

Español.accent

Juan Carlos Walls

COLLABORATORS

	<i>TITLE :</i> Español.accent		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Juan Carlos Walls	February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Español.accent	1
1.1	español.accent para translator.library	1
1.2	Copyright del fichero español	1
1.3	Copyright de las librerías	2
1.4	Introducción a la nueva librería	2
1.5	Requerimientos	2
1.6	Instalación	3
1.7	Sobre español.accent	3
1.8	Que puede hacer español.accent	4
1.9	Que no puede hacer español.accent	4
1.10	Por que no puede acentuar correctamente	4
1.11	Por que no puede traducir el numero 10000	5
1.12	Como modificar las reglas para aumentar la capacidad numérica.	6
1.13	¿Por qué habla como un yanqui?	6
1.14	Caracteres ASCII extendidos	6
1.15	Autor	7
1.16	Autor de las librerías	7
1.17	Para un correcto funcionamiento de español.accent	8
1.18	Análisis exhaustivo de español.accent	8

Chapter 1

Español.accent

1.1 español.accent para translator.library

```
español.accent para translator.library - Version 42.1
15 de Junio, 1995

-=>> English users read a resume of this file in the README file <<=-
-=>> Programmers read the file programmers.readme <<=-

Copyright (c) 1995
    Juan Carlos Pérez Walls
    por el fichero de acento.
Copyright (c) 1995
    Francesco Devitt
    por la librería.

Español.accent
    Sobre el fichero de definición del español

Requerimientos

Instalación

Copyright y distribución de español.accent

Copyright y distribución de translator.library

translator library 42.1
    Sobre la nueva librería transductora
```

1.2 Copyright del fichero español

Los ficheros correspondientes a español.accent son copyright de Juan Carlos

Pérez Walls
y está permitida su libre distribución, especialmente su ↔
inclusión

en la próxima distribución de translator.library de
Francesco Devitt
;-)

1.3 Copyright de las librerías

La version 42.1 de translator library no es dominio público. El fuente no está disponible. La librería y los ficheros de acento son libremente distribuibles siempre que no se obtenga un provecho económico de ello. Los ficheros de acento pueden tener restricciones adicionales o separadas colocadas por sus autores. Note que los ficheros American y English se han derivado del trabajo del Naval Research Laboratory, USA -estos no pueden ser usados comercialmente.

1.4 Introducción a la nueva librería

Con versiones de Amiga OS anteriores a la 2.1 Commodore ↔ proporcionó software para transcribir texto a voz en los discos del Workbench. Esta librería reemplaza la librería transcriptora proporcionada por Commodore. La original transcribe texto a fonemas para ser usados por el narrator device. Es usada por programas que producen salida hablada como 'Say' y 'Term'. Desafortunadamente para los usuarios no americanos la librería original transcribe todo el texto como si fuera inglés americano. No maneja otros lenguajes o dialectos adecuadamente.

La versión 42.1 de la librería transcriptora es un sustituto directo y funciona con todo el software que actualmente usa el sistema de voz de Commodore. Con esta versión de la librería transcriptora el usuario puede especificar que lenguaje debe usar el transcriptor. La librería transcribe el texto más rápido que la original y, cuando se le proporciona un fichero de acento extraído de la librería original, con la misma calidad. No es difícil escribir ficheros de transcripción para la mayoría de los lenguajes - con la excepción de lenguajes como el inglés y el francés que tienen más excepciones que la mayoría. Espero que los usuarios de este sistema se motiven a crear ficheros de descripción de acentos para el/los lenguaje(s) que ellos hablen.

Pulsa aquí para saber como conseguir el paquete completo de translator.library

---->00<----

1.5 Requerimientos

Necesarios:

- narrator.device (proporcionado con las versiones del WB anteriores a 2.1).
- Kickstart 1.3 o superior.
- Algún programa que hable ;-)

No necesarios (pero que mejoran el funcionamiento):

- kickstart 2.0 o superior.
- procesador 68020 o superior.
- locale.library.
- Installer de Commodore (RIP) para el script de instalación. (proporcionado con el WB 2.1 o superior y disponible en AmiNet).

