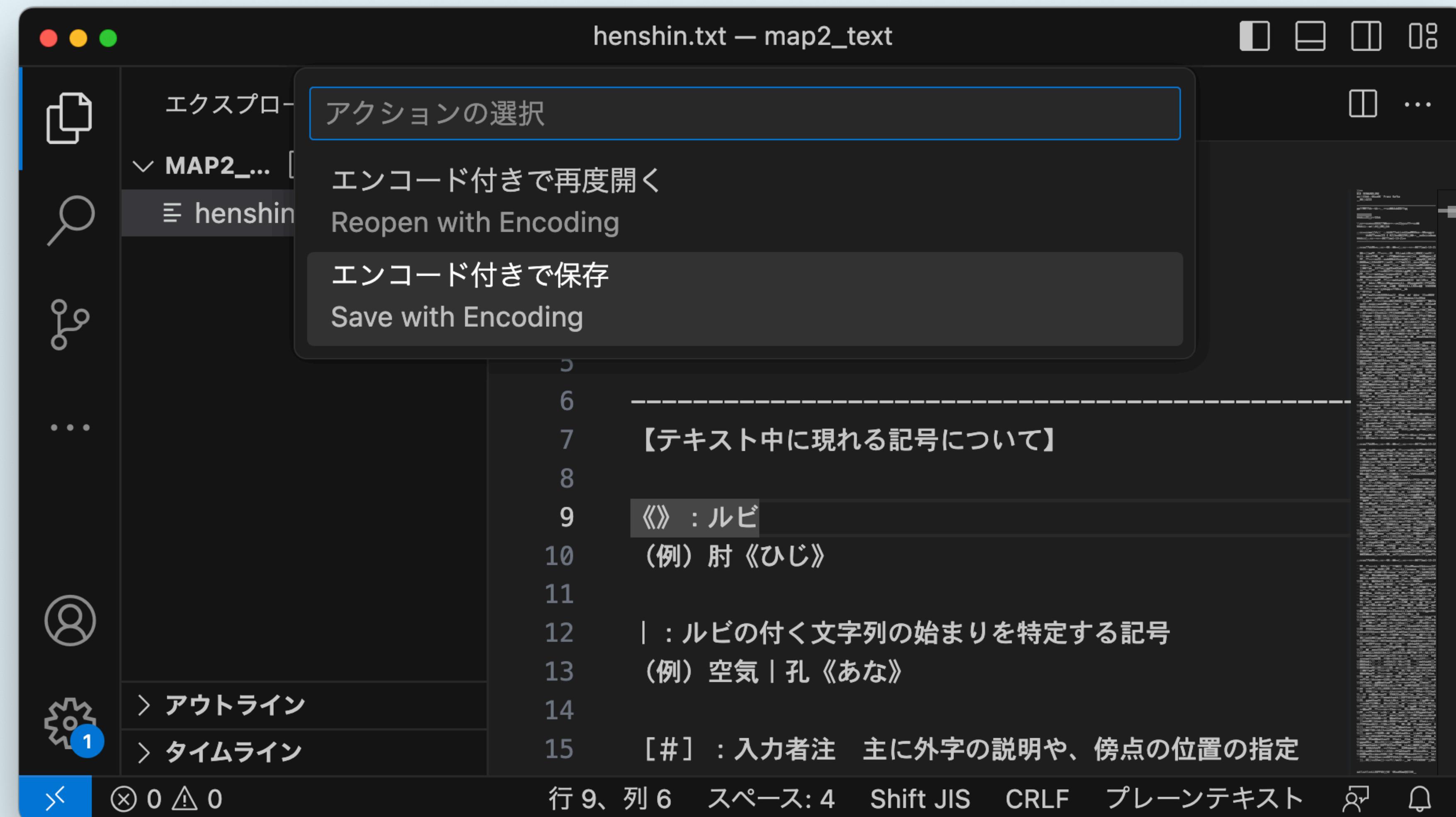
henshin.txt

1 变身
2 DIE VERWANDLUNG
3 フランツ・カ夫カ Franz Kafka
4 原田義人訳
5
6 -----
7 【テキスト中に現れる記号について】
8
9 《》 : ルビ
10 (例) 肘《ひじ》
11
12 | : ルビの付く文字列の始まりを特定する記号
13 (例) 空気 | 孔《あな》
14
15 [#] : 入力者注 主に外字の説明や、傍点の位置の指定

行 204、列 1 スペース: 4 Shift JIS CRLF プレーンテキスト ⌂ ⌂

開けた。今度はこれをUTF-8に変換して保存する





henshin.txt — map2_text

エクスプローラー

MAP2_TEX

henshin

...
...

