



bits



Binary	"Switch" value	"Human" Number (Decimal)
0 0	(OFF, OFF)	0
0 1	(OFF, ON)	1
1 0	(ON, OFF)	2
1 1	(ON, ON)	3

# Human "digits"

# X



## BYTES



### 8 bit BYTE

8 bit BYTE							
							
0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1
2	2x2	2x2x2	2 <sup>4</sup>	2 <sup>5</sup>	2 <sup>6</sup>	2 <sup>7</sup>	2 <sup>8</sup>
2	4	8	16	32	64	128	256

# Early Computing



# Early Computing

7bit BYTE							
							
0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1
2	2x2	2x2x2	2 <sup>4</sup>	2 <sup>5</sup>	2 <sup>6</sup>	2 <sup>7</sup>	Parity
2	4	8	16	32	64	128	Bit



# 128 points to fill

- 10 for numbers [0-9]
- 26 for lowercase letters [a-z]
- 26 for uppercase letters [A-Z]
- 32 for punctuation and miscellaneous
- Rest for “non-printable”  
[TAB], [BKSP], [DEL], [ESC]....etc.

## Early Computing

### ASCII

(pronounced “ASS-key”)

American Standard Code  
for Information  
Interchange

**Circa: 1963**

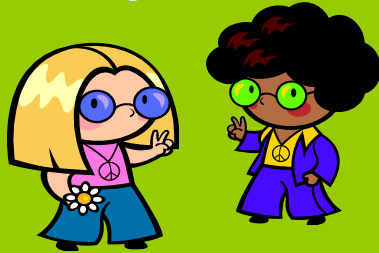
*(graph on right is the final  
1967 version which is still  
used today)*

code	0	1	2	3	4	5	6	7	8	9
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT
10	LF	VT	NP	CR	SO	SI	DLE	DC1	DC2	DC3
20	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS
30	RS	US	SP	!	"	#	\$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{		}	~	DEL		

# Early Computing

In the 60' & 70's  
everybody's feelin'

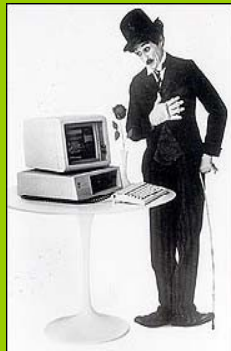
**groovy** about ASCII...



...except those who  
couldn't write in English!

# Early Computing

In the 80's data transmission became more  
reliable and the parity bit was freed up



*USELESS FACTOID:  
IBM released the first commercial  
8-bit system in 1981*

1st ... 7th bit	8th bit
ASCII	ASCII + 128 new points
128 code points	256 code points

# Now what to do?



128 new code points?!  
Yeah!  
Let's go nuts and add  
our languages!

## Incompatibility becomes the norm

EBDIC  
(European -  
IBM Standard)



JISCII  
(Japanese -  
7ビット及び8ビットの  
情報交換用符号化文字集合)

GOST  
(Russian -  
государственный  
стандарт)



CP 850  
(West European  
- MS-DOS 3.3)

...just a few of hundreds of national and  
proprietary code pages developed during the time

## Good news?

All code pages preserved ASCII in the first  
128 code points

## Bad news?

The new 128 non-ASCII points varied  
greatly among the “standards”

## ISO moves in...



Tries to create internationally  
recognized standards



# ISO Standard For W. Europe

## ISO-8859-1

(pronounced "EYE-so-eight- eight-five-nine-dash-one")

International Organization  
for Standardization –  
Standard 8859-1

Circa: 1987

The image shows a tilted representation of the ASCII character set table. The table is organized into rows and columns, with characters ranging from 0 to 255. The word "ASCII" is prominently displayed in the center of the table in a large, bold, black font.

## Standards are for wimps!



More or less base their early  
operating systems on the ISO drafts



Completely ignores ISO.  
Goes their own way.  
Ends up near bankruptcy by the 90's



- » ISO-8859-1 Latin1 (West European)
- » ISO-8859-2 Latin2 (East European)
- » ISO-8859-3 Latin3 (South European)
- » ISO-8859-4 Latin4 (North European)
- » ISO-8859-5 Cyrillic
- » ISO-8859-6 Arabic
- » ISO-8859-7 Greek
- » ISO-8859-8 Hebrew
- » ISO-8859-9 Latin5 (Turkish)
- » ISO-8859-10 Latin6 (Nordic)



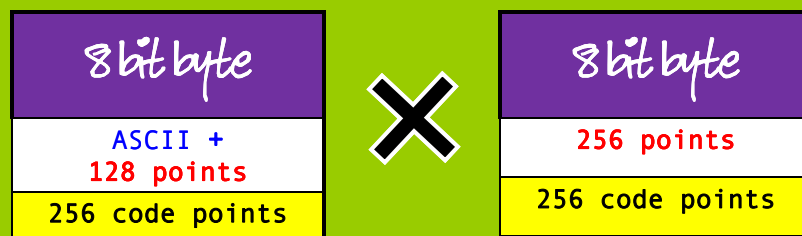
- » CP 1250 East European Latin
- » CP 1251 Cyrillic
- » CP 1252 West European Latin
- » CP 1253 Greek
- » CP 1254 Turkish
- » CP 1255 Hebrew
- » CP 1256 Arabic
- » CP 1257 Baltic
- » CP 1258 Vietnamese

Is there anyone  
missing?

