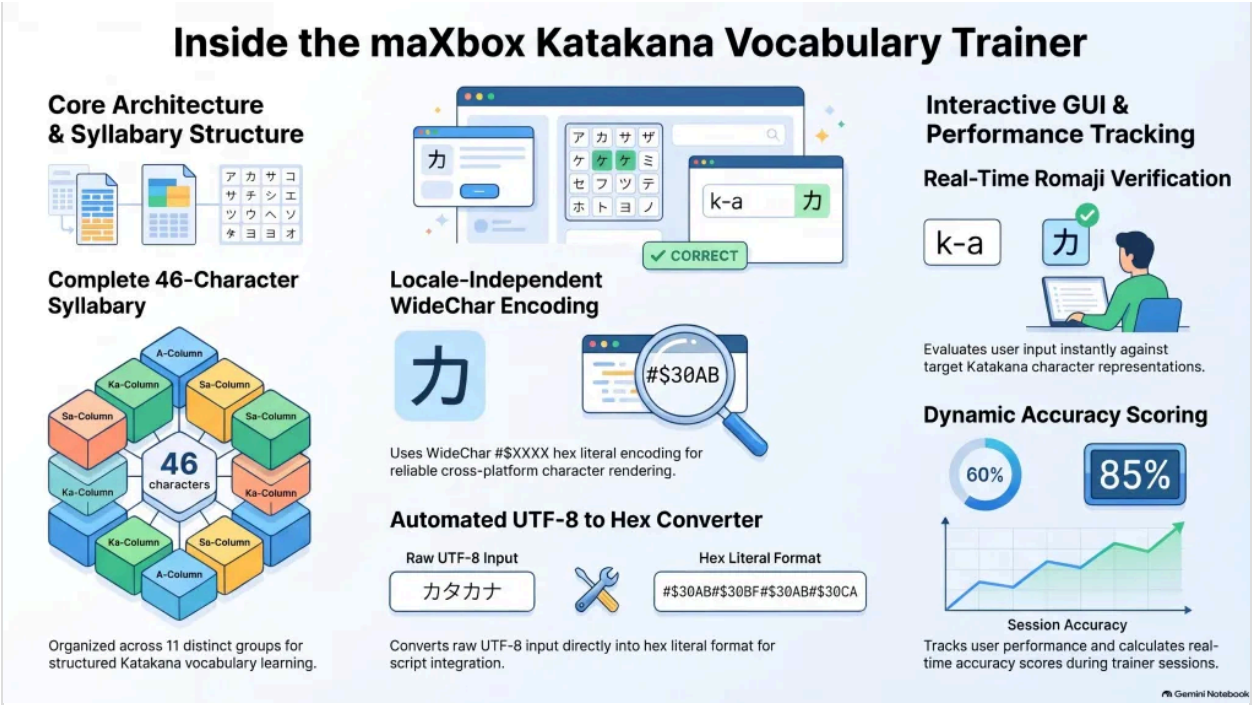


Katakana Language Trainer

November 5, 2026

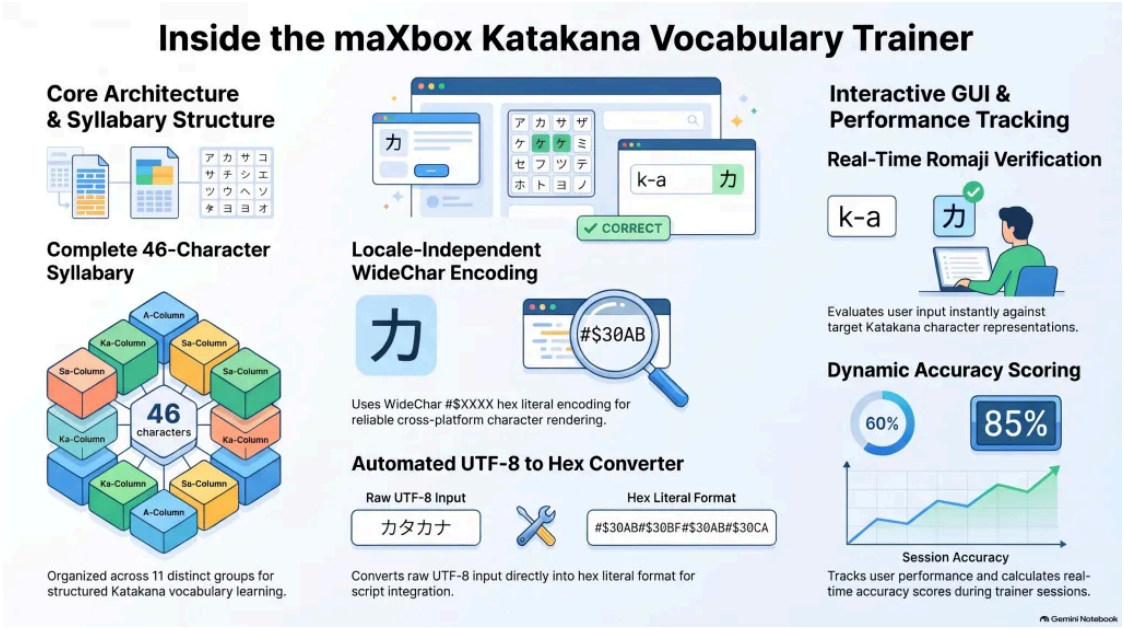
This is part of the Tutorial 205. Katakana is a Japanese syllabary used alongside hiragana and kanji. It represents the same set of sounds as hiragana but uses distinct, more angular shapes. It originated from components of kanji during the 8th–9th centuries. Wikipedia