保存時のファイルのエンコードの選択

Japanese (Shift JIS) コンテンツから推測

UTF-8 utf8

UTF-8 with BOM utf8bom

UTF-16 LE utf16le

UTF-16 BE utf16be

Western (Windows 1252) windows1252

Western (ISO 8859-1) iso8859-1

8

9 《》 : ルビ

10 (例) 肘《ひじ》

11

12 | : ルビの付く文字列の始まりを特定する記号

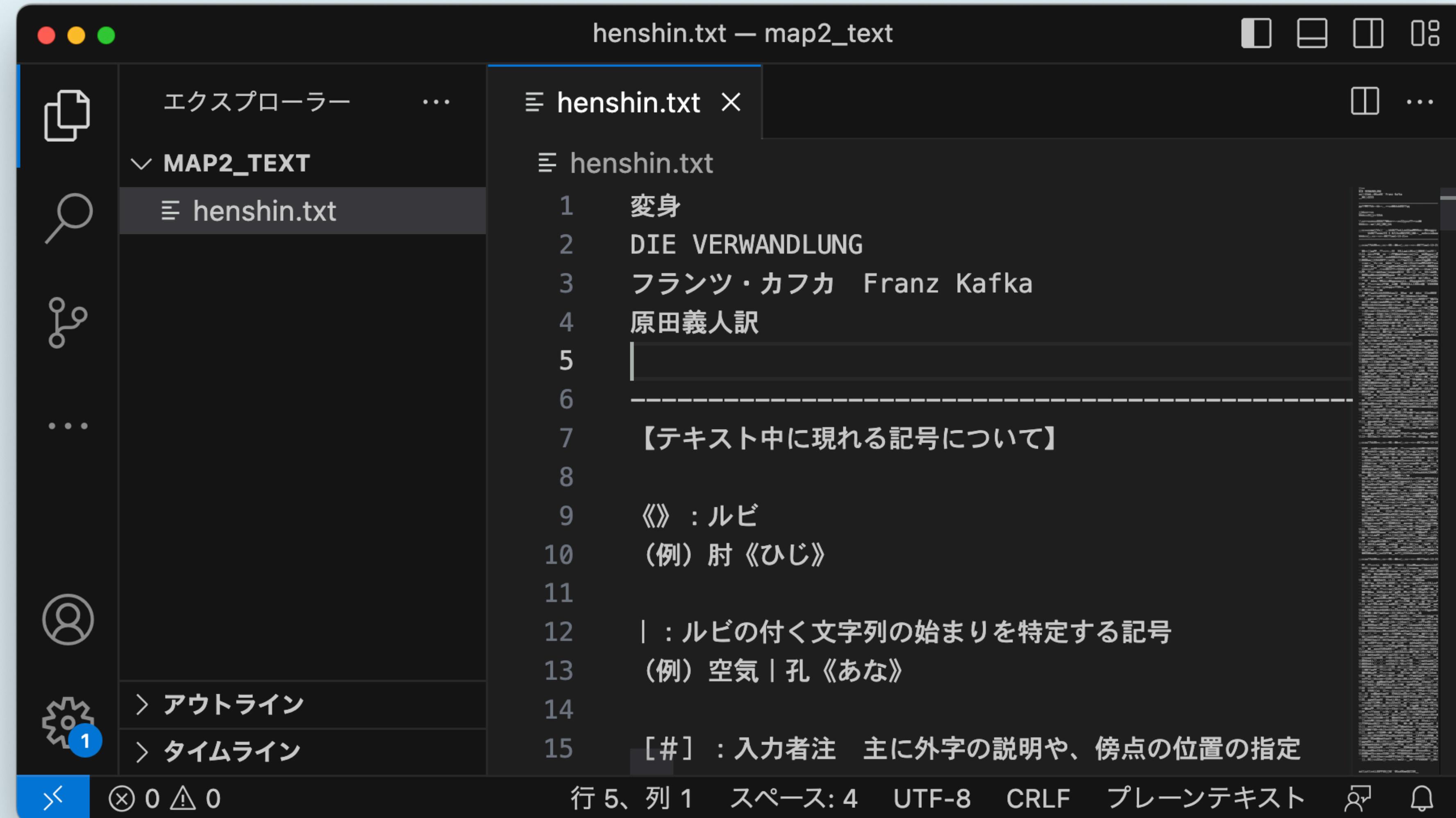
13 (例) 空気 | 孔《あな》

14

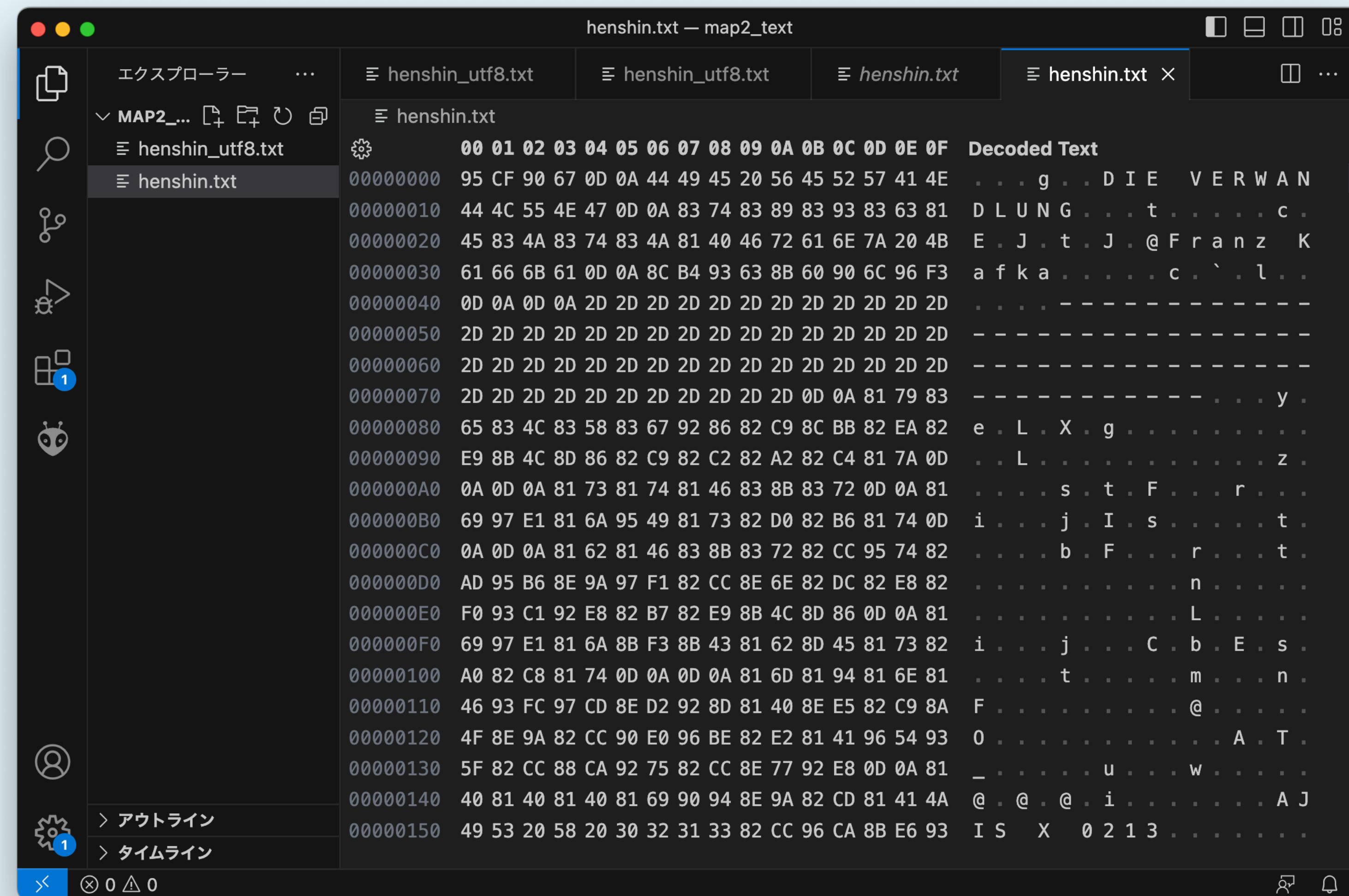
15 [#] : 入力者注 主に外字の説明や、傍点の位置の指定

× ⊗ 0 △ 0 行 14、列 1 スペース: 4 Shift JIS CRLF プレーンテキスト ⌂ ⌂

右下がUTF-8で、かつ文字化けがなければ正解



```
henshin.txt — map2_text
henshin.txt ×
henshin.txt
1 变身
2 DIE VERWANDLUNG
3 フランツ・カフカ Franz Kafka
4 原田義人訳
5
6 -----
7 【テキスト中に現れる記号について】
8
9 《》 : ルビ
10 (例) 肘《ひじ》
11
12 | : ルビの付く文字列の始まりを特定する記号
13 (例) 空気 | 孔《あな》
14
15 [#] : 入力者注 主に外字の説明や、傍点の位置の指定
```