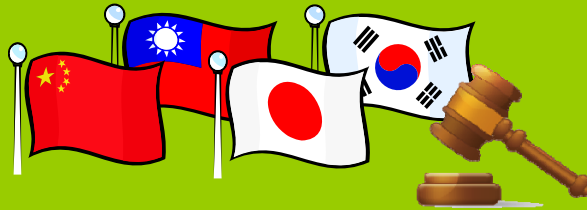


How do you turn  
256 code points  
into more than  
50,000?

Double byte



= 65,536 points



» China (Simplified Chinese)	GB-2312		CP 936
» Taiwan (Traditional Chinese)	Big5		CP 950
» Japan	Shift-JIS		CP 932
» Korea	KSC-5601		CP 949

## World Wide Wackiness

By the late 90's when the WWW  
is spreading rapidly we have:

<b>ISO</b>	> 10 separate code pages
<b>Windows</b>	> 18 separate code pages
<b>Mac</b>	> 14 separate code pages

*...and then dozens of national and  
government "standards"!*



The idea of a single, unified  
code page holding the  
scripts of all the world's  
languages was born in 1991

# Universal Code Page

## Unicode

(pronounced  
"YOU-nee-kohd")

Circa: 1996



## UTF-16

(for software)

## UTF-8

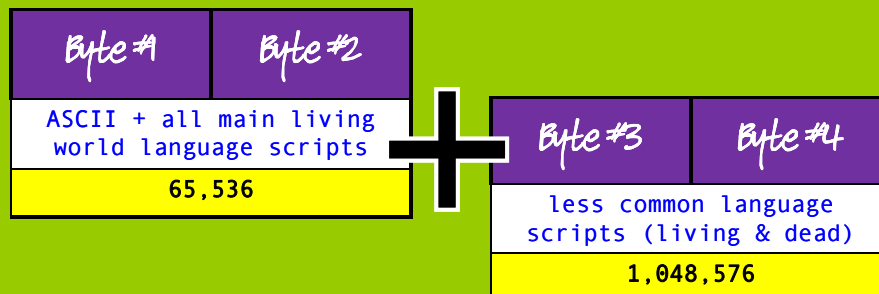
(for the web)



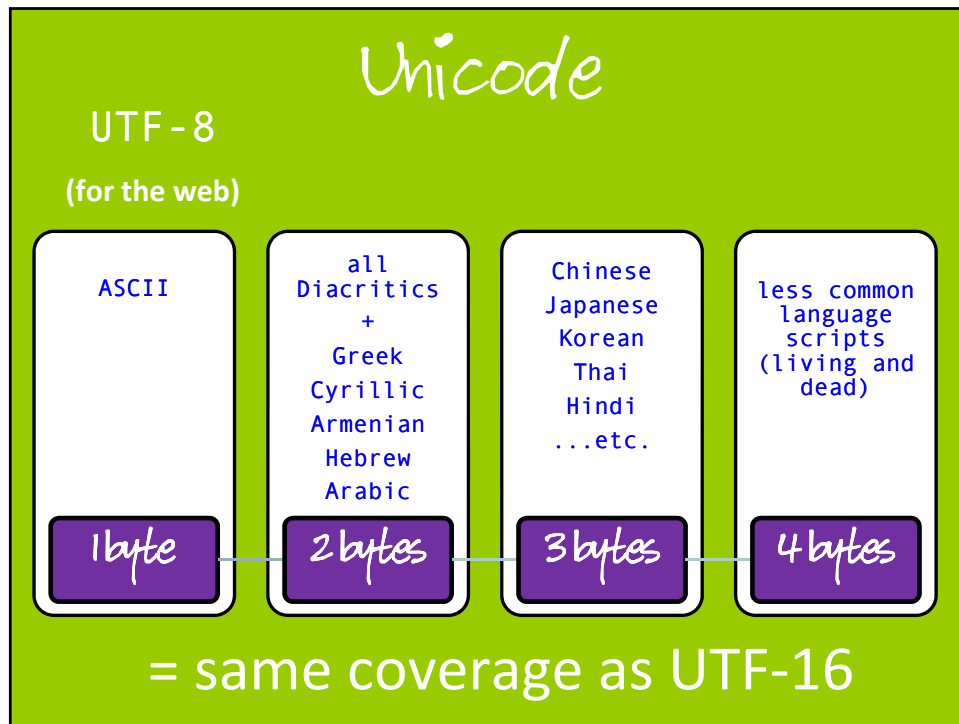
## Unicode

### UTF-16

(for software)