Todas las versiones del device parecen seguir funcionando con los sistemas 3.1. Desafortunadamente el narrador device pronuncia todo con un acento americano y no soporta algunos fonemas necesarios para pronunciar algunos lenguajes (incluido el Inglés). Pero por lo menos usando la nueva translator library la pronunciación será más cercana a lo que debería ser.

Nótese que el comando 'Say' proporcionado con Amiga OS 2.04 parece tener un fallo en el GUI cuando se usa con KingCON. Este causa un comportamiento extraño al actualizar la ventana. Digo esto porque con el lanzamiento de esta librería puede que sea la primera vez que uses el comando Say.

1.6 Instalación

Pulsa sobre el icono Install y sigue las instrucciones. Este instalará la librería y los ficheros asociados en tu disco del Workbench llamado 'SYS:'. Este puede ser un disco duro.

Si no tienes el programa Installer puedes hacerlo tu mismo. Basicamente esto es lo que tienes que hacer:

```
Copy "XXX_translator.library" TO Libs:translator.library
Copy español.accent TO LOCALE:Accents/
```

La versión de la librería a instalar depende de tu sistema. El directorio 'Libs/' contiene 3 ficheros librería:

- 'v33' está diseñada para sistemas con cualquier CPU y una versión de OS anterior a la 2.04 aunque funcionará en cualquier Amiga.
- 'v37' es para sistemas con una versión del OS 2.04 o superior y cualquier CPU.
- '020' es para sistemas con un 68020 or superior y Amiga OS 2.04 o superior.

Por alguna razón la versión 020 es mayor que la version v37. Pero *es* mucho más rápida.

1.7 Sobre español.accent

Español.accent es un fichero de acento que define las reglas fonéticas del español y permite el uso de translator.library para producir salida hablada en español desde tu AMIGA. ↔

Que puede hacer
Español.accent y

que no puede

.

Análisis exhaustivo de Español.accent

1.8 Que puede hacer español.accent

- Puede acentuar correctamente casi todas las palabras del lenguaje español. ↔

- Puede pronunciar cualquier cifra desde 0 hasta 9999. por ejemplo, si intruces el número 1969, tu AMIGA diría "Mil novecientos sesenta y nueve".

- Entiende las abreviaturas más comunes del español, así, si escribes "Sr. Pérez" tu AMIGA diría "Señor Pérez".

- Reconoce varias palabras extranjeras y las transcribe al idioma correspondiente, así, si escribes Commodore, Apple o Walls (ese soy yo:-), tu Amiga diría estas palabras en inglés y no en español.

- Entiende el carácter cedilla

'Ç'

que se usa en muchos dialectos de España, Portugal y Brasil, y también entienelos caracteres

Ñ, Á, É, Í, Ó, Ú y Ü

.

1.9 Que no puede hacer español.accent

- No puede acentuar correctamente las palabras agudas que no lleven tilde como ↔

"tapiz" que debe escribirse "tapíz" si se quiere que sea pronunciada correctamente.

¿Por qué no puede?

- No puede traducir correctamente números mayores de 9999. El usuario puede ↔

extender las reglas concernientes a los números para permitir que translator entienda números mayores (necesitas al menos 9 reglas por cada nuevo dígito, probablemente más).

¿Por qué no puede?

- No puede decir una frase sin parecer un yanqui con un terrible acento a lo ↔

"Far West".

¿Por qué no puede?

1.10 Por que no puede acentuar correctamente

Las reglas de transcripción definidas por
 Francesco Devitt
 para los
 ficheros de acento impiden que el Amiga pueda saber cuando una palabra que no
 lleva tilde es aguda o llana.

Nosotros podemos saberlo porque si es llana acabará en vocal n ó s.

El problema está en que no hay una manera de indicarle a translator library
 que averigüe si una palabra termina en una de estas letras.

De todos modos cuanto más pienso en ello, más convencido estoy de que debe de
 haber alguna manera de hacerlo con las reglas existentes pero se me escapa :-)

Si alguien lo consigue solucionar, no dude en ponerse en contacto
 conmigo
 o
 en actualizar este archivo en AmiNet.