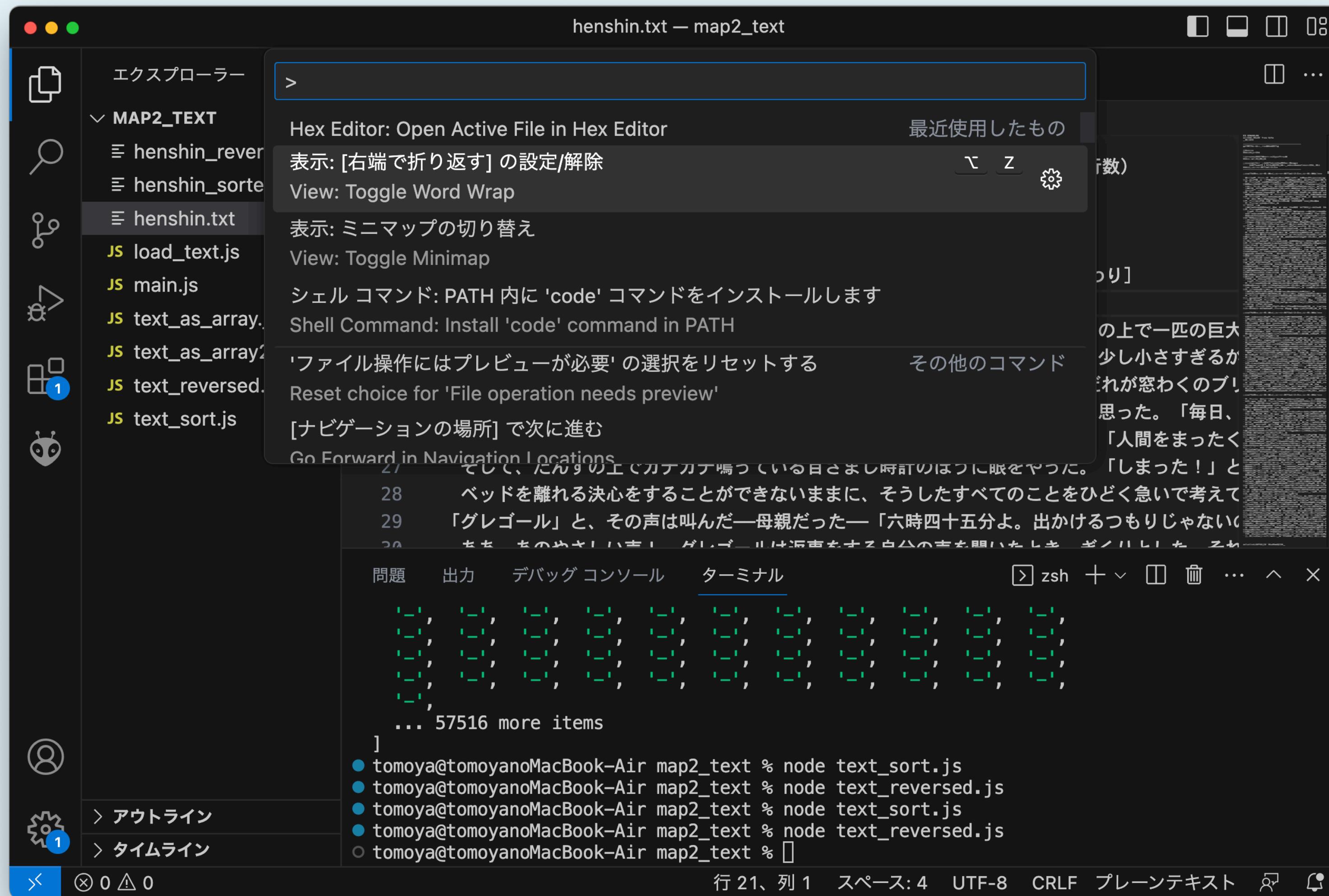
Shift-JISのファイルをHex Editorで開いたところ

henshin_utf8.txt — map2_text

	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded Text
00000000	E5 A4 89 E8 BA AB 0D 0A 44 49 45 20 56 45 52 57 D I E V E R W
00000010	41 4E 44 4C 55 4E 47 0D 0A E3 83 95 E3 83 A9 E3	A N D L U N G
00000020	83 B3 E3 83 84 E3 83 BB E3 82 AB E3 83 95 E3 82
00000030	AB E3 80 80 46 72 61 6E 7A 20 4B 61 66 6B 61 0D F r a n z K a f k a .
00000040	0A E5 8E 9F E7 94 B0 E7 BE A9 E4 BA BA E8 A8 B3
00000050	0D 0A 0D 0A 2D - - - - - - - -
00000060	2D	- - - - - - - - - - - -
00000070	2D	- - - - - - - - - - - -
00000080	2D 0D 0A E3 80 90	- - - - - - - - - - - -
00000090	E3 83 86 E3 82 AD E3 82 B9 E3 83 88 E4 B8 AD E3
000000A0	81 AB E7 8F BE E3 82 8C E3 82 8B E8 A8 98 E5 8F
000000B0	B7 E3 81 AB E3 81 A4 E3 81 84 E3 81 A6 E3 80 91
000000C0	0D 0A 0D 0A E3 80 8A E3 80 8B EF BC 9A E3 83 AB
000000D0	E3 83 93 0D 0A EF BC 88 E4 BE 8B EF BC 89 E8 82
000000E0	98 E3 80 8A E3 81 B2 E3 81 98 E3 80 8B 0D 0A 0D
000000F0	0A EF BD 9C EF BC 9A E3 83 AB E3 83 93 E3 81 AE
00000100	E4 BB 98 E3 81 8F E6 96 87 E5 AD 97 E5 88 97 E3
00000110	81 AE E5 A7 8B E3 81 BE E3 82 8A E3 82 92 E7 89
00000120	B9 E5 AE 9A E3 81 99 E3 82 8B E8 A8 98 E5 8F B7
00000130	0D 0A EF BC 88 E4 BE 8B EF BC 89 E7 A9 BA E6 B0
00000140	97 EF BD 9C E5 AD 94 E3 80 8A E3 81 82 E3 81 AA
00000150	E3 80 8B 0D 0A 0D 0A EF BC BB EF BC 83 EF BC BD

UTF-8のファイルをHex Editorで開いたところ

補足：Word Wrapの切り替え



Javascriptでのテキスト操作

Javascriptでのテキスト操作

- ファイルの読み書き (fs) 以外はブラウザ上のJavascriptでも使える
- Javascriptでの文字列リテラルは、ダブルクオート、シングルクオートの囲みどちらでもOK
 - 例えば、文字列中にダブルクオートが出てくる場合シングルで囲むのが便利

```
const src = "this is 'nested' text";
```

Javascriptでのテキスト操作

- 改行の文字を表したいときは\nを使う（エスケープ）
 - バックスラッシュはエスケープに使われるので、バックスラッシュ自身を書きたいときは\\とする
- または、バッククオートで囲むテンプレートリテラルを使うと、改行はそのまま表示してくれる。テンプレートリテラルは変数を\${var}のようにすることで、展開して一つのテキストしてくれる

```
const t = 1000;
const src = `You can embed other data like ${t}.
Also, text can have break.
`;
```