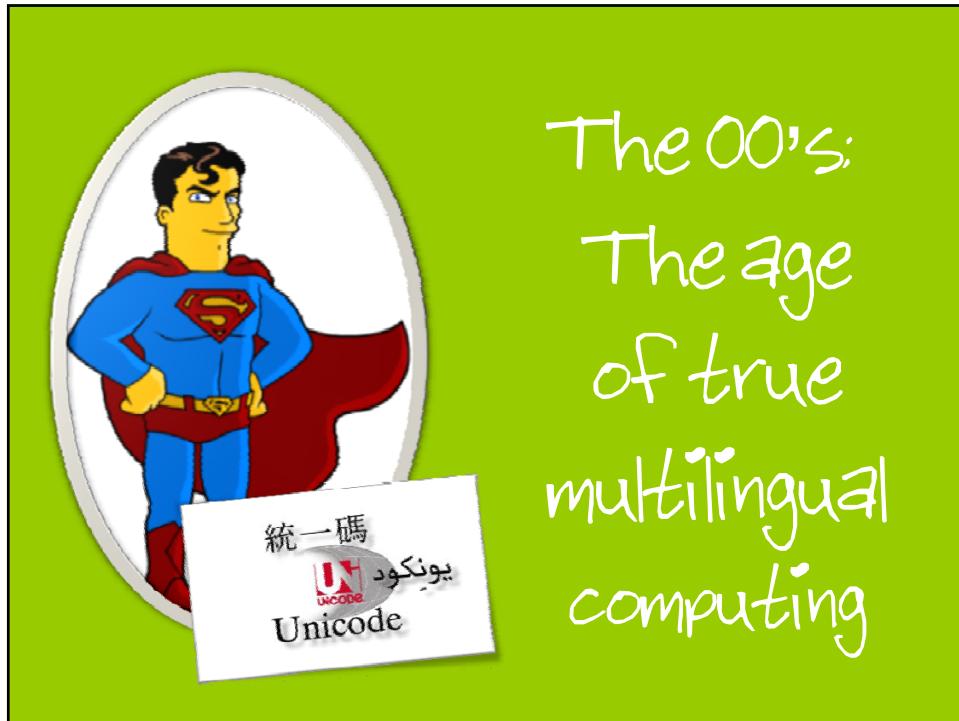
= 1,112,064 points



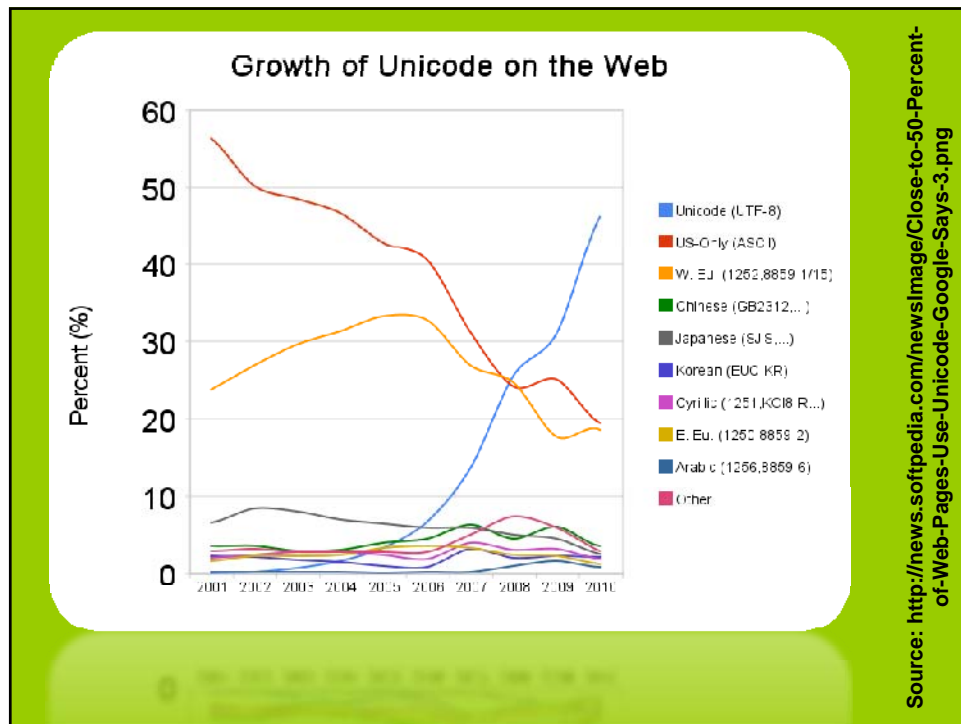
By mid 2000, Unicode is at  
the core of most  
mainstream software... this  
trend continues today







But, the web has been  
slower to catch on:  
In 2011, ~46% of websites  
are in Unicode (UTF8)



Have you encountered  
this?



## Identifying encoding in webpages

For *most* web pages,  
a META tag declares the encoding

```
<head>
```

```
...
```

```
<meta http-equiv="content-type"  
content="text/html; charset=utf-8">
```

```
...
```

```
</head>
```

The Internet browser reads the  
tag and “knows” what encoding  
to load the page in



