ภาษาคอมพิวเตอร์

ข้อมูลต่าง ๆ ที่ถูกป้อนเข้าสู่เครื่องคอมพิวเตอร์จะถูกแปลงเป็นสัญญาณดิจิทัล (สัญญาณไฟฟ้า) ด้วยรหัสเลขฐานสอง (binary number) ซึ่งประกอบด้วยเลข 1 และ 0 โดย 1 แทนสัญญาณเปิด และ 0 แทนสัญญาณปิด หลังจากนั้นจะนำตัวเลขมาประกอบเป็นชุดตัวเลขเพื่อแทนตัวเลข ตัวอักษร และสัญลักษณ์ต่าง ๆ

ระบบเลขฐานที่ใช้ในคอมพิวเตอร์

เป็นที่ทราบกันดีว่าคอมพิวเตอร์ทำงานด้วยกระแสไฟฟ้า ดังนั้นจึงมีการแทนที่สภาวะของกระแสไฟฟ้าได้ 2 สภาวะ คือ สภาวะที่มีกระแสไฟฟ้า และสภาวะที่ไม่มีกระแสไฟฟ้า และเพื่อให้โปรแกรมเมอร์สามารถสั่งการคอมพิวเตอร์ได้ จึงได้มีการสร้างระบบตัวเลขที่นำมาแทนสภาวะของกระแสไฟฟ้า โดยตัวเลข 0 จะแทนสภาวะไม่มีกระแสไฟฟ้า และเลข 1 แทนสภาวะมีกระแสไฟฟ้า

สภาวะมีกระแสไฟฟ้า แทนด้วยตัวเลข 1

สภาวะไม่มีกระแสไฟฟ้า แทนด้วยตัวเลข 0

ระบบตัวเลขที่มีจำนวน 2 จำนวน (2 ค่า) เรียกว่าระบบเลขฐานสอง (Binary Number System) ซึ่งเป็นระบบตัวเลขที่สามารถนำมาใช้ในการสั่งงานคอมพิวเตอร์ โดยการแทนที่สภาวะต่างๆ ของกระแสไฟฟ้า แต่ในชีวิตประจำวันของคนเราจะคุ้นเคยกับตัวเลขที่มีจำนวน 10 จำนวน คือ เลข 0 - 9 ซึ่งเรียกว่าระบบเลขฐานสิบ (Decimal Number System) ดังนั้นจึงมีความจำเป็นต้องศึกษาระบบเลขฐาน ประกอบการการศึกษาวิชาด้านคอมพิวเตอร์

ระบบจำนวนที่ใช้ในทางคอมพิวเตอร์ ประกอบด้วย

- ระบบเลขฐานสอง (Binary Number System) ประกอบด้วยตัวเลข 0 และ 1
- ระบบเลขฐานแปด (Octal Number System) ประกอบด้วยตัวเลข 0 - 7
- ระบบเลขฐานสิบ (Decimal Number System) ประกอบด้วยตัวเลข 0 - 9
- ระบบเลขฐานสิบหก (Hexadecimal Number System) ประกอบด้วยตัว เลข 0 - 9 และ A - F

ระบบจำนวน	**จำนวนหลัก (Digit)**															
ฐานสอง	0	1														
ฐานแปด	0	1	2	3	4	5	6	7	8							
ฐานสิบ	0	1	2	3	4	5	6	7	8	9						
ฐานสิบหก	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

ระบบเลขฐานสอง

ระบบเลขฐานสอง มีสัญลักษณ์ที่ใช้เพียงสองตัว คือ 0 และ 1 ถ้าเปรียบเทียบเลขฐานสอง กับเลขฐานสิบแล้ว ค่าของหลักที่ถัดจากหลักที่น้อยที่สุด (LSD) ขึ้นไป จะมีค่าเท่ากับ ฐานสองยกกำลังหมายเลขหลัก แทนที่จะเป็น 10 ยกกำลัง ดังนี้

เลขฐานสิบ				เลขฐานสอง		
10^0	= 1	หน่วย		2^0	= 1	หนึ่ง
10^1	= 10	สิบ		2^1	= 2	สอง
10^2	= 100	ร้อย		2^2	= 4	สี่
10^3	= 1000	พัน		2^3	= 8	แปด

ระบบเลขฐานสองเกิดจากการใช้ตัวเลขเพียง 2 ตัว คือ 0 และ 1 ดังนั้น สมการคือ

$$N = ... + (d_3 \times 2^3) + (d_2 \times 2^2) + (d_1 \times 2^1) + (d_{0 \times} 2^0)$$

เมื่อ d คือค่า 0 หรือ 1

เช่น $1101 = (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$

เพื่อตัดปัญหายุ่งยาก ในการแทนค่าของเลขระบบต่างๆ เรานิยมเขียน ตัวเลขอยู่ในวงเล็บ และเขียนค่าของฐานนั้นอยู่นอกวงเล็บ เช่น $(101101)_2 = (45)_{10}$

สำหรับเศษส่วน จะเขียนค่าของเศษส่วนอยู่หลังจุด (Binary Point) ยกกำลังเป็นลบ เพิ่มขึ้นตามลำดับ ดังตัวอย่าง

