Character encoding (Generation III)

The Generation III games use a proprietary **character encoding** to store text data. The Generation III encoding is greatly different from the encodings used in previous generations, with characters corresponding to different bytes. Versions of the games in different languages may use different encodings, some more different than others.

Some text strings are stored in fixed-length structures while others are stored in a block of text with separate strings simply terminated by 0xFF. In the large, variable-length blocks, usually another structure will have pointers to the appropriate string(s) within that block of text. In the fixed-length structures, strings are still terminated by 0xFF, but any remainder of the allotted space is padded out with 0x00.

Character sets

Every international game in Generation III (English, French, Italian, German, and Spanish games) contains two character sets: their native set and the Japanese set. The different international character sets are mostly identical save for a few regional differences.

For most text, the game's native character set is used, but if a Pokémon's origin language is Japanese, its nickname and its Original Trainer's name use the Japanese character set. The Japanese games only have the Japanese character set, but with the exception of `0xB8`, all inputtable text is identical between international and Japanese character sets.

Note that 0x00 in the following tables is a space (" "), not empty.

International

The table below shows the English character set in Pokémon Emerald. Some differences do exist between different revisions and games and between different languages, detailed afterward.

Characters on a white background are the only characters that can be input in names; `0xF1` - `0xF6` are only available for input in German games. Those on a light gray background may be used in other text strings (such as dialogue) depending on the language of the game. Characters on a dark gray background are unused values that mostly display as spaces in FireRed, LeafGreen, and Emerald; in Ruby and Sapphire, they are holdovers from the Japanese encoding. Characters

with a dotted underline differ between regions.

	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-		À	Á	Â	Ç	È	É	Ê	Ë	Ì		Î	Ï	Ò	Ó	Ô
1-	Œ	Ù	Ú	Û	Ñ	ß	à	á		ç	è	é	ê	ë	ì	
2-	î	ï	ò	ó	ô	œ	ù	ú	û	ñ	º	ª	er	&	+	
3-					Lv	=	;									
4-																
5-	▯	¿	¡	PK	MN	PO	Ké				Í	%	(	)		
6-									â							í
7-										⬆	⬇	⬅	➡	*	*	*
8-	*	*	*	*	e	<	>									
9-																
A-	re	0	1	2	3	4	5	6	7	8	9	!	?	.	-	•
B-	…	“	”	‘	’	♂	♀		,	×	/	A	B	C	D	E
C-	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
D-	V	W	X	Y	Z	a	b	c	d	e	f	g	h	i	j	k
E-	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	▶
F-	:	Ä	Ö	Ü	ä	ö	ü				*Control characters*					

`0x7D` - `0x83`, marked by asterisks (*) above, print spaces 1-7 pixels wide (in ascending order of the hex value). While `0xB0` is "…" in the main fonts of Emerald, FireRed, and LeafGreen, it still displays as ".." in certain other fonts that exist in the games - for example, the small font used on the party screen and the narrow font used in the Pokédex, bag, and stores.

Differences between games and revisions

In all revisions of Ruby and Sapphire, `0xB0` prints "..". In certain languages, `0x34`, `0x57` - `0x59`, and `0x64` also have differences in different games, as detailed in the section below.

All other differences concern unused character values.

In FireRed and LeafGreen, `0x50` and `0x7D` - `0x83` are not used and print as spaces like other unused characters.

In Ruby and Sapphire, many values print Japanese characters, holdovers from the original Japanese encoding. These include:

- All unused characters (on a dark gray background above)
- `0x50` and `0x7D` - `0x83`
- `0x36`, `0x84` - `0x86`, and `0xA0`, in version 1.0 of the English Ruby and Sapphire only

Regional differences

A few characters differ between regions, and among them are quotation marks. These can be input into names, which means a Pokémon with quotation marks in its nickname or OT name will display differently if traded to a game of a different region.

In the table below, the underscores (`_`) stand for spaces.

	English	Spanish	Italian	German	French
`0x34`	Lv	Nv[RS] Nv.[EFRLG]	L.	Lv.	N.
`0x57` - `0x59`			_ _ M[RS] _ _ _[EFRLG]		
`0x5E` - `0x63`					
`0x64`					Pco[E]
`0xB1`		″		″	«
`0xB2`		″		″	»

Japanese

Only the characters on a white background below can be input in names. The characters on a dark gray background are printed as spaces in Pokémon FireRed, LeafGreen, and Emerald. Otherwise, the Japanese character set has no differences between games or revisions.

