



Software Park Thailand  
</Code Camp>

# Outline

- ❑ วิทยาการคอมพิวเตอร์คืออะไร
- ❑ ไบนารี
- ❑ การแสดงผลข้อมูล
- ❑ อัลกอริทึม
- ❑ รหัสเทียม (Pseudo code)
- ❑ สแครช (Scratch)
- ❑ คำจำกัดความของเทคโนโลยี
- ❑ ทฤษฎี และการออกแบบ อัลกอริทึม
- ❑ สถาปัตยกรรมคอมพิวเตอร์

# การแสดงผลข้อมูล

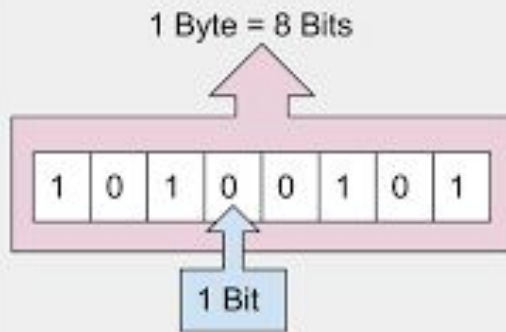
## Abstraction (ความคิดเชิงนามธรรม)

เป็นแนวคิดในวิทยาการคอมพิวเตอร์ที่มีการนำไปใช้ในระดัต่ำ (เช่นวิธีที่ข้อมูลถูกเก็บไว้ในรูปแบบไบนารีในท้ายที่สุด) หรือทำให้ง่ายขึ้นเพื่อให้ได้รับค่า  
ดังนั้น เราสามารถใช้การดำเนินการนั้นในระดับที่สูงขึ้น  
(เช่น การแสดงตัวอักษรที่เราสามารถใช้ในโปรแกรมของเรา)



Software Park Thailand  
</Code Camp>

# หน่วย ทาง Computer



1 byte	= 8 bits
1 kilobyte	= 1024 bytes
1 megabyte	= 1024 kilobyte
1 gigabyte	= 1024 megabyte
1 terabyte	= 1024 gigabyte

Created by: Frank Campt 2015  
License: Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License



Software Park Thailand  
</Code Camp>

# การแสดงผลข้อมูล ASCII

## มาตรฐาน ASCII



รหัสมาตรฐานของสหรัฐอเมริกาเพื่อการแลกเปลี่ยนสารสนเทศ

- เป็นรหัสอักขระที่ประกอบด้วยอักษรละติน เลขอารบิก เครื่องหมายวรรคตอน และสัญลักษณ์ต่างๆ โดยแต่ละรหัสจะแทนด้วยตัวอักขระหนึ่งตัว เช่น รหัส 65 (เลขฐานสิบ) ใช้แทนอักษรเอ (A) พิมพ์ใหญ่ เป็นต้น
- 1 ตัวอักษร = 1 byte