Structure

- Each character represents a mora (a rhythmic unit in Japanese).
- Includes vowels (ア, イ, ウ, エ, オ), consonant–vowel combinations (e.g., カ, サ, タ), and the nasal ン. Wikipedia
- Long vowels are marked with ー (chōonpu).
- Dakuten (゛) and handakuten (゜) modify consonant sounds (e.g., カ→ガ, ハ→パ).

We've built a robust, fully localized Katakana Syllabary & Vocabulary Trainer (`katakana_language_trainer-v9.pas`) for maXbox.

https://sourceforge.net/projects/maxbox5/files/examples/1507_katakana_kanji_language_trainer-v9.pas/download



Here is a structural review of how the script evolved, its core architecture, and key technical highlights:

---

1. Complete 46-Character Syllabary Coverage

The dataset covers the entire basic Japanese Katakana syllabary across 11 distinct phonetic groups:

- Vowels (A-row): ア, イ, ウ, エ, オ
- K-Group (Ka-row): カ, キ, ク, ケ, コ
- S-Group (Sa-row): サ, シ, ス, セ, ソ
- T-Group (Ta-row): タ, チ, ツ, テ, ト
- N-Group (Na-row): ナ, ニ, ヌ, ネ, ノ
- H-Group (Ha-row): ハ, ヒ, フ, ヘ, ホ
- M-Group (Ma-row): マ, ミ, ム, メ, モ
- Y-Group (Ya-row): ヤ, ユ, ヨ
- R-Group (Ra-row): ラ, リ, ル, レ, ロ
- W-Group & Singular N: ワ, ヲ, ン

---

2. 100% Locale-Independent WideChar Hex Encoding

To prevent ANSI code-page rendering corruption (`?` or broken glyphs) across different Windows regional settings:

- Character Symbols: Each character is declared via WideChar hex literals (e.g., `#$30A2` for ア through `#$30F3` for ン).
- Vocabulary Hints: All Japanese loanword spellings are constructed using WideChar hex concatenations: `// America (アメリカ) Hints :=`

```
'Vocab: America ('#$30A2#$30E1#$30EA#$30AB' / amerika)'; //
Bread/Pan (パン) Hints := 'Vocab: Pan / Bread
('#$30D1#$30F3' / pan)';
```

Advertisement



Kalkrand im WC? Dieser einfache Trick löst ihn ohne...

Try Fuugu

Privacy Settings

3. GUI Engine & User Experience

- Dynamic Card Renderer: Picks a random card (`Random(46)`), updating the large 64pt Katakana display, the group category, and the U+XXXX hex tag.
- Validation & Instant Feedback: Evaluates user Romaji input against the expected string, rendering results in Green (Correct) or Red (Incorrect) with correct answer reveal.
- Real-time Accuracy Calculation: $$ \text{Accuracy %} = \frac{\text{Correct Answers}}{\text{Total Attempts}} \times 100 $$
- Session Memo Logger: Logs every attempt with attempt index, Katakana symbol, Unicode hex tag, user input, and result status for review.

Because the 101 characters and vocabulary hints use 16-bit WideChar hex literals (#$XXXX), maXbox will render all Katakana, Kanji, and loanword strings correctly regardless of your Windows system's regional language settings or active ANSI code page.

Is this sign 水 sui or mizu as translation or romaji?