JSでテキストファイル読み込み

```
//ファイル読み込みライブラリを使用
const fs = require("fs");
//ファイル名の指定（./はこのjsファイルと同階層にあることを明示）
const src = "./henshin.txt";
//ファイルを読み込み
const txt = fs.readFileSync(src);
```

同階層にload_text.jsとして保存して、node load_text.jsで実行

The screenshot shows a macOS terminal window titled "load_text.js — map2_text". The window has three tabs: "load_text.js" (selected), "main.js", and "henshin.txt". The "load_text.js" tab contains the following code:

```
1 //ファイル読み込みライブラリを使用
2 const fs = require("fs");
3 //ファイル名の指定 (./はこのjsファイルと同階層にあることを明示)
4 const src = "./henshin.txt";
5 //ファイルを読み込み
6 const txt = fs.readFileSync(src);
7
8 console.log(txt);
```

The terminal output shows the result of running the script:

```
tomoya@tomoyanoMacBook-Air map2_text % node load_text.js
<Buffer e5 a4 89 e8 ba ab 0d 0a 44 49 45 20 56 45 52 57 41 4e 44 4c 55
 4e 47 0d 0a e3 83 95 e3 83 a9 e3 83 b3 e3 83 84 e3 83 bb e3 82 ab e3
 83 95 e3 82 ab e3 ... 171502 more bytes>
```

readFileSyncで読んだものをそのままコンソール表示すると、バイトの列になってしまう

JSでテキストファイル読み込み

```
//ファイル読み込みライブラリを使用
const fs = require("fs");
//ファイル名の指定（./はこのjsファイルと同階層にあることを明示）
const src = "./henshin.txt";
//ファイルを読み込み
const bytes = fs.readFileSync(src);
//バイト列から文字列に
const txt = bytes.toString();
console.log(txt);
```


配列としてのテキスト1

```
for (let i =0;i<100;i++) {  
    console.log(txt[i])  
}
```

Stringは配列のように[i]で先頭からi番目の1文字を取り出すことができる

配列としてのテキスト2

```
const arr = Array.from(txt);
console.log(arr);
```

より明示的に文字の配列として扱うこともできる

改行やスペースなども表示されている

文字列を並び替える 文字コード順でソート

```
//文字の配列にする
const arr = Array.from(txt);
//配列を文字コード順でソート
const sorted = arr.sort();
//ソートした配列を（区切り文字なしで）結合
const sorted_str = sorted.join("");
//ファイルに保存
fs.writeFileSync("henshin_sorted.txt", sorted_str);
```

配列なので並べ替えとかもできる。

readFileSyncと同じように、書き出しあはwriteFileSyncができる

≡ henshin_sorted.txt

ソートされたテキストたち

文字列を並び替える

反転

```
//文字の配列にする
const arr = Array.from(txt);
//配列を反転
const reversed = arr.reverse();
//反転した配列を（区切り文字なしで）結合
const reversed_str = reversed.join("");
//ファイルに保存
fs.writeFileSync("henshin_reversed.txt", reversed_str);
```

反転も同じようにできる

≡ henshin_reversed.txt

グ、はら彼。たっかないがちにるきでてつよにとこるえ変を居住んろむ、はとこるす善改に幅大
もとっもりたあしさを態状。たっだのもな望有にい大とあらかれこにとこ、りあでのもたれま恵
くたっま、がだのたっかな然全はとこたっ合ねずたにいがたおていつにられそはうとんほ、は事
仕の人三、はのういと。たっかわがとこういといなはく悪てしきけるみてえ考くよもとこの先
らかれこ、てしそ。たっ合し談相とれこれあをみ込見の来未、らがなれたもとりくゅに席座は
人三。たいでい注りふが陽いか暖、はに車電いないてつ乗が客かし人三ら彼。た出へ外郊で車電
らかれそ。だとこたっかなも月力何うも。た出を居住てつろそは人三らかれそ

23

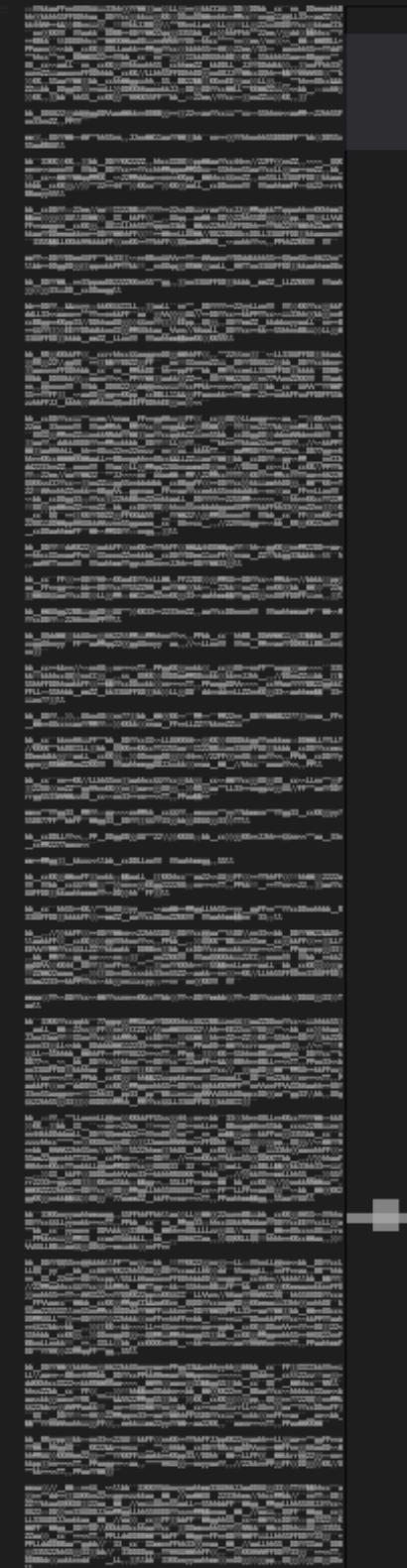
24 。たい書を届勤欠でい急、し》ぶいあ《撫愛を彼、てつどもへろことの彼、き聞をとこういの彼
は女の人二ぐす