# ตัวอย่าง ASCII

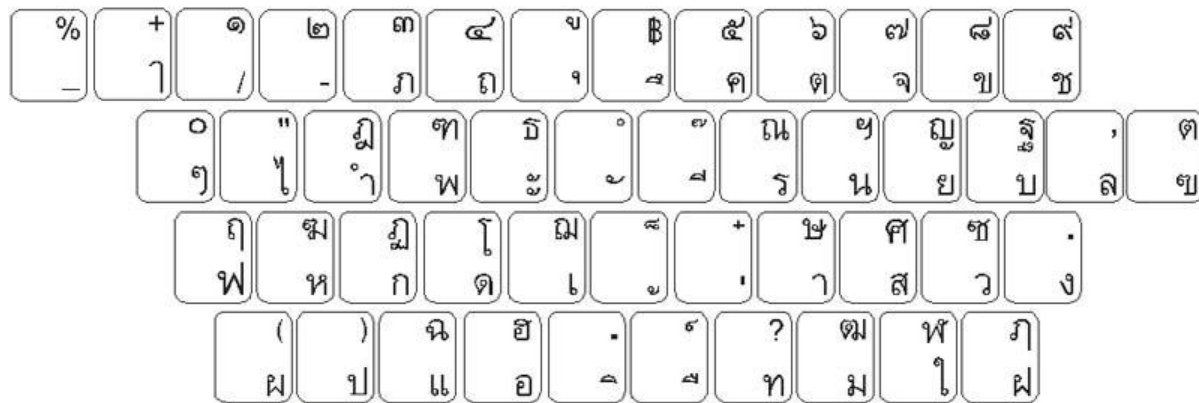
Dec	Hex	Oct	Binary	Char	Dec	Hex	Oct	Binary	Char	Dec	Hex	Oct	Binary	Char	Dec	Hex	Oct	Binary	Char
0	00	000	0000000	NUL (null character)	32	20	040	0100000	space	64	40	100	1000000	@	96	60	140	1100000	`
1	01	001	0000001	SOH (start of header)	33	21	041	0100001	!	65	41	101	1000001	A	97	61	141	1100001	a
2	02	002	0000010	STX (start of text)	34	22	042	0100010	"	66	42	102	1000010	B	98	62	142	1100010	b
3	03	003	0000011	ETX (end of text)	35	23	043	0100011	#	67	43	103	1000011	C	99	63	143	1100011	c
4	04	004	0000100	EOT (end of transmission)	36	24	044	0100100	\$	68	44	104	1000100	D	100	64	144	1100100	d
5	05	005	0000101	ENQ (enquiry)	37	25	045	0100101	%	69	45	105	1000101	E	101	65	145	1100101	e
6	06	006	0000110	ACK (acknowledge)	38	26	046	0100110	&	70	46	106	1000110	F	102	66	146	1100110	f
7	07	007	0000111	BEL (bell (ring))	39	27	047	0100111	'	71	47	107	1000111	G	103	67	147	1100111	g
8	08	010	0001000	BS (backspace)	40	28	050	0101000	(	72	48	110	1001000	H	104	68	150	1101000	h
9	09	011	0001001	HT (horizontal tab)	41	29	051	0101001	)	73	49	111	1001001	I	105	69	151	1101001	i
10	0A	012	0001010	LF (line feed)	42	2A	052	0101010	*	74	4A	112	1001010	J	106	6A	152	1101010	j
11	0B	013	0001011	VT (vertical tab)	43	2B	053	0101011	+	75	4B	113	1001011	K	107	6B	153	1101011	k
12	0C	014	0001100	FF (form feed)	44	2C	054	0101100	,	76	4C	114	1001100	L	108	6C	154	1101100	l
13	0D	015	0001101	CR (carriage return)	45	2D	055	0101101	-	77	4D	115	1001101	M	109	6D	155	1101101	m
14	0E	016	0001110	SO (shift out)	46	2E	056	0101110	.	78	4E	116	1001110	N	110	6E	156	1101110	n
15	0F	017	0001111	SI (shift in)	47	2F	057	0101111	/	79	4F	117	1001111	O	111	6F	157	1101111	o
16	10	020	0010000	DLE (data link escape)	48	30	060	0110000	0	80	50	120	1010000	P	112	70	160	1110000	p
17	11	021	0010001	DC1 (device control 1)	49	31	061	0110001	1	81	51	121	1010001	Q	113	71	161	1110001	q
18	12	022	0010010	DC2 (device control 2)	50	32	062	0110010	2	82	52	122	1010010	R	114	72	162	1110010	r
19	13	023	0010011	DC3 (device control 3)	51	33	063	0110011	3	83	53	123	1010011	S	115	73	163	1110011	s
20	14	024	0010100	DC4 (device control 4)	52	34	064	0110100	4	84	54	124	1010100	T	116	74	164	1110100	t
21	15	025	0010101	NAK (negative acknowledge)	53	35	065	0110101	5	85	55	125	1010101	U	117	75	165	1110101	u
22	16	026	0010110	SYN (synchronize)	54	36	066	0110110	6	86	56	126	1010110	V	118	76	166	1110110	v
23	17	027	0010111	ETB (end transmission block)	55	37	067	0110111	7	87	57	127	1010111	W	119	77	167	1110111	w
24	18	030	0011000	CAN (cancel)	56	38	070	0111000	8	88	58	130	1011000	X	120	78	170	1111000	x
25	19	031	0011001	EM (end of medium)	57	39	071	0111001	9	89	59	131	1011001	Y	121	79	171	1111001	y
26	1A	032	0011010	SUB (substitute)	58	3A	072	0111010	:	90	5A	132	1011010	Z	122	7A	172	1111010	z
27	1B	033	0011011	ESC (escape)	59	3B	073	0111011	;	91	5B	133	1011011	[	123	7B	173	1111011	{
28	1C	034	0011100	FS (file separator)	60	3C	074	0111100	<	92	5C	134	1011100	\	124	7C	174	1111100	
29	1D	035	0011101	GS (group separator)	61	3D	075	0111101	=	93	5D	135	1011101	]	125	7D	175	1111101	}
30	1E	036	0011110	RS (record separator)	62	3E	076	0111110	>	94	5E	136	1011110	^	126	7E	176	1111110	~
31	1F	037	0011111	US (unit separator)	63	3F	077	0111111	?	95	5F	137	1011111	_	127	7F	177	1111111	DEL