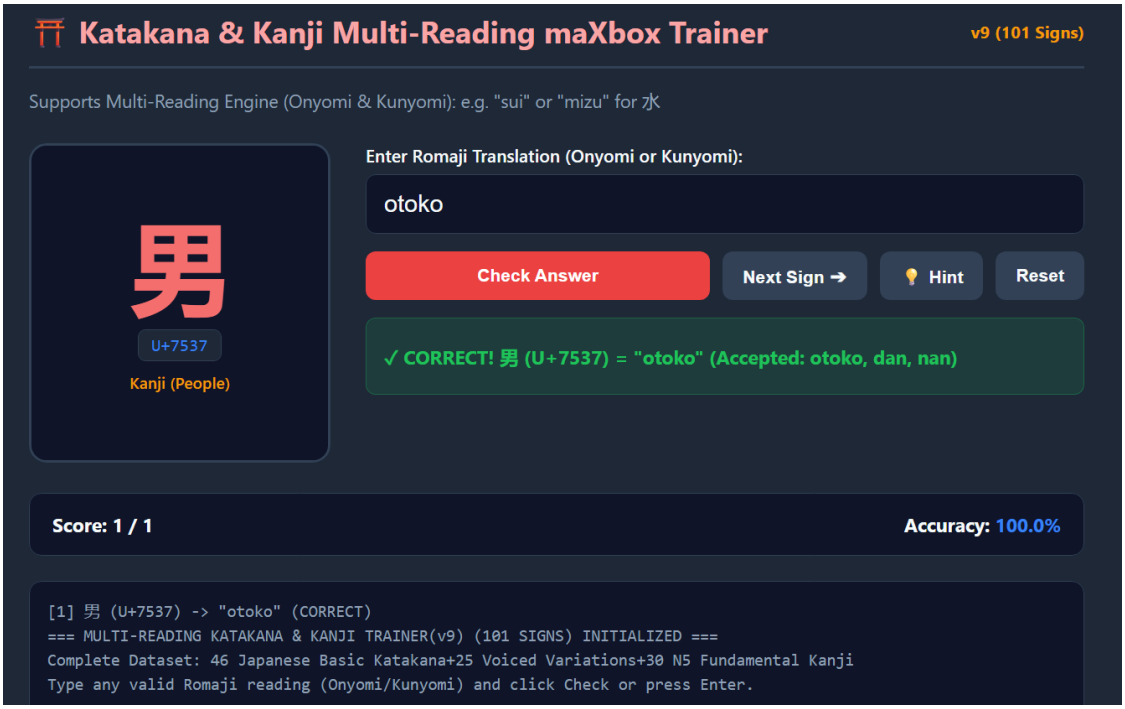
Both are correct! It depends on how the character is used:

- In our trainer program, the expected target answer is sui. This is the *Onyomi* (Sino-Japanese reading), which is used in compound words like 水曜日 (*Suiyoubi* / Wednesday) and 水田 (*Suiden* / rice field).

- When standing alone as a word, it is read as **mizu**. This is the *Kunyomi* (native Japanese reading) when referring directly to "water" (e.g., お水 / *omizu*).

I have updated the trainer dataset to support **multiple valid readings (both Onyomi and Kunyomi)** as well as expanded compound vocabulary for all 30 Kanji characters, and published:

Japanese Katakana & Kanji Multi-Reading Trainer (101 Signs)



Japanese Katakana & Kanji Multi-Reading Trainer (101 Signs)

It features the **Multi-Reading Engine**, allowing users to enter **either Onyomi or Kunyomi readings** (e.g. typing **`sui`** or **`mizu`** for 水, **`nichi`**, **`hi`**, or **`bi`** for 日, or **`hito`**, **`jin`**, or **`nin`** for 人) and get marked correct.

Key Features of Version 9 Web Application

1. **Multi-Reading Answer Verification**: **`checkAnswer()`** splits comma-separated target readings (**`item.romaji.split(',')`**) and validates user input against **any** matching Onyomi or Kunyomi sound.
2. **Comprehensive Compound Hints**: Hints now highlight both standalone meanings and compound vocabulary (e.g. 日 $\rightarrow$ *Japan* (日本 / Nihon) | *Day* (日 / hi)).
3. **Complete 101 Character Coverage**: Covers 46 Basic Katakana, 25 Voiced Dakuten/Handakuten, and 30 N5 Kanji.