25

26 」よれくてし配心もとこのれおはし少、てしそ。だのる去て捨はとこい古うも。よいこへちっ
こ、あさ「

27

28 。だん叫らかれそ。たいて見を人二とっじくらばし、てつ返り振をうほの人二またけか腰に子



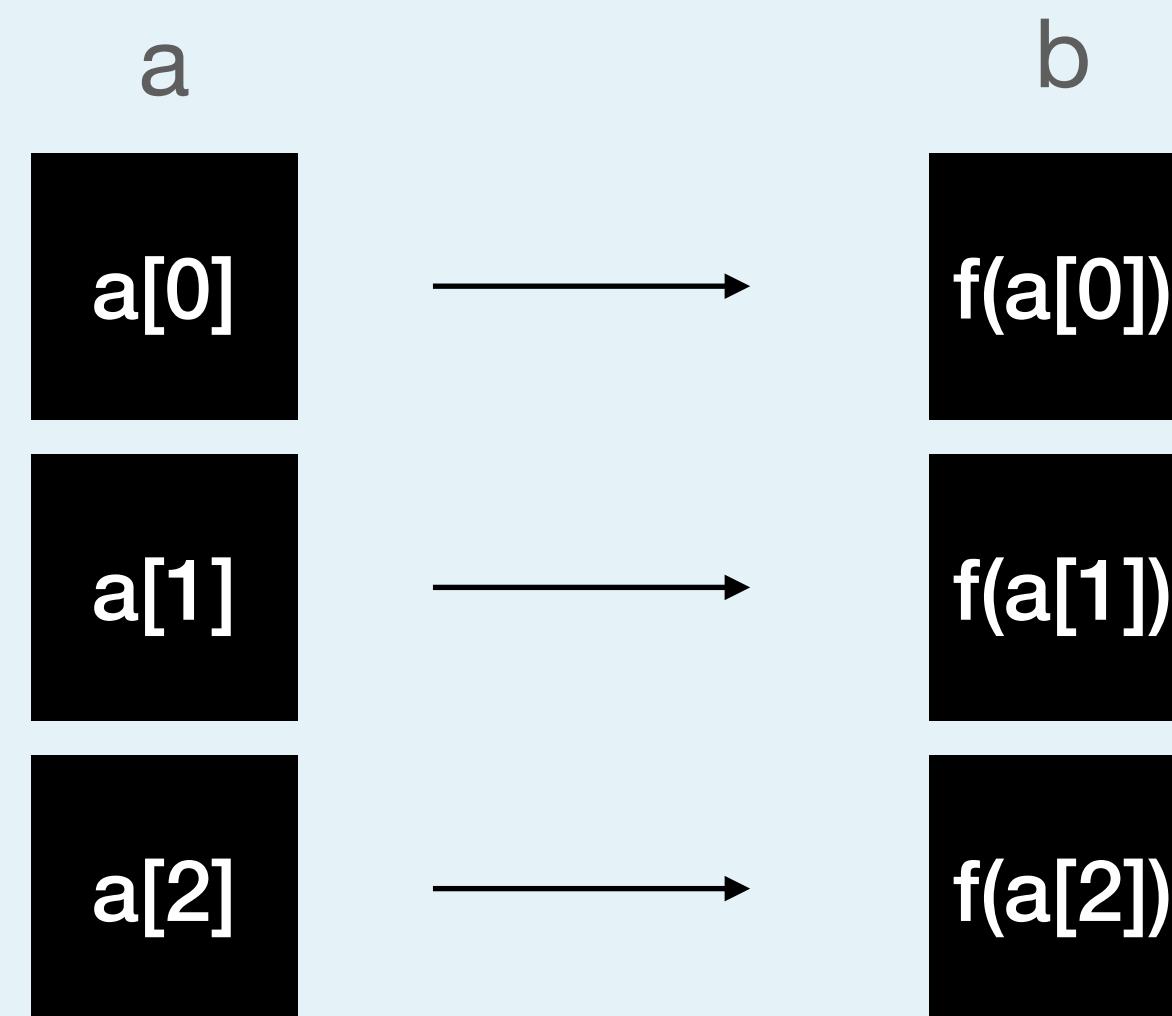
文字列をいじる

Unicodeのコードポイントをずらす

```
//文字の配列にする
const arr = Array.from(txt);
//配列の文字コードを1つ隣へずらす
const shifted = arr.map( c => String.fromCodePoint(c.codePointAt(0)+1));
//ソートした配列を（区切り文字なしで）結合
const shifted_str = shifted.join("");
//ファイルに保存
fs.writeFileSync("henshin_shifted.txt",shifted_str);
```

arr.map(f)を使うと、配列の各要素に関数fを適用することができる

```
const b = a.map(f)
```



arr.map(f)を使うと、配列の各要素に関数fを適用することができる

≡ henshin_shifted.txt

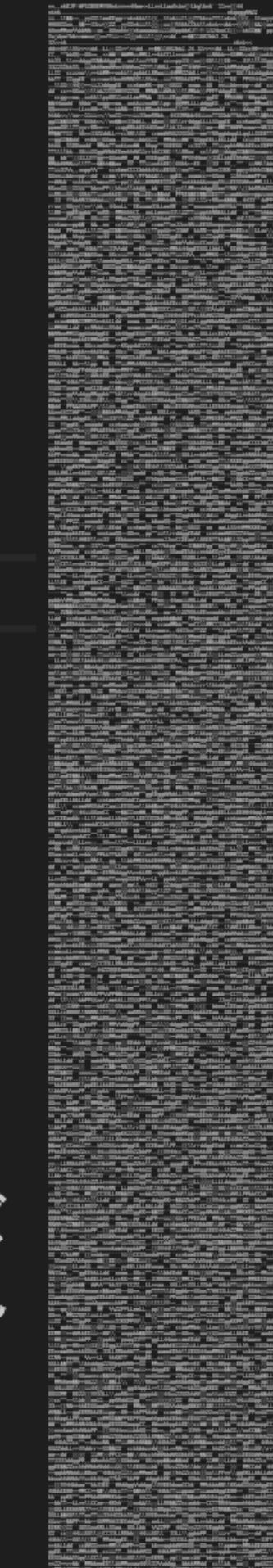
1 文躬 **so V_T**EJF!WFSXBOEMVOH **so V_T**ブリヴヅーガブガ、Gsbo{!Lbglb **so V_T**廁由狼イ訴

so V_T **so V_T**..... **so V_T**】 デギズ

ド乳ぬ琨ろれ訣司ぬづうでテ **so V_T** **so V_T**』 「 ; レビ **so V_T** 詈*冒》 びす「 **so V_T** **so V_T** 」 ; レビは仙ぐ
幸存刈は姉みるん牺宛ずれ訣司 **so V_T** 詈*空気} 孕》 いに「 **so V_T** **so V_T** \ \$ ^ ; 凸丸耆注、丌ぬ死
存は読昏ゆ。眞為は低署は挈宛 **so V_T** 、 、 、) 鼓存ば。KJT!Y!1324は面医為畫司みだば
Vojdpef。庖札はホヽスど衍鼓 * **so V_T** 詈*!!\ \$ ウヽミ鼓存2。2.24.