ระบบตัวเลขกับรหัสข้อมูล

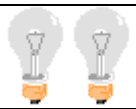
รหัสข้อมูล (Data Representation) หมายถึง รหัสที่ใช้แทนตัวเลข ตัวอักษร สัญลักษณ์ต่างๆ ที่ประกอบอยู่ในคำสั่งและข้อมูล เพื่อใช้ในการประมวลผล สามารถแบ่งได้ 2 ประเภทคือ

- รหัสภายในระบบคอมพิวเตอร์ (Internal Code) เป็นรหัสที่ใช้แทนข้อมูลในหน่วยความจำของคอมพิวเตอร์ เช่น
  - รหัส BCD - Binary Code Decimal BCD เป็นรหัสข้อมูลที่ประกอบด้วยเลขฐานสอง 6 บิต แทนข้อมูล 1 อักขระ (1 Character) จึงสามารถสร้างรหัสข้อมูลได้จำนวน $2^6 = 64$ รหัสรหัสทั้ง 6 บิต แบ่งได้เป็น 2 กลุ่ม โดย 2 บิตแรกเรียกว่า Zone Bit และ 4 บิตถัดไปเรียกว่า Numeric Bit

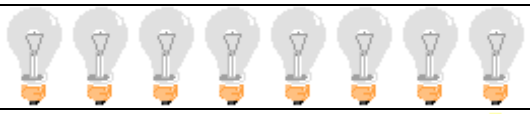
- รหัส EBCDIC - Extended Binary Coded Decimal Interchange Code
- รหัส ASCII - American Standard Code for Information Interchange
- รหัส สมอ.
- รหัส Unicode

- รหัสภายนอกระบบคอมพิวเตอร์ (External Code) เป็นรหัสที่พัฒนาสำหรับบันทึกข้อมูลนอกเครื่องคอมพิวเตอร์ เช่นรหัสที่ใช้กับบัตรเจาะรู

รหัสภายในคอมพิวเตอร์ แทนได้กับสภาวะของกระแสไฟฟ้า ตามจำนวนสายสัญญาณ เช่น ถ้ามีสายสัญญาณ 2 เส้น ก็สามารถสร้างรหัสแทนข้อมูลได้ 4 ค่า (คิดจาก 2^2) คือ

สภาวะไฟฟ้า 2 เส้น	รหัสข้อมูล
	00
	01
	10
	11

ดังนั้นถ้ามีสายสัญญาณ 8 เส้น ก็สามารถสร้างรหัสแทนข้อมูลได้ จำนวน $2^8 = 256$ ค่า เป็นต้น

สภาวะไฟฟ้า 8 เส้น	รหัสข้อมูล
	00000000
	00000001
	11111111

บิต (Bit)

สภาวะไฟฟ้า 1 เส้น หรือค่า 0 หรือ 1 แต่ละค่าเรียกว่า บิต (Bit) ซึ่งเป็นคำย่อของ "BInary digiT"

ไบต์ (Byte)

กลุ่มของบิตที่มีความหมายเฉพาะเรียกว่า ไบต์ (Byte) ดังนั้นถ้ามีสายสัญญาณ 8 เส้น แสดงว่ามีสัญญาณที่สามารถผสมผสานกันได้ 8 บิต เมื่อนำค่าสัญญาณต่างๆ มาผสมผสานกัน ก็สามารถสร้างรหัสแทนข้อมูลได้ จำนวน 2^8 = 256 ค่า เป็นต้น ดังตัวอย่างในตารางที่แสดงอักขระ, การเรียงกันของบิต และค่าเลขฐาน 10 ที่แทนอักขระ

Character	Bit pattern	Byte number		Character	Bit pattern	Byte number
A	01000001	65		ผ	10111100	188
B	01000010	66		.	00101110	46
C	01000011	67		:	00111010	58
a	01100001	97		$	00100100	36
b	01100010	98		\	01011100	92
Character	**Bit pattern**	**Byte number**		**Character**	**Bit pattern**	**Byte number**
o	01101111	111		~	01111110	126
p	01110000	112		1	00110001	49
q	01110001	113		2	00110010	50
r	01110010	114		9	00111001	57
x	01111000	120		ฉ	10101001	169
y	01111001	121		>	00111110	62
z	01111010	122			10001001	137

ดังนั้นถ้าต้องการป้อนคำว่า Hello จะมีค่าเท่ากับข้อมูลจำนวน 6 ไบต์ ซึ่งมักจะได้ยินว่า 1 ไบต์ เทียบกับ 1 ตัวอักษรนั่นเอง

American Standard Code for Information Interchange (ASCII)

ASCII เป็นรหัสที่นิยมใช้กันอย่างแพร่หลายในปัจจุบัน พัฒนาโดยสถาบันมาตรฐานแห่งชาติสหรัฐอเมริกา (American National Standard Institute: ANSI) ประกอบด้วยเลขฐานสอง 7 บิต (ปัจจุบันใช้ 8 บิต) เรียกว่า 1 ไบต์ (Byte) แทนอักขระ 1 ตัว ซึ่งเป็นรหัสที่นิยมใช้กันบนคอมพิวเตอร์ระบบ PC ชุดคือ