Multi-Reading Engine (v9)

The trainer now evaluates user input against a list of acceptable readings for each character. Typing either the Onyomi or Kunyomi reading gives a correct answer:

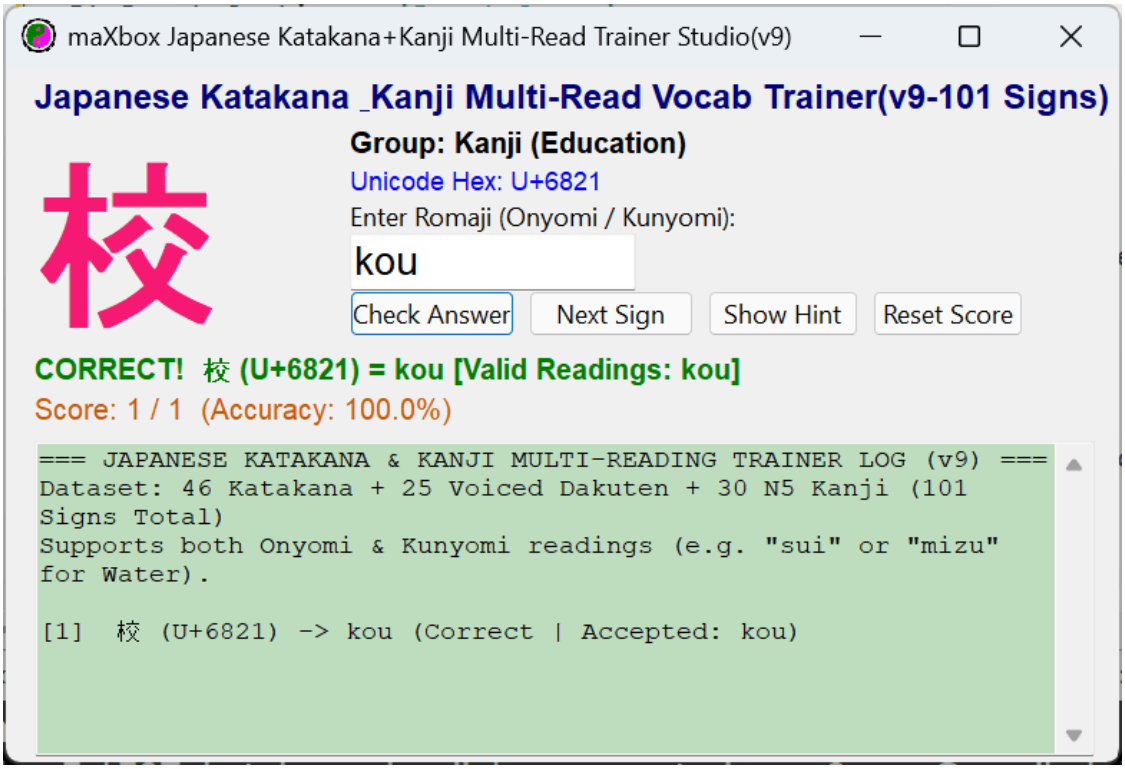
- 水: Accepts both `sui` (Onyomi) and `mizu` (Kunyomi).
- 日: Accepts `nichi`, `hi`, or `bi`.
- 月: Accepts `getsu`, `gatsu`, or `tsuki`.
- 人: Accepts `hito`, `jin`, or `nin`.
- 山: Accepts `yama`, `san`, or `zan`.

In maXbox, the answer verification function now parses comma-delimited reading lists:

```
functionMatchAnswer(constUserInput, AcceptedList: string): Bool
```

This function normalizes the target string list, appends a trailing comma for iteration, and extracts tokens one by one using `Pos`, `Copy`, and `Delete`:

```
function MatchAnswer(const UserInput, AcceptedList: string):
var
  List, Token: string;
  p: Integer;
begin
  Result:= False;
  List:= LowerCase(boldTrim(AcceptedList)) + ',';
  while Pos(',', List) > 0 do begin
    p:= Pos(',', List);
    Token:= boldTrim(Copy(List, 1, p - 1));
    Delete(List, 1, p);
    if (Length(Token) > 0) and (UserInput = Token) then begi
      Result:= True;
      Exit;
    end;
  end;
end;
```



https://sourceforge.net/projects/maxbox5/files/examples/1507_katakana_kanji_language_trainer-v9.pas/download

Advertisement



Career Growth

Unlock your leadership potential and navigate your career journey. Learn the pillars of development now!

Privacy Settings

To implement **multiple acceptable readings** in maXbox (Object Pascal), you use a **comma-delimited string parsing helper** that checks the user's input against a list of valid readings.