1.11 Por que no puede traducir el numero 10000

Las reglas de transcripción definidas por
 Francesco Devitt
 para los
 ficheros de acento impiden que el Amiga pueda traducir correctamente una
 sucesión indefinida de dígitos.
 La implementación actual utiliza una regla distinta para cada posición del
 mismo número en la cadena numérica, como ejemplo, estas son las reglas para el
 número 4:

```
[40](dg~) = { cuarenta }
[4](dg)(dg~) = { cuarenta y }
[4](dg)(dg)(dg~) = { cuatrocientos }
[4](dg)(dg)(dg)(dg~) = { cuatromil }
[4] = { cuatro }
```

La tercera regla dice: si te encuentras un 4 seguido de 2 dígitos más y algo
 que no es un dígito, tradúcelo como cuatrocientos.

Como es fácil deducir, por cada nuevo dígito que se quiera añadir a la
 capacidad de reconocimiento exigirá una nueva regla para cada cifra, lo que
 supone 9 reglas más por cada potencia de diez. Yo no tenía ganas de seguir
 añadiendo más reglas pero el usuario es libre de ampliar todas las que
 necesite.

Si
 Francesco Devitt
 cambiase un poco las reglas de transcripción, habría
 una manera de aumentar facilmente la capacidad con solo una regla por cada
 potencia de mil!.
 ¿cómo?
 Si alguien lo consigue solucionar, no dude en ponerse en contacto
 conmigo
 o
 en actualizar este archivo en AmiNet.

1.12 Como modificar las reglas para aumentar la capacidad numérica.

Si observamos bien la manera de decir los números en el español ←
veremos que

estos se dividen en grupos de tres cifras separados por las palabras "mil", "millones", "mil", "billones", "mil", "trillones", etc.

Por ejemplo, si cogemos el número 123123 (ciento veintitrés mil ciento veintitrés), observamos que las tres primeras cifras se llaman igual que las tres últimas separándolas por la palabra mil.

Podríamos definir las mismas reglas que hay ahora añadiéndole al final que estas pueden ir seguidas de cualquier cantidad de grupos de tres cifras (incluída ninguna) y definir después otras reglas para generar las palabras "mil", "millón", etc.

Las reglas actuales permiten definir grupos de 3 cifras pero no permiten sustituir la cadena vacía por una palabra que es lo que necesitarían las reglas de "mil", "millon", etc.

La única solución que veo es utilizar comas para separar los tríos pero el problema es que uno suele escribir (sobre todo con un ordenador) las fechas y otros números como 1995 en vez de 1,995 que se presta a otras interpretaciones como "uno coma novecientos noventa y cinco"

Si alguien lo consigue solucionar, no dude en ponerse en contacto
conmigo
o
en actualizar este archivo en AmiNet.

1.13 ¿Por qué habla como un yanqui?

El problema se encuentra en el `narrator.device`. Este dispositivo ←
tiene

información sobre todos y cada uno de los fonemas del lenguaje Americano y por lo tanto es capaz de pronunciar este correctamente.

Pero la mayoría de los otros lenguajes existentes (incluído el Inglés no americano) poseen fonemas que no están definidos en el `narrator.device`.

Actualmente la única solución es utilizar los fonemas americanos que más se parezcan al sonido que queremos producir, lo que inevitablemente hará que el Amiga hable con un acento marcadamente americano en cualquier otro idioma.

La solución ideal sería que alguien modificara el `narrator.device` o creara uno nuevo que reconociera fonemas de varias lenguas.

Asimismo, la `translator.library` debería modificarse para incluir las definiciones de estos nuevos fonemas.

Si alguien lo consigue solucionar, no dude en ponerse en contacto
conmigo
o
en actualizar este archivo en AmiNet.

1.14 Caracteres ASCII extendidos

Hay un problema con SPEAK-HANDLER versión 37.4. Parece que filtra ←
el código
ASCII extendido y por lo tanto palabras como "cigüeña" o "heiβ" (caliente en alemán) son transformadas en "cig e a" y "hei " eliminando los caracteres mayores de 127 antes de enviar la cadena a translator library.
Esto no es un problema de translator library ni de español.accent sino del speak-handler por lo que haría falta que alguien creara una nueva versión de este o por lo menos lo parcheara.

Si alguien lo consigue solucionar, no dude en ponerse en contacto
conmigo
o
en actualizar este archivo en AmiNet.

1.15 Autor

Este fichero de definición de acento ha sido realizado por:

Juan Carlos Pérez Walls
Estudiante en el Centro Superior de Informática de la Universidad de La Laguna.

