



2.3.1 การแปลงชนิดข้อมูลในภาษา c#

การแปลงชนิดข้อมูลในภาษา c# หากข้อมูลไม่ใช่ชนิดเดียวกันจะไม่สามารถนำมากระทำใดๆ กันได้ จึงต้องมีการแปลงชนิดข้อมูลให้เป็นชนิดเดียวกันก่อน ซึ่งมีรูปแบบการแปลงชนิดข้อมูล ดังนี้

1. implicit type conversion
2. explicit type conversion
3. การใช้ เมธอด Parse
4. การใช้ เมธอด ToString

1) **Implicit type conversion** คือ เป็นการแปลงชนิดข้อมูลที่เกิดขึ้นโดยอัตโนมัติ เกิดจากการที่เรานำข้อมูลชนิดเดียวกันหรือต่างกันมากระทำกันทางคณิตศาสตร์ แล้วผลลัพธ์ที่ได้เกินขอบเขตของชนิดข้อมูล โดยจะแปลงไปสู่ขนาดของข้อมูลที่ใหญ่กว่าเสมอ ตามลำดับนัยสำคัญ ดังนี้

ตารางที่ 2.3.1 ตารางแสดงลำดับนัยสำคัญของชนิดข้อมูล

ลำดับนัยสำคัญ	ชนิดข้อมูล
1	long double
2	double
3	float
4	unsigned long int
5	long int
6	unsigned int
7	int
8	short
9	char

ตัวอย่างที่ 1

```
int number1 = 20 ;  
double number2 = 3.5 ;  
answer = number1 + number2
```

ผลลัพธ์ (answer) ที่ได้จะถูกแปลงเป็นชนิด double โดยอัตโนมัติ เนื่องจาก number2 เป็นชนิดข้อมูลแบบ double ซึ่งใหญ่กว่า number1 ที่เป็นชนิดข้อมูลแบบ int

ตัวอย่างที่ 2

```
int number1 = 20 ;  
double number2 = 3.5 ;  
double answer = number1 + number2
```

ชนิดข้อมูลที่น่ามารองรับไม่จำเป็นต้องเป็น double ก็ได้ เพียงแต่ขอให้รองรับกับผลลัพธ์ของ number1 + number2 ได้ก็พอ

ตัวอย่างที่ 3 เป็นตัวอย่างที่กำหนดชนิดข้อมูลผิด

```
int number1 = 20 ;  
double number2 = 3.5 ;  
int answer = number1 + number2
```

จะทำให้โปรแกรม error เพราะเกินขอบเขตของตัวแปร int

2) **Explicit type conversion** คือ ผู้เขียนโปรแกรมทำการแปลงชนิดข้อมูลหนึ่งไปเป็นอีกชนิดหนึ่งด้วยตัวเอง การแปลงชนิดข้อมูลนี้เรียกอีกอย่างว่า cast ซึ่งทำได้โดยระบุชนิดข้อมูลปลายทางที่ต้องการไว้ในวงเล็บ แล้ววางไว้หน้านิพจน์ที่ต้องการแปลงชนิดข้อมูล

ตัวอย่าง

```
int number1 = 20 ;  
double number2 = 3.5 ;  
int Result = (int)(number1 + number2);
```

จะเป็นการแปลง number1 + number2 เป็นข้อมูลชนิด int เพื่อที่จะทำให้ไม่เกินขอบเขตของตัวแปร Result ที่มารับ (ผลลัพธ์ที่เป็นทศนิยมจะถูกปัดทำให้เกิดความคลาดเคลื่อนได้)

3) การใช้ เมธอด Parse คือ เป็นการแปลงข้อมูลชนิด string หรือข้อความให้เป็นข้อมูลชนิดตัวเลข

ตัวอย่างที่ 1

```
i = int.Parse(textBox1.Text);
```

เป็นการแปลงข้อมูลที่รับจาก textBox1 ซึ่งเป็นชนิดข้อมูลแบบข้อความ(string) ให้เป็นชนิดข้อมูลแบบตัวเลขจำนวนเต็ม(int) แล้วเก็บค่าในตัวแปร i

ตัวอย่างที่ 2

```
i = double.Parse("22.45");
```

เป็นการแปลงข้อมูล "22.45" ซึ่งเป็นชนิดข้อมูลแบบข้อความ(string) ให้เป็นชนิดข้อมูลแบบตัวเลขทศนิยม(double) แล้วเก็บค่าในตัวแปร i

4) การใช้ เมธอด ToString คือ เป็นการแปลงข้อมูลประเภทใดๆ ให้เป็นข้อมูลชนิดแบบข้อความ