Software Park Thailand  
</Code Camp>

# การแสดงผลข้อมูล - ตัวอักษร

แล้วถ้ามีตัวอักษรหลายภาษาล่ะ ?



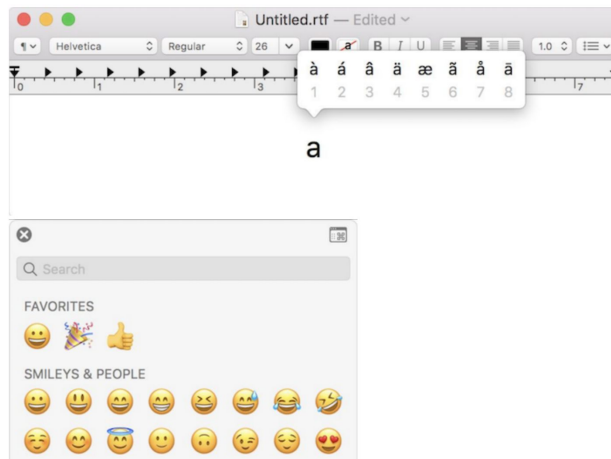
Thai (Kedmanee) Keyboard Layout



Software Park Thailand  
</Code Camp>

# การแสดงผลข้อมูล - ตัวอักษร

- จำเป็นต้องมีการเข้ารหัส (Encoding) เพื่อแก้ปัญหาดังกล่าว ให้ คอมพิวเตอร์สามารถแสดงตัวอักษรที่มีมาตรฐานนอกเหนือจาก ASCII
- แกรมบางที่ ในปัจจุบัน เรามี Emoji อีก ควรจะจัดการยังไงดี?





Software Park Thailand  
</Code Camp>

# character-set คืออะไร ทำไมต้องมี character set

- พยัญชนะภาษาอังกฤษ มี 26 ตัว คิดแยกเป็นตัวใหญ่ตัวเล็ก รวมตัวเลขและอักษรพิเศษต่างๆ แล้ว ก็ยังมีจำนวนไม่มาก (ไม่มากคือประมาณ 100 ตัว)
- ผู้ที่คิดค้นคอมพิวเตอร์ในยุคแรกๆ นั้นเป็นผู้ที่ใช้ภาษาอังกฤษ ดังนั้นจึงคิดค้น ระบบการเก็บข้อมูลโดยใช้รหัสยาว 7 บิต (ASCII) สามารถเก็บอักขระต่างๆ ได้ แตกต่างกันไปทั้งหมด  $2^7$  ตัว = 128 ตัว นับว่าเพียงพอสำหรับการใช้งานทั่วไป เช่น ตัวอักษร A จะตรงกับรหัส ASCII 65 เขียนเป็นเลขฐาน 2 เรียงกัน 7 ตัวได้ดังนี้ 100 0001
- ต่อมาคอมพิวเตอร์ได้แพร่หลายไปยังประเทศที่ไม่ได้ใช้แค่ภาษาอังกฤษ ได้มีวิธีการแก้ไขแบบเฉพาะหน้า โดยการเพิ่มความยาวของรหัสที่เก็บจาก 7 ตัว ให้เป็น 8 ตัว ก็จะสามารถเก็บอักขระได้ทั้งหมดเป็น  $2^8 = 256$  ตัว
- ประเทศต่างๆ ก็สามารถใช้ส่วนที่เกินมาอีก 128 ตัวนั้น ใช้เก็บรหัสอักขระเพิ่มเติม ที่เป็นเฉพาะของภาษาตนเอง เช่น กลุ่มประเทศยุโรปแถบตะวันตก ก็จะใช้เก็บอักขระเช่น อักขระที่มีลักษณะคล้าย a, e, i แต่มันจุดหรือมีขีดอยู่บนตัวอักษรเหล่านั้น (ซึ่งใช้กันในภาษาของพวกเขา) ตัวอย่างเช่น รหัส 228 จะเป็นตัว a umlaut คือ a แล้วมีจุดอยู่ข้างบน 2 จุด ใช้กันในภาษาเยอรมัน เป็นต้น
- ประเทศไทยก็ได้ใช้ประโยชน์ของรหัส 128 ตัวที่เกินมานี้เช่นกัน เช่น เราเก็บรหัส ก ไก่ ไว้ที่หมายเลข 161, เราเก็บไม้มลาย “ไ” ไว้ที่รหัสหมายเลข 228 เป็นต้น