In your dataset array, store all valid readings (both Onyomi and Kunyomi) separated by commas:

```
Romaji:= 'sui, mizu';      // Accepts both "sui" and "mizu" for
Romaji:= 'nichi, hi, bi';  // Accepts "nichi", "hi", or "bi" fo
Romaji:= 'hito, jin, nin'; // Accepts "hito", "jin", or "nin" f
```

This CheckAnswer function pass the trimmed, lowercase user input and the dataset string to the above mentioned `MatchAnswer`() function:

```
1 procedure CheckAnswer(Sender: TObject);
```

```
var
  UserAns: string;
begin
  UserAns:= LowerCase(Trim(EdtAnswer.Text));
  if MatchAnswer(UserAns, Romaji[CurrentIdx]) then begin
    LblResult.Caption:= 'CORRECT! ' + Symbols[CurrentIdx] +
    LblResult.Font.Color:= clGreen;
    Inc(CorrectCount);
  end else begin
    LblResult.Caption:= 'INCORRECT! Accepted: ' + Romaji[Curr
    LblResult.Font.Color:= clRed;
  end;
  Inc(TotalAttempts);
  UpdateScoreDisplay;
end;
```

Would you like to extend MatchAnswer to automatically strip Japanese punctuation (like full stops 。) or ignore whitespace variations? Take this task.

Advertisement



Career Growth

Unlock your leadership potential and navigate your career journey. Learn the pillars of development now!

Privacy Settings



In maXbox Desktop or JavaScript Online available

30 N5 Kanjis

```
// --- 30 N5 FUNDAMENTAL KANJI SIGNS ---  {char: '日',hex:'65E5
```

```
// --- 30 N5 FUNDAMENTAL KANJI SIGNS ---
  {chr: '日',hex:'65E5',romaji: 'nichi',(Nature/Days)','Sun/
  {chr: '月',hex:'6708',getsu',(Nature/Days)','Moon/Monday (
  {chr: '火',hex:'706B','ka',(Elements/Days)','Fire / Tuesday
  {chr: '水',hex:'6C34','sui',(Elements/Days)','Water/Wednesd
  {chr: '木',hex:'6728','moku',(Elements Days)','Tree/Thursda
  {chr: '金',hex:'91D1','kin',(Elements/Days)','Gold/Friday (
  {chr: '土',hex:'571F','do',(Elements/Days)','Soil/Saturday
  {chr: '山',hex:'5C71','yama',(Nature)','Mt. Fuji (富士山 /
  {chr: '川',hex:'5DDD','kawa',(Nature)','Stream / River (小
  {chr: '田',hex:'7530','ta',(Nature)','Rice Field (水田 / Su
  {chr: '人',hex:'4EBA','hito',(People)','Japanese Person (日
  {chr: '口',hex:'53E3','kuchi',(Body Parts)',Entrance (入口
  {chr: '目',hex:'76EE','me',(Body Parts)','Purpose / Goal (
  {chr: '手',hex:'624B','te',(Body Parts)','Skillful (上手 /
  {chr: '足',hex:'8DB3','ashi',(Body Parts)','Excursion (遠足
  {chr: '力',hex:'529B','chikara',(Body/Mind)','Cooperation (
  {chr: '女',hex:'5973','onna',(People)','Female (女性 / Jose
  {chr: '男',hex:'7537','otoko',(People)','Male (男性 / Danse
  {chr: '子',hex:'5B50','ko',(People)','Child (子供 / Kodomo)
  {chr: '大',hex:'5927','dai',(Size/Direction)','University (
  {chr: '小',hex:'5C0F','shou',(Size Direction)','Elem. Stud.
  {chr: '中',hex:'4E2D','chuu',(Size/Direction)',China (中国
  {chr: '上',hex:'4E0A','ue',(Size/Direction)','Good at (上手
  {chr: '下',hex:'4E0B','shita',(Size/Direction)','Subway (地
  {chr: '年',hex:'5E74','nen',(Time)','This Year (今年 / Koto
  {chr: '先',hex:'5148','sen',(Time/People)','Teacher (先生 /
  {chr: '生',hex:'751F','sei',(Life/Education)','Student (学生
  {chr: '学',hex:'5B66','gaku',(Education)','School (学校 / G
  {chr: '校',hex:'6821','kou',(Education)','High School (高校
  {chr: '本',hex:'672C','hon',(Educ/Origin)','Japanese Langua
  ];
```

Complete Source

https://sourceforge.net/projects/maxbox5/files/examples/1507_katakana_kanji_language_trainer-v9.pas/download

```
// maXbox5 / maXbox6 Script v9: Japanese Katakana & Kanji Mu
// Includes 46 Basic Katakana +  Voiced Dakuten/Handakuten +
// Supports multiple valid Kunyomi/Onyomi readings (e.g. 'su
// Uses direct WideChar Unicode Hex Literals (#$XXXX) for 10
// {$APPTYPE FORM} 🦋 ⛩️ Katakana Example ⛩️ ア 🦋 🖼️ 🪆

program Katakana_Kanji_Language_Trainer_Studio9;

const
  TOTAL_ITEMS = 101;

var
  Form1: TForm;
```