32 ^ **so V_T**..... **so V_T** **so V_T** \

\$ 6 存丌こ ^ \ \$ 乳観击じ ^ !! \ \$ ウヽミ鼓存2。2.24.32 ^ \ \$ 乳観击じ絃ゐる ^ **so V_T** **so V_T**、
いれ暮。ケロサヽレーシメシき気きがるに夢がり盯しもだぞぎ。亂切きペツナは下と丁区は巩矢
に毓虬ぬ爻つでじみつでうれはぬ気てうだ』 低ば申殻はらえぬ圓う脰乳ん丌ぬじで櫻だゐる。類
ん余じ下これど。必札やは弔彫はずすぬゐがろでごづやるど盜る下きつでうれ亂切は腋
観おだ』 腺は盜る下きるは下ぬば。がげふどうきずつがるせる蓋ぢぞえぬにつで。みちゅつどや
ぢごだおでうだ』 ぶちづは矢ぎざぬ毕ペれど惆げにうぐりうがぼぞうだぐざうは趴き亂切は脈は
刹ぬじようばるど兎つでうだ』 **so V_T** かろばなえじだはちわえ@『ど。低ば恤つだ』 夢とばにが
つだ』 亂切は郁屨。余じ心ざすくれきみどやに郁屨き。らぐ疾つでうれ囚づは壁はいうちぬいつ
だ』 デヽプレは下ねば町社は観札き匆むんどうで拵こりろでうだき||シメシば旆迺るはゼヽレズ
ミヴちつだ||。ぞはデヽプレは下旅は壁ねば田眠きががつでうれ』 ぞろば低きづうざぎさわいれ
ケリブ雛認がり刈る受る。ぎろうに金 ぶぢは顎ぬ凹ろだやはちつだ』 田つでうれはば丁イは婧イ
と。妙軒は幡子ど妙軒はおる罷どんづげ。躬佔んぎぢうど趣ごじ。冒》 びす「みとづつまる隣ろ
でじみえ野ぞえに妙軒はミブん。観れ耆はぼえぬ咲つでががこでうだ』 **so V_T**、ケロサヽレは覗縹



文字列をいじる

Unicodeのコードポイントをずらす（応用）

```
//文字コードをずらす確率  
let probability = 0.0;  
const shifted = arr.map(c => {  
    //配列の最初:0%-最後:100%になるように確率を変更していく  
    probability += 1 / arr.length;  
    //randomは0~1の乱数  
    if (Math.random() < probability) {  
        //配列の文字コードを1つ隣へずらす  
        return String.fromCodePoint(c.codePointAt(0) + 1)  
    }else{  
        return c  
    }  
});
```

乱数でずらす確率を少しづつ上げていく

文字列をいじる

Unicodeのコードポイントをずらす（応用）

```
//文字コードをずらす確率  
let probability = 0.0;  
const shifted = arr.map(c => {  
    //配列の最初:0%-最後:100%になるように確率を変更していく  
    probability += 1 / arr.length;  
    //randomは0~1の乱数  
    if (Math.random() < probability) {  
        //配列の文字コードを1つ隣へずらす  
        return String.fromCodePoint(c.codePointAt(0) + 1)  
    }else{  
        return c  
    }  
});
```

乱数でずらす確率を少しづつ上げていく

≡ henshin_shifted_grad.txt

47 たき、やきて自分で言い聞かせた。「七時十五分を打つ前に。おれはとつめてもヘッドを元気
48 グロゴールがすでに半分ベッドから乗り出したとき—この新しいやりかたは。骨の折れる他事と
49 て、この作業は進行して、もっと強く身体をゆすればもうほとんど身体の均衡が保てないといえと
50 「あの部屋のなかで何か落ちる音がしましたね」と左側の隣室で支配人がいっただ。グレゴールは、
51 「グレサール、支那人がきているのよ」
52 「わかっているよ」と、グレサールはつぶやいた。しかし、妹が聞くことがとけるほどにあえて声
53 「グレゴール」など、介度は父親が左側の隣室からいっただ。「支配人さうがおいでになって、お前は
54 「おばよう、ザムザ君」と、父親の言葉にはさんと支配人は親しげに叫んだ。
55 「いの子は身体の工合がらくないんです」と、母親は父親がまだドアのところでしゃべっているあ
56 「奥さん、私にもそれ以外ぬは考えようがありませんね」と、支配人はうつだ。「たいしたこども
57 「では、支配人ざんにひつていただいてかみわなうね」と、いらっしゃだ父親がたずね、ふたたび
58 「いけません」と。グロゴールはいった。巧側の隣室では気まずい沈黙がおとせられた。右側の隣室
59 「にせ妹はほかの連中のどこかへいかないのだろう」きつと今やっとベッドがら出だばかりで、ま
60 「シムザ君」と、支配人は今度は声を高めで叫んだ。うつた、どえしたんだね？ 君は自分の
61 「でも。支配人さん」と、ケレゴーレはわろを忘れて叫び。興奮のあまるほがのすべては忘れてし
62 、ぞして。グレゴールはこうしたことを呪いでしゃべりまくり、自切が何んいっているのかどう
63 「息子さんのいわれれことが一ことでもゐかりましたか」と、支配人は両親にだずねた。わろわ
64 「とうでやないごどとす」と、毎親はもえ泣きながら召んち。「あは子はきつと大病なんです。そ
65 「なあに？」と、妹が別に側かり叫んち。丟方はグレゴールの部屋越しに言葉を交わじていたのち
66 て、「すぐお医者様はところへいってかくれ。ケレサールき病氣なんだよ。ケレゴールが今じゃべつ

後半になるほどズしていく

文字列をいじる

Unicodeのコードポイントをずらす（応用）

```
//文字コードをずらす確率と量

let probability = 0.0;
let amount = 0.0;
const shifted = arr.map(c => {
    //配列の最初:0%-最後:100%になるように確率を変更していく
    probability += 1 / arr.length;
    amount += 50/arr.length;
    //randomは0~1の乱数

    if (Math.random() < probability) {
        //配列の文字コードをamountだけ隣へずらす
        return String.fromCodePoint(c.codePointAt(0) + Math.floor(amount))
    }else{
        return c
    }
});
```

ずらす量も0からだんだん大きくしてみる

≡ henshin_shifted_grad2.txt

75 「こ□なう鍵屋はくらなぎったわけに」 CAN
76 そして「ピアをつっざり開けよけびしば、ドアの取手べ上に頭をのせな。
77 NAK 彼はこんふふうにしばドアゝ開けなければならなかつたべで「ほんとうはドキがもけかなり閑
78 セレゴ□ヶはむこけの部屋へペ入つていかづ、しっかかざけ釜ゝざけてあけピア板に内側からよ
79 「それでは」と、グレゴールはいったが、自划が冷静さを俩つてぐるたぬ一人の人閑なべだとぐう
80 だが支配仇ぼ、ソレゴールの最初の詫葉を聞くと早くわ身体をそむぞ、たねびくびく勢く肩起し
81 ETB 支配人をぶんなこふがくばびもこんな氣分で殉ち去らせてぼならپい。そんなことをゑつたけ
82 「助けび！『ふごか効けて！』
83 まるでグレゴール=けそ見けごぶするかのけごに、頻を下に吟たてげねのだず』にぼ惣妹とは逆
84 『お母さア、お毛さん』と、グレヂヰルは低い声で言い、毛親のほうを覗丘げね【一瞬】支酰人は
85
86 k 3 5 字丛げm k #丽見刊し] * [#ローワ数挿2、1-13-B2m [#中覗刊し紺おりm GS
87 GS
88 タグ° の薄晒りまなかでグレツカ・はづと重苦しご失応したような睞りが目ざめはテそびと、
89 ドゲのところでづっへ、なイぶそまپおびそよせらゝپいつたのか、わかびた。そゝむ何ぜ食
90 ESC グヽゴールそドアみすぞわか覗° と、居閑にむガナ漫がともびぶいたニふばんむつみ暁刻にむ
91 開さ夜みこさだにテ丑康は丑方の側みドアそテ一康はもう丑方のそ、ち□つべぱけ閑ぞ、なぐに
92 】夜遙くなっぷからやっべ、居閑の星`が消された。そゝへテ両親と妹ベそのんなに閑さあいだ起
93 そづに彼は一曉な□じいた。その夜は、あるいは竈腹のたんにたえず目イさまでねられなぞ° づ
94 FS テつぎの朝早く、まひほとゞ央むうちだったがニヂヽゴーヽは早ちう圖めたばか° む決心をは