	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-		あ	い	う	え	お	か	き	く	け	こ	さ	し	す	せ	そ
1-	た	ち	つ	て	と	な	に	ぬ	ね	の	は	ひ	ふ	へ	ほ	ま
2-	み	む	め	も	や	ゆ	よ	ら	り	る	れ	ろ	わ	を	ん	ぁ
3-	ぃ	ぅ	ぇ	ぉ	ゃ	ゅ	ょ	が	ぎ	ぐ	げ	ご	ざ	じ	ず	ぜ
4-	ぞ	だ	ぢ	づ	で	ど	ば	び	ぶ	べ	ぼ	ぱ	ぴ	ぷ	ぺ	ぽ
5-	っ	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
6-	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
7-	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ヲ	ン	ァ
8-	ィ	ゥ	ェ	ォ	ャ	ュ	ョ	ガ	ギ	グ	ゲ	ゴ	ザ	ジ	ズ	ゼ
9-	ゾ	ダ	ヂ	ヅ	デ	ド	バ	ビ	ブ	ベ	ボ	パ	ピ	プ	ペ	ポ
A-	ッ	0	1	2	3	4	5	6	7	8	9	！	？	。	ー	・
B-	‥	『	』	「	」	♂	♀	円	.	×	/	A	B	C	D	E
C-	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
D-	V	W	X	Y	Z	a	b	c	d	e	f	g	h	i	j	k
E-	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	▶
F-	:	Ä	Ö	Ü	ä	ö	ü	⬆	⬇	⬅	*Control characters*					

Control characters

- 0xFA and 0xFB both mark a prompt for the player to press a button to continue the dialogue. However, they will print the new line of dialogue differently: 0xFA will scroll the previous dialogue up one line before printing the next line, while 0xFB will clear the dialogue box entirely.
- 0xFC is an escape character that leads to several different functions (see below).
- 0xFD is an escape character for variables, such as the player's name or a Pokémon's name (see below).
- 0xFE is a line break.
- 0xFF is a terminator, marking the ends of strings.

0xFC functions

When 0xFC is followed by...

- 0x01, it will change the color of the text, depending on the byte following. The available colors are listed below.
- 0x02, the text will be highlighted, depending on the byte following. The available colors are listed below.
- 0x03, the text's shadow will have its color changed, depending on the byte following. The available colors are listed below.
- 0x04, the text will be colored and highlighted. The byte immediately following determines the text's color, while a second byte afterward will determine the highlight color. The available colors are listed below.
- 0x06, the text will change size, depending on the byte following. 0x00 will make the font smaller, while anything else will make the font the default size.
- 0x08 and another byte, it produces a pause in the text. The byte after 0x08 determines the length of the pause.
- 0x09, the game will pause text display, and resume upon pressing a button.
- 0x0C, it will escape the byte that follows 0x0C if it is a control character and print a new character. If the second byte after 0xFC is not a control character byte, that byte prints normally.
  - When the third byte is 0xFA, "➡" is produced.
  - When the third byte is 0xFB, "+" is produced (though in the Japanese games, within the Options screen, it produces "=").
  - The other control characters do not produce any characters. In the English games, nothing is printed, while in the Japanese games, miscellaneous data appears to be printed.
- 0x0D, the text will be shifted by a certain amount of pixels, depending on the byte following this one. The effect wears off upon entering a new line.
- 0x10, music will begin to play. Music is specified by the two bytes following, in little endian format.
- 0x15, text will be rendered in the Japanese font
- 0x16, text will be rendered in the International font
- 0x17, music will be paused.

- 0x18, music will resume playing.

Color values

A table of available text, highlight, and shadow colors is shown below.

Byte	RS	FRLG	E
0x00	Transparent		
0x01			
0x02			
0x03			
0x04			
0x05			
0x06			
0x07			
0x08			
0x09			
0x0A			
0x0B			
0x0C			
0x0D			
0x0E			
0x0F			

0xFD variables

When 0xFD is followed by the bytes below, the following variables are printed.

- 0x01: the player's name
- 0x02, 0x03, or 0x04: whatever text has been assigned to one of three buffers using a variety of script commands
- 0x06: the rival's name
- 0x07: the game's name[RSE]
- 0x08: the name of the villainous team[RSE]
- 0x09: the name of the non-villainous team[RSE]

- 0x0A: the name of the villainous team's leader[RSE]
- 0x0B: the name of the non-villainous team's leader[RSE]
- 0x0C: the name of the villainous team's legendary Pokémon[RSE]
- 0x0D: the name of the opposing legendary Pokémon[RSE]

Trivia

- In the name field for Eggs, the game places the bytes `0x60 0x6F 0x8B` corresponding to タマゴ (*tamago*, the Japanese word for egg). This remains in the English version even though the characters have been replaced.

Data structure in the Pokémon games

Generation I	Pokémon species • Pokémon • Poké Mart • Character encoding • Save
Generation II	Pokémon species • Pokémon • Trainer • Character encoding • Save
Generation III	Pokémon species (Pokémon evolution • Pokédex • Type chart) Pokémon (substructures) • Move • Contest • Contest move • Item Trainer Tower • Battle Frontier • Character encoding • Save
Generation IV	Pokémon • Save
TCG GB and GB2	Character encoding

This data structure article is part of **Project Games**, a Bulbapedia project that aims to write comprehensive articles on the Pokémon games.