Dirección:
Calle Eliseo López #1-11-A
38010 Santa Cruz de Tenerife
Islas Canarias, ESPAÑA

E-mail:
Sorry, no e-mail, yet...

Teléfono:
+(922) 64-03-67

ATENCIÓN:

-----Para que este fichero de acento funcione correctamente-----

1.16 Autor de las librerías

La translator library versión 42 ha sido creada por:

Francesco Devitt

E-mail:
ffranc@comp.vuw.ac.nz

Teléfono:
+64 4 388-3215

translator library está disponible en Aminet:
util/libs/translator42.lha

Home page:

<http://www.comp.vuw.ac.nz/~ffranc/amiga/translator.html>

1.17 Para un correcto funcionamiento de español.accent

Para que el fichero de acento funcione correctamente, es \leftrightarrow
 IMPRESCINDIBLE que
 el usuario envíe al
 autor
 una bonita postal de su país :-)

1.18 Análisis exhaustivo de español.accent

Este documento es similar al fichero español.accent pero los comentarios han sido añadidos siguiendo la sintaxis del lenguaje C (translator.library v42 no permite comentarios de fin de línea).

```
%class dg 0 1 2 3 4 5 6 7 8 9 /* los dígitos */
%class cero 0 /* el cero para casos en que no se pronuncia */
%class cons b c d f g h j k l m n ñ p q r s t v w x y z /* las consonantes */
%class ei e i é í /* las vocales e i para las excepciones de c, g y qu */
%class cg c g /* c g para palabras acabadas en cua o gua */
%class nls n l s /* n l s para la excepción de la r */
```

```
%stress -2 /* por defecto, las palabras sin tilde son llanas */
%emphasis3 /* grado de énfasis en el acento */
```

```
[APPLE] = AE3PUL /* La conocida firma de ordenadores */
[Á] = AA4 /* A con tilde */
[AI] = AY /* diptongo AI */
[A] = AA /* A por defecto */
```

```
$(cons*)[B](cons*)$ = { be } /* B aislada */
[B] = B /* B por defecto */
```

```
[COMMODORE] = KAA3MAXDOHR /* R.I.P. */
$(cons*)[Ç](cons*)$ = { cedilla } /* Ç aislada */
$(cons*)[C](cons*)$ = { ce } /* C aislada */
[Ç] = TH /* Ç por defecto */
[CH] = CH /* CH por defecto */
[C](ei) = TH /* C antes de E I */
[C] = K /* C por defecto */
```

```
$(DR.)$ = { doctor } /* abreviatura */
$(D.)$ = { don } /* abreviatura */
$(DÑA.)$ = { doña } /* abreviatura */
[DISKETTE] = { diskét } /* palabra de origen inglés */
$(cons*)[D](cons*)$ = { de } /* D aislada */
[D] = DH /* D por defecto */
```

```
$(ETC)$ = { etcétera } /* abreviatura */
```

```

[É] = EH4           /* E con tilde */
[EI] = EY           /* diptongo EI */
[E] = EH            /* E por defecto */

$(FT.)$ = { pies } /* abreviatura inglesa */
$(cons*) [F] (cons*)$ = { efe } /* F aislada */
[F] = FF           /* F por defecto */

$(cons*) [G] (cons*)$ = { ge } /* G aislada */
[G] (ei) = /H      /* G antes de E I */
[G]Ü(ei) = G       /* G antes de ÜE ÜI */
[GU] (ei) = G      /* G antes de UE UI */
[G] = G            /* G por defecto */

$(cons*) [H] (cons*)$ = { hache } /* H aislada */
[H]UE = G          /* caso especial: HUE- se pronuncia GÜE- */
[H] =              /* H por defecto (muda) */

[Í] = IY4          /* I con tilde */
[IA]s$ = `IYAA     /* caso especial: -IAS */
[IE]s$ = `IYEH     /* caso especial: -IES */
[IO]s$ = `IYOH     /* caso especial: -IOS */
[IA]$ = `IYAA      /* diptongo IA */
[IE]$ = `IYEH      /* diptongo IE */
[IO]$ = `IYOH      /* diptongo IO */
[I] = IY           /* I por defecto */

$(cons*) [J] (cons*)$ = { jota } /* J aislada */
[J] = /H           /* J por defecto */

$(cons*) [K] (cons*)$ = { ka } /* K aislada */
[K] = K            /* K por defecto */

$[LB.]$ = { libras } /* abreviatura */
$[LOG]$ = { logaritmo de } /* abreviatura */
$[LN]$ = { logaritmo natural de } /* abreviatura */
$(cons*) [LL] (cons*)$ = { elle } /* LL aislada */
$(cons*) [L] (cons*)$ = { ele } /* L aislada */
[LL] = YY         /* LL por defecto */
[L] = L           /* L por defecto */

[MACINTOSH] = MAE5KINTAA1SH /* conocido modelo de ordenador */
$[MR.]$ = MIH2STER /* abreviatura inglesa */
$[MS.]$ = MIH2Z /* abreviatura inglesa */
$[MRS.]$ = MIH2SIXZ /* abreviatura inglesa */
$[MI.]$ = { millas } /* abreviatura */
$(cons*) [M] (cons*)$ = { eme } /* M aislada */
[M] = M           /* M por defecto */

$(cons*) [Ñ] (cons*)$ = { eñe } /* Ñ aislada */
[Ñ] = NY          /* Ñ por defecto */
$(cons*) [N] (cons*)$ = { ene } /* N aislada */
[NG]$ = NX        /* unión NG */
[N] = N           /* N por defecto */

$[OZ.]$ = { onzas } /* abreviatura */
[OO] = UH4        /* OO como U ya que no existe en español */