ตัวอย่างที่ 1

```
int a = 20;  
textBox1.Text = a.ToString();
```

เป็นการแปลงตัวเลข 20 ซึ่งเป็นข้อมูลชนิดจำนวนเต็ม (int) ให้เป็นชนิดข้อมูลแบบข้อความ แล้วแสดงผลใน textBox1 (textbox จะแสดงผลได้เฉพาะข้อมูลแบบข้อความ string)

ตัวอย่างที่ 2

```
tme = DateTime.Now.ToString();
```

เป็นการแปลงข้อมูลวันเวลาปัจจุบันให้เป็นชนิดข้อมูลแบบข้อความแล้วเก็บค่าในตัวแปร tme

2.3.2 ฟังก์ชันสำเร็จรูป

เป็นฟังก์ชันที่สามารถเขียนเรียกใช้งานได้ทันทีโดยไม่ต้องมาเขียนฟังก์ชันเองทำให้ผู้ใช้งานสะดวกในการนำมาใช้งาน ดังนี้

1. ฟังก์ชันทางคณิตศาสตร์ คือฟังก์ชันที่ใช้ในการคำนวณค่าทางคณิตศาสตร์อย่างใดอย่างหนึ่ง โดยมีเทห์อด (Method) ที่ใช้งานดังนี้

ตารางที่ 2.3.2 ตารางแสดงฟังก์ชันทางคณิตศาสตร์

เมทอด	ความหมาย	ตัวอย่าง
Math.Pow(ฐาน , เลขชี้กำลัง)	หาค่ายกกำลัง	Math.Pow(x , 2) คือ หาค่า x^2
Math.Sqrt(ตัวเลข)	หาค่ารากที่สอง	Math.Sqrt(x) คือ หารากที่สองของ x
Math.Abs(ตัวเลขทศนิยม)	หาค่า absolute	Math.Abs(x) คือ หาค่า $ x $
Math.PI	หาค่า π	มีค่าเท่ากับ 22/7
Math.Sin(มุมที่มีค่าเป็นเรเดียน)	หาค่า sine	Math.Sin(x) คือ หาค่า $\sin(x)$
Math.Cos(มุมที่มีค่าเป็นเรเดียน)	หาค่า cosine	Math.Cos(x) คือ หาค่า $\cos(x)$
Math.Tan(มุมที่มีค่าเป็นเรเดียน)	หาค่า tangent	Math.Tan(x) คือ หาค่า $\tan(x)$
Math.BigMul(จำนวนเต็มตัวที่1 , จำนวนเต็มตัวที่2)	หาค่าผลคูณของตัวเลข 2 ตัว	Math.BigMul(a,b) คือ หาค่าผลคูณของ a และ b
Math.Celing(ตัวเลขที่มีทศนิยม)	หาค่าจำนวนเต็มที่ถูกปัดขึ้นมาโดยไม่สนใจทศนิยม	Math.Celing(10.14) จะได้ผลลัพธ์คือ 11
Math.Floor(ตัวเลขที่มีทศนิยม)	หาค่าจำนวนเต็มที่ถูกปัดลงโดยไม่สนใจทศนิยม	Math.Floor(10.98) จะได้ผลลัพธ์คือ 10

ตารางที่ 2.3.2 ตารางแสดงฟังก์ชันทางคณิตศาสตร์ (ต่อ)

เมทอด	ความหมาย	ตัวอย่าง
Math.Round(ตัวเลขที่มีทศนิยม)	หาค่าจำนวนเต็ม โดยจะปัดขึ้นเมื่อมีทศนิยมมากกว่า 0.5 แต่ถ้าเลขที่เป็นจุดทศนิยมมีค่าเท่ากับ 0.5 ให้ดูที่เลขจำนวนเต็มหากเป็นเลขคู่จะปัดทศนิยมทิ้ง หากเป็นเลขคี่จะปัดทศนิยม 0.5 ขึ้น	Math.Round(10.4)ผลลัพธ์คือ 10 Math.Round(10.5)ผลลัพธ์คือ 10 Math.Round(10.8)ผลลัพธ์คือ 10 Math.Round(11.4)ผลลัพธ์คือ 11 Math.Round(11.5)ผลลัพธ์คือ 12 Math.Round(11.7)ผลลัพธ์คือ 12
Math.Round(ตัวเลขที่มีทศนิยม , จำนวนตำแหน่งทศนิยมที่ต้องการ)	หาค่าทศนิยมที่มีตำแหน่งทศนิยมตามต้องการ	Math.Round(23.1243565 , 2)ผลลัพธ์คือ 23.12 Math.Round(62.57879821 , 4)ผลลัพธ์คือ 62.5788 (เลข 7 ตำแหน่งที่ 4 เป็นเลขคี่ เมื่อเลขที่ต่อท้ายมากกว่า 5 จึงถูกปัดขึ้น) Math.Round(62.58879821 , 2)ผลลัพธ์คือ 62.58 (เลข 8 ตำแหน่งที่ 2 เป็นเลขคู่ ถึงแม้เลขที่ต่อท้ายจะมากกว่า 5 ก็ตามจะถูกปัดทิ้งไป)