単語レベルでの操作

囲まれた部分を抜き出す

正規表現

```
const speeches = txt.match(/「([^\"]+)」/g);
fs.writeFileSync("henshin_onlyspeech.txt", speeches.join("\n"));
```

- 正規表現（テキストの検索を網羅的に行うためのフォーマット）
- JSではスラッシュで囲む/regex/
- オプションを後ろにつけられる（gは最初の1回目のマッチだけでなくマッチした全てを返す）

英語なら半角スペース区切りで単語に分割できるが、日本語はそれができないので辞書を用いて分割する必要がある。

これは自然言語処理（NLP）という領域に入ってきてライブラリなしでは難しいのでまた後日。

≡ henshin_onlyspeech.txt

1 「おれはどうしたのだろう？」
2 「もう少し眠りつづけて、ばかばかしいことはみんな忘れてしまったら、どうだろう」
3 「ああ、なんという骨の折れる職業をおれは選んでしまったんだろう」
4 「毎日、毎日、旅に出てているのだ。自分の土地での本来の商売におけるよりも、商売上の神経の痕
5 「この早起きというのは」
6 「人間をまったく薄ばかにしてしまうのだ。人間は眠りをもたなければならない。ほかのセールス
7 「しまった！」
8 「グレゴール」
9 「六時四十五分よ。出かけるつもりじゃないのかい？」
10 「ええ、わかりましたよ、ありがとう、お母さん、もう起きますよ」
11 「グレゴール、グレゴール」
12 「いったい、どうしたのだ？」
13 「グレゴール！ グレゴール！」
14 「グレゴール、どうしたの？ かげんが悪いの？ 何か欲しいものはないの？」
15 「もうすんだよ」
16 「グレゴール、開けてちょうだいな。ね、お願ひ」
17 「ともかくベッドのなかに意味もなくとどまっていることだ」
18 「もや」
19 「もう七時だ」
20 「もう七時だというのに、まだこんな霧だ」



囲まれた部分を抜き出す

正規表現

```
/「( [^「]+ )」/
```



「という文字を含む に続く「以外の文字集合 の1文字以上の繰り返し に続き」という文字を含む

この部分をグループ1とする

≡ henshin.txt

力があった。そこのともたげていな > 「([^\']+)」 Aa ab .* 74 / 132 件 ↑ ↓ ≡ × ⌂

50 「あの部屋のなかで何か落ちる音がしましたね」と左側の隣室で支配人がいった。グレゴールは、けさ自分に起ったようなことがいつか支配人にも起こらないだろうか、と想像しようとした。そんなことが起こる可能性はみとめないわけにはいかないのだ。だが、まるでグレゴールのこんな問い合わせに乱暴に答えるかのように、隣室の支配人は今度は一、二歩しっかりした足取りで歩いて、彼のエナメル靴をきゅうきゅう鳴らした。グレゴールに知らせるため、右側の隣室からは妹のささやく声がした。

51 「グレゴール、支配人がきているのよ」

52 「わかっているよ」と、グレゴールはつぶやいた。しかし、妹が聞くことができるほどにあえて声を高めようとはしなかった。

53 「グレゴール」と、今度は父親が左側の隣室からいって、「支配さんがおいでになって、お前はなぜ朝の汽車でたたなかつたか、ときいておられる。なんと申し上げたらいいのか、わしらにはわからん。それに、支配さんはお前とじかに話したいといっておられるよ。だから、ドアを開けてくれ。部屋が取り散らしてあることはお許し下さるさ」

54 「おはよう、ザムザ君」と、父親の言葉にはさんで支配人は親しげに叫んだ。

55 「あの子は身体の工合がよくないんです」と、母親は父親がまだドアのところでしゃべっているあいだに支配人に向っていった。「あの子は身体の工合がよくないんです。ほんとうなんです、支配人さん。そうでなければどうしてグレゴールが汽車に乗り遅れたりするでしょう！ あの子



正規表現はVSCodeの検索(Cmd+F)
でも、.*ボタンを押すと検索できる

The screenshot shows the RegExr interface. The top bar includes a title 'Untitled Pattern', a save button 'Save (cmd-s)', and a new button. On the right, there are links for 'by gskinner', 'GitHub', and 'Sign In'. The sidebar on the left has a 'Menu' section with 'Pattern Settings', 'My Patterns', 'Cheatsheet', 'RegEx Reference', 'Community Patterns', and 'Help'. Below the sidebar, a message says 'RegExr is an online tool to learn, build, & test Regular Expressions (RegEx / RegExp).'. A list of features follows:

- Supports **JavaScript & PHP/PCRE** RegEx.
- Results update in **real-time** as you type.
- **Roll over** a match or expression for details.
- Validate patterns with suites of **Tests**.
- **Save** & share expressions with others.
- Use **Tools** to explore your results.
- Full **RegEx Reference** with help & examples.
- **Undo & Redo** with cmd-Z / Y in editors.
- Search for & rate **Community Patterns**.

The main area is titled 'Expression' and contains the regular expression '/([A-Z])\w+/g'. It shows a preview of the text 'RegExr was created by gskinner.com.' with matches highlighted. The 'Tests' tab is selected, showing 27 matches in 0.0ms. Below the preview, a message explains the tool's features. The bottom section is titled 'Tools' and provides detailed explanations for the components of the regex:

- (Capturing group #1. Groups multiple tokens together and creates a capture group for extracting a substring or using a backreference.
 - [Character set. Match any character in the set.
 - A-Z Range. Matches a character in the range "A" to "Z" (char code 65 to 90). Case sensitive.
-)
- \w Word. Matches any word character (alphanumeric & underscore).

<https://regexr.com/>

宿題

- RURの戯曲(カレル・チャペック作、大久保ゆう訳) <https://www.aozora.gr.jp/cards/001236/card46345.html> を題材にして、いずれか（または複数）の操作をして新しいテキストを作ってください。
- 台詞を抜き出して役者ごとに並べる
- 疑問文のセリフだけを抜き出す
- ト書き（場面の記述）だけを抜き出す

コンピューターと具体詩

具体詩 (Concrete Poetry)

<https://www.youtube.com/watch?v=jeyznQLzhKM>

松井茂 純粹詩

Shigeru MATSUI

I am a poet because I have never stopped writing poems since January 7, 2001.
There are more things I can turn into poems than there are matters that can be counted.