```

```

$[OK]$ = OWKEY3      /* abreviatura inglesa */
$[O.K.]$ = OWKEY3   /* abreviatura inglesa */
[Ó] = AO4           /* O con tilde */
[O] = AO            /* O por defecto */

$[PTS.]$ = { pesetas } /* abreviatura */
$[PROF.]$ = { profesór } /* abreviatura */
$(cons*) [P] (cons*)$ = { pe } /* P aislada */
$[P]S = /* caso especial: PS- (P muda) */
[PH] = FF /* PH como F ya que no existe en español */
[P] = P /* P por defecto */

$(cons*) [Q] (cons*)$ = { ku } /* Q aislada */
[QU] (ei) = K /* QU antes de E I */
[Q] = K /* Q por defecto */

$(cons*) [R] (cons*)$ = { erre } /* R aislada */
$[R] = RR /* RR por defecto */
(nls) [R] = RR /* R antes de N L S */
[R] = R /* R por defecto */

$[ST.]$ = STRIY2T /* abreviatura inglesa */
$[SR.]$ = { señor } /* abreviatura */
$[SRA.]$ = { señora } /* abreviatura */
$[SRTA.]$ = { señorita } /* abreviatura */
$(cons*) [S] (cons*)$ = { ese } /* S aislada */
[SCH] = SH /* SCH como SH (aleman) */
[SH] = SH /* unión SH */
$[S]T = { es } /* ST- en palabras de origen inglés */
$[S]P = { es } /* SP- en palabras de origen inglés */
[S] = S /* S por defecto */

$(cons*) [T] (cons*)$ = { te } /* T aislada */
[TR] = TTRX /* unión TR */
[T] = TT /* T por defecto */

$[U.S.A.]$ = YUW3EHSEY1 /* abreviatura inglesa */
G[ÛE]$ = `UHEH /* ÛE después de G */
G[ÛI]$ = `UHIY /* ÛI después de G */
[Ú] = UH4 /* U con tilde */
(cg) [UA]$ = `UHAA /* caso especial: -CUA -GUA */
C[UE]s$ = `UHEH /* caso especial: -CUES */
C[UO]s$ = `WHAO /* caso especial: -CUOS */
C[UE]$ = `UHEH /* caso especial: -CUE */
C[UO]$ = `WHAO /* caso especial: -CUO */
[U] = UH /* U por defecto */

$(cons*) [V] (cons*)$ = { uve } /* V aislada */
[V] = V /* V por defecto */

$(cons*) [W] (cons*)$ = { doble v } /* W aislada */
[WALLS] = WAO4LZ /* mi nombre :- ) */
[W] = V /* W por defecto */

(dg) [X] (dg) = { por } /* signo de multiplicación */
(dg) [X] (dg) = { por } /* idem */
$(cons*) [X] (cons*)$ = { ekis } /* X aislada */