Software Park Thailand  
 </Code Camp>

# ภาษาไทยเอาส่วนที่เกิน มาเก็บข้อมูลเพิ่ม

HEX DIGITS 1st → 2nd →	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0			๐	0	@	P	`	p				ฐ	ภ	ะ	เ	อ
-1			!	1	A	Q	a	q				ก	ท	ม	แ	อ
-2			"	2	B	R	b	r				ข	ฅ	ย	า	โ
-3			#	3	C	S	c	s				ข	ณ	ร	ำ	ใ
-4			\$	4	D	T	d	t				ค	ค	ฤ	ำ	ไ
-5			%	5	E	U	e	u				ค	ด	ล	ำ	ไ
-6			&	6	F	V	f	v				ข	ณ	ภ	ำ	อ
-7			'	7	G	W	g	w				ง	ก	ว	ำ	อ
-8			(	8	H	X	h	x				จ	อ	ค	ำ	อ
-9			)	9	I	Y	i	y				ฉ	น	ษ	ำ	อ
-A			*	:	J	Z	j	z				ช	บ	ส	ำ	อ
-B			+	:	K	[	k	{				ช	บ	ห	ำ	อ
-C			,	<	L	\	l					ณ	ณ	พิ	ำ	อ
-D			-	=	M	]	m	}				ณ	ณ	อ	ำ	อ
-E			.	>	N	^	n	~				ณ	พ	อ	ำ	อ
-F			/	?	O	_	o	~				ณ	พ	ำ	อ	อ



Software Park Thailand  
</Code Camp>

# character-set คืออะไร ทำไมต้องมี character set

- ดังที่กล่าวมาแล้วนั้น เราจำเป็นต้องมีตารางที่จะแสดงว่าหมายเลขอะไร หมายถึงตัวอักษรอะไร ตารางนี้ก็จะแตกต่างกันไปในแต่ละภาษา ซึ่งมีอักขระซึ่งต่างๆ กัน
- ดูๆ ไปแล้ว ก็ยังไม่เห็นว่าจะมีปัญหาอะไร ตราบใดที่ในเครื่องคอมพิวเตอร์ของท่าน มีใช้งาน แค่ 2 ภาษา เช่น ภาษาไทย กับ ภาษาอังกฤษ ถ้าท่านได้รับรหัสที่มีค่า 228 ก็สรุปได้เลยว่าผู้ส่งต้องการจะส่งไม้มลาย มา
- แต่หากท่านต้องการจะส่งข้อมูล ไปยังเครื่องคอมพิวเตอร์เครื่องอื่น หรือเปิด Web Site ให้ผู้คนจากทั่วโลกเข้ามาดู ที่ไม่ได้ใช้ภาษาไทยอยู่ด้วย ผู้คนเหล่านั้นจะเข้าใจได้อย่างไรว่า ท่านต้องการส่ง ไม้มลาย ซึ่งเป็นอักขระหนึ่งของภาษาไทย?
- ถ้าคนเยอรมันได้รับรหัสนั้น ย่อมต้องแปลความหมายของรหัส 228 ที่ท่าน ส่งไปว่าเป็นอักขระ a umlaut ที่พวกเขาใช้กัน? เขาควรจะรู้ได้อย่างไรว่า เขาควรจะหยิบตารางที่ใช้แปลงค่ารหัสเป็นภาษาไทย แทนที่จะใช้ตารางของภาษาเยอรมัน?
- วิธีแก้ปัญหานี้ก็ง่าย่ายดาย เพียงแต่ระบุปะหน้าไปว่าข้อมูลของท่านนั้น ขอให้ใช้ตารางสำหรับภาษาไทยในการถอดรหัส
- สิ่งทีระบุปะหน้าไปนั้น ก็เป็นที่เรียกกันว่า character set