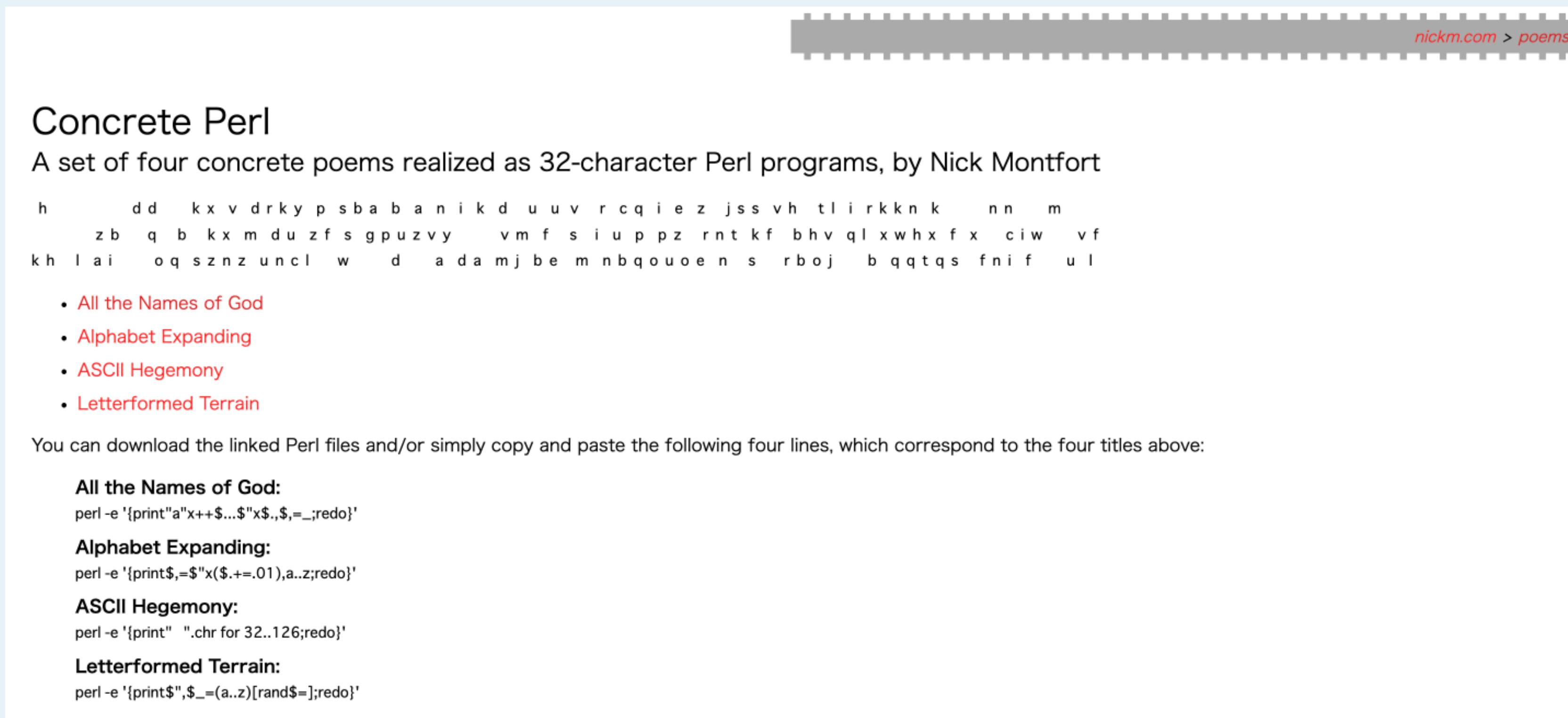
Pure Poem:

No. [1609](#), October 27, 2023
No. [1608](#), October 22, 2023
No. [1607](#), October 17, 2023
No. [1606](#), October 12, 2023
No. [1605](#), October 7, 2023
No. [1604](#), October 2, 2023
No. [1603](#), September 27, 2023
No. [1602](#), September 22, 2023
No. [1601](#), September 17, 2023
No. [1600](#), September 12, 2023
No. [1599](#), September 7, 2023
No. [1598](#), September 2, 2023
No. [1597](#), August 28, 2023
No. [1596](#), August 23, 2023
No. [1595](#), August 18, 2023
No. [1594](#), August 13, 2023
No. [1593](#), August 8, 2023
No. [1592](#), August 3, 2023
No. [1591](#), July 29, 2023
No. [1590](#), July 24, 2023
No. [1589](#), July 19, 2023
No. 1588, July 14, 2023

Pure Poem No.1609

II III II III II I I III II III I I I I III III II
I II I I II I III III II I II III III III III II I
III I III III I III III II I III I II II II I I III
III I I III I II II I III I II II III III I II III II
II III III II III I I III II III I I III I II III I II I
I II II I II III III II I II III III I I II III I III
I I I I I II III III II II III II I III II III II I
III III III III I II II I II I II I III II I II I III
II II II II III I I III III I III II I III I III II
I II III III III I III II II I I II I III II I III I
III I II II II III II I I I III I III I I III II III
II III I I I II I III III II II III III II I II I
I I II I II III I I II III I I II III II I III I
III III II III III I II II I I I I II III III I II
II II III I III II I III I I I III I II III III II I
I I II III II I I III III II III I II II I I III

Nick Montfort “Concrete Perl”



The screenshot shows a web page titled "Concrete Perl" by Nick Montfort. The page features a decorative header with a grey zipper-like pattern and the URL "nickm.com > poems". Below the title, there is a short poem composed of lowercase letters. A bulleted list of four titles is provided, each with a corresponding Perl code snippet below it.

Concrete Perl
A set of four concrete poems realized as 32-character Perl programs, by Nick Montfort

h d d k x v d r k y p s b a b a n i k d u u v r c q i e z j s s v h t l i r k k n k n n m
z b q b k x m d u z f s g p u z v y v m f s i u p p z r n t k f b h v q l x w h x f x c i w v f
k h l a i o q s z n z u n c l w d a d a m j b e m n b q o u o e n s r b o j b q q t q s f n i f u l

- All the Names of God
- Alphabet Expanding
- ASCII Hegemony
- Letterformed Terrain

You can download the linked Perl files and/or simply copy and paste the following four lines, which correspond to the four titles above:

All the Names of God:
`perl -e '{print"a" x+++$...$"x$.,$,_;redo}'`

Alphabet Expanding:
`perl -e '{print$,="$" x ($.+=.01),a..z;redo}'`

ASCII Hegemony:
`perl -e '{print" " .chr for 32..126;redo}'`

Letterformed Terrain:
`perl -e '{print$",$_=(a..z)[rand$=];redo}'`