¿Os ha pasado alguna vez?

```
UnicodeDecodeError: 'ascii' codec
can't decode byte 0xc4 in position
10: ordinal not in range(128)
```

```
UnicodeEncodeError: 'ascii' codec
can't encode characters in position
3-8: ordinal not in range(128)
```

Fundamental

El texto plano NO EXISTE

(y no son los padres)

Encoding

Una cadena de texto no tiene sentido si no especificamos qué encoding se está usando para mostrarla

Encoding

Conjunto de reglas que asignan valores numéricos a cada carácter de texto

Problema en Python 2.x

Python 2.x usa por defecto encoding ASCII

(o sea, que no sabe qué hacer con los caracteres cuya representación es un byte mayor que 127)

Encoding ASCII

ASCII

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x																
1x																
2x		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x																
9x																
Ax																
Bx																
Cx																
Dx																
Ex																
Fx																

¿Qué hacemos?

Usar otro encoding (Python 2.x soporta más de 100)

Otros encodings

ISO 8859-1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x																
1x																
2x		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x																
9x																
Ax		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Otros encodings

Windows-1252

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x																
1x																
2x		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x	€		‚	ƒ	„	…	†	‡	ˆ	‰	Š	‹	Œ		Ž	
9x		‘	’	“	”	•	–	—	˜	™	š	›	œ		ž	Ÿ
Ax		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

¿Cuál elegir?

UTF-8

Longitud variable, soporte completo de **Unicode**

Unicode

Mapea caracteres con “code points”, no con enteros (1 - 127).

Hay 1.114.122 puntos de código (capaz de representar cualquier lenguaje)

Unicode

Sample Unicode

ℙƴ☂ɦøñ

U+2119: DOUBLE-STRUCK CAPITAL P
U+01B4: LATIN SMALL LETTER Y WITH HOOK
U+2602: UMBRELLA
U+210C: BLACK-LETTER CAPITAL H
U+00F8: LATIN SMALL LETTER O WITH STROKE
U+1F24: GREEK SMALL LETTER ETA WITH PSILI AND OXIA

Unicode

Ian Albert

22017x42807 px

Unicode y Python 2.x

(Python 2)

```
<type 'basestring'>
    |
    +--<type 'str'>
    |
    +--<type 'unicode'>
```

Unicode y Python 2.x

Str: secuencia de bytes

Unicode: secuencia de codepoints

Unicode y Python 2.x

Python intenta transformar texto en Unicode de manera automática

Unicode <-> Str

s.decode(*encoding*)

- `<type 'str'>` to `<type 'unicode'>`

u.encode(*encoding*)

- `<type 'unicode'>` to `<type 'str'>`

Cómo trabajar

- 1. Decode early
- 2. Unicode everywhere
- 3. Encode late

Más información

- http://www.joelonsoftware.com/articles/Unicode.html
- http://nedbatchelder.com/text/unipain/unipain.html
- http://farmdev.com/talks/unicode/