Software Park Thailand  
</Code Camp>

# character-set สำคัญควรรู้

จึงเป็นที่มาของการเข้ารหัส

- TIS-620
- Windows-874
- Unicode
- UTF-8



Software Park Thailand  
</Code Camp>

# character-set อื่นๆ

ภาษาของประเทศแถบยุโรปตะวันตก iso-8859-1  
ภาษาของประเทศแถบยุโรปกลาง iso-8859-2  
ภาษาญี่ปุ่น ใช้ euc\_JP, iso-2022-jp และ Shift\_JIS



Software Park Thailand  
</Code Camp>

# การแสดงผลข้อมูล TIS-620

- มีเรียกอีกชื่อหนึ่งว่ารหัส มอก.620 หรือ รหัส สมอ. เป็นมาตรฐานรหัสตัวอักษรภาษาไทยที่กำหนดโดยสำนักงานมาตรฐานอุตสาหกรรม (สมอ.) โดยพัฒนาเพิ่มเติมจากมาตรฐาน ISO-646-1983 และ IBM GX20-1850-4 ซึ่งเป็นรหัสที่ใกล้เคียงกับ ASCII แบบ 7 bit[11] มาตรฐาน TIS-620 ออกมาครั้งแรกในปี พ.ศ. 2529 และมีการปรับปรุงเพิ่มเติมอีกครั้งหนึ่งในปี พ.ศ. 2533 ตัวอย่างตาราง TIS-620
- 1 ตัวอักษร = 1 byte

“รหัส 1 byte ที่รองรับภาษาไทย”



Software Park Thailand  
 </Code Camp>

# การแสดงผลข้อมูล TIS-620

HEX DIGITS 1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0			๐	๑	@	P	`	p				ส	ภ	ะ	เ	อ
-1			!	1	A	Q	a	q				ก	ฑ	ม	ั	แ
-2			"	2	B	R	b	r				ข	ฅ	ย	า	โ
-3			#	3	C	S	c	s				ช	ณ	ร	ำ	ใ
-4			\$	4	D	T	d	t				ค	ด	ถ	เ	ไ
-5			%	5	E	U	e	u				ค	ด	ล	า	เ
-6			&	6	F	V	f	v				ข	ถ	ภ	ั	า
-7			'	7	G	W	g	w				ง	ท	ว	า	จ
-8			(	8	H	X	h	x				จ	ธ	ศ	า	จ
-9			)	9	I	Y	i	y				ฉ	น	ษ	า	จ
-A			*	:	J	Z	j	z				ช	บ	ส	.	า
-B			+	;	K	[	k	{				ช	บ	ห	+	า
-C			<	L	\							ณ	ณ	ณ	.	า
-D			-	=	M	]	m	}				ณ	ณ	ณ	.	า
-E			.	>	N	^	n	~				ณ	ณ	ณ	.	า
-F			/	?	O	_	o	~				ณ	ณ	ณ	.	า



Software Park Thailand  
</Code Camp>

# การแสดงผลข้อมูล Windows-874

- Microsoft นำเอามาตรฐาน TIS-620 มาพัฒนาต่อ เพื่อใช้เป็นรูปแบบตัวอักษรภาษาไทย สำหรับการแลกเปลี่ยนข้อมูลในระบบปฏิบัติการ Windows ตัวมาตรฐาน Windows-874 นั้นถูกพัฒนาขึ้นมาโดยใช้พื้นฐานจาก TIS-620 จึงทำให้มีความเข้ากันได้กับไฟล์ที่ถูกบันทึกมาในมาตรฐาน TIS-620 แต่เนื่องจากว่าใน Windows-874 นั้นมีอักขระบางอย่างที่ถูกใส่เพิ่มขึ้นมาภายหลังซึ่งไม่มีใน TIS-620
- จึงอาจก่อให้เกิดปัญหาในกรณีที่บันทึกไฟล์โดยใช้มาตรฐาน Windows-874 แต่ไปเปิดอ่านโดยใช้มาตรฐาน TIS-620
- 1 ตัวอักษร = 1 byte