```

```

$[X] = Z                /* X al principio */
[X] = KS                /* X por defecto */

$(cons*)[Y](cons*)$ = " IY " /* Y aislada */
[Y]$ = IY              /* Y al final */
[Y](cons) = Y         /* Y antes de consonante */
[Y] = `IY             /* Y por defecto */

$(cons*)[Z](cons*)$ = { zeta } /* Z aislada */
[Z] = TH              /* Z por defecto */

(dg~)(cero*)[0] = { cero } /* ceros a la izquierda */
(dg)[0](dg) =          /* cero entre 2 digitos (mudo) */
(dg)[0] =              /* cero al final (mudo) */
[0] = { cero }        /* cero por defecto */

[10](dg~) = { diez } /* reglas para los numero del 10 al 15 */
[11](dg~) = { once }
[12](dg~) = { doce }
[13](dg~) = { trece }
[14](dg~) = { catorce }
[15](dg~) = { quince }
[1](dg)(dg~) = { dieci } /* decena */
[100](dg~) = { cien } /* centena */
[1](dg)(dg)(dg~) = { ciento } /* centena y pico */
[1](dg)(dg)(dg)(dg~) = { mil } /* unidad de millar */
[1] = { uno } /* unidad */

[20](dg~) = { veinte } /* para el resto de los números, */
[2](dg)(dg~) = { veinti } /* las reglas son las mismas */
[2](dg)(dg)(dg~) = { doscientos }
[2](dg)(dg)(dg)(dg~) = { dosmil }
[2] = { dos }

[30](dg~) = { treinta }
[3](dg)(dg~) = { treinta y }
[3](dg)(dg)(dg~) = { trescientos }
[3](dg)(dg)(dg)(dg~) = { tresmil }
[3] = { tres }

[40](dg~) = { cuarenta }
[4](dg)(dg~) = { cuarenta y }
[4](dg)(dg)(dg~) = { cuatrocientos }
[4](dg)(dg)(dg)(dg~) = { cuatromil }
[4] = { cuatro }

[50](dg~) = { cincuenta }
[5](dg)(dg~) = { cincuenta y }
[5](dg)(dg)(dg~) = { quinientos }
[5](dg)(dg)(dg)(dg~) = { cincomil }
[5] = { cinco }

[60](dg~) = { sesenta }
[6](dg)(dg~) = { sesenta y }
[6](dg)(dg)(dg~) = { seiscientos }
[6](dg)(dg)(dg)(dg~) = { seismil }
[6] = { seis }

```

```
[70](dg~) = { setenta }
[7](dg)(dg~) = { setenta y }
[7](dg)(dg)(dg~) = { setecientos}
[7](dg)(dg)(dg)(dg~) = { sietemil }
[7] = { siete }
```

```
[80](dg~) = { ochenta }
[8](dg)(dg~) = { ochenta y }
[8](dg)(dg)(dg~) = { ochocientos}
[8](dg)(dg)(dg)(dg~) = { ochomil }
[8] = { ocho }
```

```
[90](dg~) = { noventa }
[9](dg)(dg~) = { noventa y }
[9](dg)(dg)(dg~) = { novecientos}
[9](dg)(dg)(dg)(dg~) = { nuevemil }
[9] = { nueve }
```

/* el resto de las reglas son comunes en todos los lenguajes */

```
[ " " ] = " "
```

```
[.](dg+) = { punto }
[,](dg+) = { coma }
(dg+)[/](dg+) = { partido por }
[/] = { barra }
[-] = { menos }
[+] = { más }
[=] = { igual a }
[>] = { mayor que }
[<] = { menor que }
[#] = { número }
[*] = { asterisco }
[@] = AA
[|] = AO
[^] = { elevado a }
```

```
[.]$ = .
[,] = ,
[\\$] = { dólares }
[£] = { libras }
[&] = IY
[! ]$ = .
[.] = " "
[%] = { por ciento }
[:] = ". "
[;] = .
[?] = ?
[ "\\(" ] = ,
[ "\\)" ] = ,
[ \\ ] = " "
[ "\\[" ] = " "
[{ ] = " "
[ } ] = " "
[ _ ] = " "
```

[^] = " "

["] = " "