- 32 ชุดแรก (ตำแหน่งที่ 0 - 31) แทนรหัสควบคุมต่างๆ
- ตำแหน่งที่ 32 - 127 แทนอักขระภาษาอังกฤษ, ตัวเลขและสัญลักษณ์ต่างๆ เรียกว่า Lower ASCII
- 128 ชุดหลัง (ตำแหน่งที่ 128 - 255) แทนอักขระในภาษาต่างๆ เช่น อักขระภาษาไทย เป็นต้น ทำให้

คอมพิวเตอร์สามารถรับ/ส่งข้อมูลภาษาอื่นๆ ได้ เรียกว่า Higher ASCII

ภาษาคอมพิวเตอร์

ภาษาของคอมพิวเตอร์นั้น มีหลายภาษา ซึ่งแต่ละภาษาก็จะมีเอกลักษณ์เป็นของตัวเอง แตกต่างกันไปตามการใช้งาน โดยแยกเป็น 3 แบบ คือ ภาษาที่ใช้ทั่วๆ ไป ภาษาหลักอื่น ๆ และ ภาษาพิเศษ

ภาษาที่ใช้ทั่วๆ ไป

• **ภาษาเบสิก** เป็นภาษาที่ใช้กันมากที่สุดสำหรับไมโครคอมพิวเตอร์ มินิคอมพิวเตอร์ และเมนเฟรมคอมพิวเตอร์ ผู้คิดค้นคือ DARTMOUTH COLLEGE เนื่องจากว่าเป็นภาษาที่สามารถใช้โต้ตอบไปมาระหว่างกันได้ทันที จึงง่ายสำหรับผู้ที่ใช้ที่จะเริ่มต้นจะเรียน หรือต้องการจะเรียนรู้หรือทำการแก้ไข และแก้ไขซินแทกซ์เออเร่อส์ได้ง่าย

• **ภาษาโคบอล** ย่อมาจากคำว่า Common Business Oriented Language เหมาะที่จะใช้กับการประมวลผลข้อมูลทางธุรกิจ และการสร้างแฟ้มข้อมูลสำหรับงานต่างๆ ใช้ได้กับระบบคอมพิวเตอร์ทั่วๆ ไป

ภาษาหลักอื่นๆ

• **ภาษาปาสคาล** ผู้คิดค้นคือ KINKLAUS WIRTH ระหว่างปี ค.ศ. 1969-1971 ออกแบบขึ้นมาโดยใช้structure โปรแกรมมิ่งเทคนิคเข้าช่วย เพื่อให้โปรแกรมมีความเป็นมาตรฐานและง่ายแต่การแก้ไขโดยใช้คำสั่ง IF-THEN-ELSE และ DO-WHILE ปาสคาลเป็นภาษาที่ใช้กับเครื่องไมโครคอมพิวเตอร์

• **ภาษาพีแอลวัน** ย่อมาจาก Programming Language/1 เป็นภาษาที่รวมเอาข้อดีของภาษาฟอร์แทนเข้ามารวมกับข้อดีของภาษาโคบอล

ภาษาพิเศษ

• **ภาษาซี ยูนิกซ์** จะเป็นออเปอเรติ้งซิสเต็มที่เขียนอยู่ในรูปของภาษาซี ผู้คิดค้นได้แก่ BELL LABS ภาษาซีจะใช้ในการจัดทำโอเอสและโปรแกรมระบบงานสำหรับงานด้านวิจัยและธุรกิจ

• **ภาษาฟอร์ท** พบใน ค.ศ. 1970 ใช้กับงานที่มีลักษระที่เป็นงานด้านวิศวกรรม ในเริ่มแรกได้รับการเรียกว่า Fourth ทั้งนี้เพราะว่าเป็น Fourth-Generation Language และใช้ IBM1130 ในการประมวลผล ซึ่งยินยอมให้ใช้เพียง 5 ตัวอักษร สำหรับการตั้งชื่อใดๆ ก็ตามจึงต้องเปลี่ยนมาเป็น forth โดยปริยาย forth เหมาะสำหรับระบบเล็กๆ สามารถจัดทำโดยโอเอสได้และมักจะใช้กับคอมพิวเตอร์ขนาดเล็ก

• **ภาษาโลโก้** พบเมื่อ ค.ศ. 1970 ใช้สอนเด็กในการฝึกหัดโต้ตอบกับเครื่องคอมพิวเตอร์ โลโก้ใช้ในการจัดทำกราฟฟิก วาดภาพได้ ปกติใช้กันมากในโรงเรียน ใช้กับไมโครคอมพิวเตอร์

• **ภาษาซิมูเลชั่น** ภาษาซิมมูเลชั่นที่ใช้ในการจัดทำโมเดล จะได้แก่ GPSS ที่มีชื่อเต็มว่า General Purpose System Simulator และ SIMULA ซึ่งมีชื่อย่อมาจาก Simulation Language