Software Park Thailand  
 </Code Camp>

# การแสดงผลข้อมูล Windows-874

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 0000	STX 0001	SOT 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
10	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
20	SP 0020	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL 007F
80	€ 20AC					...										
90		2018	2019	201C	201D	2022	2013	2014								
A0	NEBP 00A0	ก 0E01	ข 0E02	ฃ 0E03	ค 0E04	ฅ 0E05	ง 0E06	จ 0E07	ฉ 0E08	ช 0E09	ฌ 0E0A	ญ 0E0B	ฉ 0E0C	ญ 0E0D	ฎ 0E0E	ฏ 0E0F
B0	ฐ 0E10	ฑ 0E11	ฒ 0E12	ณ 0E13	ด 0E14	ต 0E15	ถ 0E16	ท 0E17	ธ 0E18	น 0E19	บ 0E1A	ป 0E1B	ผ 0E1C	ฝ 0E1D	พ 0E1E	ฟ 0E1F
C0	ภ 0E20	ม 0E21	ย 0E22	ร 0E23	ล 0E24	ว 0E25	ร 0E26	ศ 0E27	ษ 0E28	ส 0E29	ห 0E2A	ฬ 0E2B	ฬ 0E2C	อ 0E2D	ฮ 0E2E	ย 0E2F
D0	๕ 0E30	๖ 0E31	๗ 0E32	๘ 0E33	๙ 0E34	๐ 0E35	๑ 0E36	๒ 0E37	๓ 0E38	๔ 0E39	๕ 0E3A					฿ 0E3F
E0	๖ 0E40	๗ 0E41	๘ 0E42	๙ 0E43	๐ 0E44	๑ 0E45	๒ 0E46	๓ 0E47	๔ 0E48	๕ 0E49	๖ 0E4A	๗ 0E4B	๘ 0E4C	๙ 0E4D	๐ 0E4E	๑ 0E4F
F0	๐ 0E50	๑ 0E51	๒ 0E52	๓ 0E53	๔ 0E54	๕ 0E55	๖ 0E56	๗ 0E57	๘ 0E58	๙ 0E59	๐ 0E5A	๑ 0E5B				





# การแสดงผลข้อมูล - Unicode

**Unicode** ต่างจาก ASCII คือ ASCII เก็บ byte เดียว แต่ Unicode เก็บ 2 byte ซึ่งข้อมูล 2 byte เก็บข้อมูลได้มากมายมหาศาล สามารถเก็บข้อมูลได้มากมายหลายภาษาในโลกอย่างภาษาไทยก็อยู่ใน Unicode นี้ด้วยเหมือนกัน ดังนั้นรหัสภาษาไทยเอาไปเปิดในภาษาจีน ก็ยังเป็นภาษาไทยอยู่ ไม่ออกมาเป็นภาษาจีน เพราะว่ามี code ตายตัวอยู่ว่า code นี้จองไว้สำหรับภาษาไทย แล้ว code ตรงช่วงนั้นเป็นภาษาจีน ตรงโน้นเป็นภาษาญี่ปุ่น จะไม่ใช้ที่ซ้ำกันเป็นต้น



## การแสดงผลข้อมูล - UTF-8

- UTF-8 คือ รหัสภาษานานาชาติ หรือ Unicode เป็นการเข้ารหัสชุดอักขระที่ใช้ชุดข้อมูล **1 ถึง 4 byte** เพื่อแทนตัวอักษรเกือบทั้งโลก โดยใช้หลักการใช้ชุดข้อมูลแบบความยาวไม่คงที่ แทนตัวอักขระเป็นชุดๆ ไป
- UTF-8 มีข้อดีตรงที่ว่า ใน 1 หน้าเว็บนั้น เราสามารถแสดงผลรวมกันได้ ในหน้านั้นจะมีภาษาไทย อังกฤษ จีน ญี่ปุ่น ซาฮู สวีเดน ฯลฯ มันแสดงผลรวมกันได้เลยใน 1 หน้า แต่ถ้าเราเลือกเป็น TIS-620 ตรงภาษาอื่นๆ มันจะอ่านไม่ออกเพราะรหัสนี้มีแค่ ภาษาไทย กับ อังกฤษ



Software Park Thailand  
 </Code Camp>

# UTF-8 - Thai

<https://www.utf8-chartable.de/unicode-utf8-table.pl?start=3584&number=128&utf8=0x>

Unicode code point	character	UTF-8 (hex.)	name
U+0E00	□	0xe0 0xb8 0x80	
U+0E01	ก	0xe0 0xb8 0x81	THAI CHARACTER KO KAI
U+0E02	ข	0xe0 0xb8 0x82	THAI CHARACTER KHO KHAI
U+0E03	ฃ	0xe0 0xb8 0x83	THAI CHARACTER KHO KHUAT
U+0E04	ค	0xe0 0xb8 0x84	THAI CHARACTER KHO KHWAI
U+0E05	ฅ	0xe0 0xb8 0x85	THAI CHARACTER KHO KHON
U+0E06	ฆ	0xe0 0xb8 0x86	THAI CHARACTER KHO RAKHANG
U+0E07	ง	0xe0 0xb8 0x87	THAI CHARACTER NGO NGU
U+0E08	จ	0xe0 0xb8 0x88	THAI CHARACTER CHO CHAN
U+0E09	ฉ	0xe0 0xb8 0x89	THAI CHARACTER CHO CHING
U+0E0A	ช	0xe0 0xb8 0x8a	THAI CHARACTER CHO CHANG
U+0E0B	ซ	0xe0 0xb8 0x8b	THAI CHARACTER SO SO
U+0E0C	ฌ	0xe0 0xb8 0x8c	THAI CHARACTER CHO CHOE
U+0E0D	ญ	0xe0 0xb8 0x8d	THAI CHARACTER YO YING
U+0E0E	ฎ	0xe0 0xb8 0x8e	THAI CHARACTER DO CHADA
U+0E0F	ฏ	0xe0 0xb8 0x8f	THAI CHARACTER TO PATAK
U+0E10	ฐ	0xe0 0xb8 0x90	THAI CHARACTER THO THAN
U+0E11	ฑ	0xe0 0xb8 0x91	THAI CHARACTER THO NANGMONTHO
U+0E12	ฒ	0xe0 0xb8 0x92	THAI CHARACTER THO PHUTHAO
U+0E13	ณ	0xe0 0xb8 0x93	THAI CHARACTER NO NEN
U+0E14	ด	0xe0 0xb8 0x94	THAI CHARACTER DO DEK
U+0E15	ต	0xe0 0xb8 0x95	THAI CHARACTER TO TAO
U+0E16	ถ	0xe0 0xb8 0x96	THAI CHARACTER THO THUNG



Software Park Thailand  
</Code Camp>

# UTF-8 Emoticon

## 1. Emoticons ( 1F601 - 1F64F )

[Back to top](#)

Native <sup>[1]</sup>	Apple <sup>[2]</sup>	Android <sup>[3]</sup>	Android <sup>[3]</sup>	Symbola <sup>[4]</sup>	Twitter <sup>[5]</sup>	Unicode	Bytes (UTF-8)	Description
						U+1F601	\xF0\x9F\x98\x81	grinning face with smiling eyes
						U+1F602	\xF0\x9F\x98\x82	face with tears of joy
						U+1F603	\xF0\x9F\x98\x83	smiling face with open mouth
						U+1F604	\xF0\x9F\x98\x84	smiling face with open mouth and smiling eyes
						U+1F605	\xF0\x9F\x98\x85	smiling face with open mouth and cold sweat
						U+1F606	\xF0\x9F\x98\x86	smiling face with open mouth and tightly-closed eyes
						U+1F609	\xF0\x9F\x98\x89	winking face
						U+1F60A	\xF0\x9F\x98\x8A	smiling face with smiling eyes
						U+1F60B	\xF0\x9F\x98\x8B	face savouring delicious food
						U+1F60C	\xF0\x9F\x98\x8C	relieved face
						U+1F60D	\xF0\x9F\x98\x8D	smiling face with heart-shaped eyes
						U+1F60F	\xF0\x9F\x98\x8F	smirking face
						U+1F612	\xF0\x9F\x98\x92	unamused face
						U+1F613	\xF0\x9F\x98\x93	face with cold sweat
						U+1F614	\xF0\x9F\x98\x94	pensive face
						U+1F616	\xF0\x9F\x98\x96	confounded face

ทดสอบโดยการสร้างไฟล์ข้อความที่มีคำว่า “English,ภาษาไทย” บันทึกโดยใช้ Encoding ที่แตกต่างกันคือ ASCII (US), ANSI, TIS-620, Windows-874, UTF-8 และ Unicode เมื่อนำไฟล์ที่ได้มาเปิดดูข้อมูลในรูปแบบ Hexadecimal

0   45 6E 67 6C 69 73 68 2C-3F 3F 3F 3F 3F 3F 3F 0A	English,???????	ASCII-US
0   45 6E 67 6C 69 73 68 2C-C0 D2 C9 D2 E4 B7 C2 0A	English,ÀÒÉÒä-Â	TIS-620
0   45 6E 67 6C 69 73 68 2C-C0 D2 C9 D2 E4 B7 C2 0A	English,ÀÒÉÒä-Â	Windows-874
0   45 6E 67 6C 69 73 68 2C-C0 D2 C9 D2 E4 B7 C2	English,ÀÒÉÒä-Â	ANSI
00   45 6E 67 6C 69 73 68 2C-E0 B8 A0 E0 B8 B2 E0 B8	English,à, à, 'à,	UTF-8
10   A9 E0 B8 B2 E0 B9 84 E0-B8 97 E0 B8 A2 0A	@à, 'à¹ -à, -à, ¢	
00   FF FE 45 00 6E 00 67 00-6C 00 69 00 73 00 68 00	ÿþE·n·g·l·i·s·h·	Unicode
10   2C 00 20 0E 32 0E 29 0E-32 0E 44 0E 17 0E 22 0E	, · -2·) ·2·D·...·	



Software Park Thailand  
</Code Camp>

# ตัวอย่างระบบ ASCII , UTF-8 และอื่นๆ

Characters:		A	é	κ
character sets	ASCII	code points (hexadecimal) 65 (41)	n/a	n/a
	ISO-8859-1 (Latin 1)	65 (41)	233 (E9)	n/a
	Unicode (UCS)	65 (41)	233 (E9)	1488 (05D0)
character encodings	7-bit ASCII	hex byte values 41	n/a	n/a
	ISO-8859-1	41	E9	n/a
	UTF-8	41	C3 A9	D7 90
	UTF-16	00 41	00 E9	05 D0
	UTF-32	00 00 00 41	00 00 00 E9	00 00 05 D0





Software Park Thailand  
</Code Camp>

# การแสดงผลข้อมูล - ภาพ

- คอมพิวเตอร์ยังสามารถใช้ไบนารีเพื่อแสดงรูปภาพ ด้วยไบต์ 3 ไบท์ (1 byte = 8 bit)
- แต่ละอันแทน สีแดง (Red), เขียว(Green) และน้ำเงิน (Blue) หรือ RGB นั่นเอง



1 pixel = 3 bytes  
 $50 \times 50 \text{ pixels} = 50 \times 50 \times 3 = 7,500 \text{ bytes}$

- ค่าสีแดง, สีเขียว และสีน้ำเงิน ถูกรวมเข้าด้วยกันเพื่อให้ได้ สีเหลืองอ่อน :





Software Park Thailand  
</Code Camp>

# การแสดงผลข้อมูล - ภาพ

- ภาพแต่ละภาพประกอบด้วยพิกเซลหลายพัน หรือล้านพิกเซล หรือสี่เหลี่ยมสี ที่เราสามารถมองเห็นได้ เมื่อเราซูมเข้าไปมากพอ:







# การแสดงผลข้อมูล - ภาพ -> วิดีโอ

- และวิดีโอเป็นจำนวนมาก ภาพจำนวนมาก จะแสดงทีละภาพด้วยจำนวนเฟรมต่อวินาที
- สำหรับ iPhone รุ่นใหม่พีเจอร์ “แอนนิโมจิ” นั้นเป็นเพียงภาพจำนวนมากที่สร้าง และแสดงผลทีละภาพ :



- เราสามารถคิดว่าวิดีโอเป็น Abstractions เหนือรูปภาพ, รูปภาพเป็น Abstractions เหนือพิกเซล และ พิกเซลเป็น Abstractions บนบิต



Software Park Thailand  
</Code Camp>

# การแทนที่ Video ด้วยภาพ