2. ฟังก์ชันเกี่ยวกับ String คือ ฟังก์ชันที่ใช้ในการจัดการเกี่ยวกับข้อความ โดยมีเมทอดที่ใช้งาน ดังนี้

ตารางที่ 2.3.3 ตารางแสดงฟังก์ชันเกี่ยวกับ String

เมทอด	ความหมาย	ตัวอย่าง
Length	นับจำนวนอักขระภายใน string	<code>len1=str1. Length;</code> len1 มีค่าเท่ากับจำนวนอักขระทั้งหมดของ str1
ToLower	เปลี่ยนตัวอักษรให้เป็นตัวอักษรพิมพ์เล็กทั้งหมด	<code>str2=str1. ToLower();</code> str2 จะมีค่าเป็นตัวอักษรพิมพ์เล็กทั้งหมด
ToUpper	เปลี่ยนตัวอักษรให้เป็นตัวอักษรพิมพ์ใหญ่ทั้งหมด	<code>str2=str1. ToLower();</code> str2 จะมีค่าเป็นตัวอักษรพิมพ์เล็กทั้งหมด

ตารางที่ 2.3.3 ตารางแสดงฟังก์ชันเกี่ยวกับ String (ต่อ)

เมทอด	ความหมาย	ตัวอย่าง
Compare	ตรวจสอบ string 2 ค่า ว่าเหมือนกันหรือไม่	result=String.Compare(str1,str2); result จะมีผลลัพธ์ได้ 3 ค่า ดังนี้ <ol style="list-style-type: none"> 1. ผลลัพธ์เป็น 1 เมื่อ str1 มีค่ารหัสแอสกีมากกว่า str2 2. ผลลัพธ์เป็น 0 เมื่อ str1 มีค่ารหัสแอสกีเท่ากับ str2 3. ผลลัพธ์เป็น -1 เมื่อ str1 มีค่ารหัสแอสกีน้อยกว่า str2
Replace	แทนที่ string ด้วย string อีกตัว ใน string ที่ระบุ	Result=str1. Replace(str2,str3); Result คือผลลัพธ์ที่เปลี่ยนข้อความใน str1 ที่มีข้อความตรงกับ str2 ให้เปลี่ยนเป็น str3 str2=str1. Replace("in", "me"); str2 คือ ผลลัพธ์ที่เปลี่ยนข้อความใน str1 ที่มีคำว่า in ให้เปลี่ยนเป็นคำว่า me
Remove	ตัด string ในตำแหน่งที่ต้องการทิ้ง	str2=str1.Remove(3); str2 จะมีค่าเป็น str1 ที่ตัดข้อความในตำแหน่งที่ 4 ทิ้งไป str2=str1.Remove(4,2); str2 จะมีค่าเป็น str1 ที่ตัดความตำแหน่งที่ 5 และนับไป 2 ตัว นั่นคือตัดตำแหน่งที่ 5และ6 ออกไป
Insert	แทรก string ในตำแหน่งที่ต้องการ	str3=str1.Insert(3,str2); str3 จะมีค่าเป็น str1 ที่ถูกแทรก str2 ลงไปในตำแหน่งที่ 4 str2=str1.Insert(3, "love"); str2 จะมีค่าเป็น str1 ที่ถูกแทรกคำว่า love ลงไปในตำแหน่งที่ 4
Trim	ตัดข้อความที่เป็น space ส่วนหัว และ ส่วนท้ายออก	result = str.Trim(); result จะมีค่าเป็น str ที่มีการตัด space ส่วนหัว และ ส่วนท้ายออกแล้ว

ตารางที่ 2.3.3 ตารางแสดงฟังก์ชันเกี่ยวกับ String (ต่อ)

เมทอด	ความหมาย	ตัวอย่าง
TrimStart	ตัดข้อความที่เป็น space ส่วนหัวออก	result = str. TrimStart (); result จะมีค่าเป็น str ที่มีการตัด space ส่วนหัวออกแล้ว
TrimEnd	ตัดข้อความที่เป็น space ส่วนท้ายออก	result = str. TrimEnd (); result จะมีค่าเป็น str ที่มีการตัด space ส่วนท้ายออกแล้ว

3. ฟังก์ชันเกี่ยวกับเวลา คือ ฟังก์ชันที่ใช้ในการจัดการเกี่ยวกับเวลา โดยมีเมทอดที่ใช้งาน ดังนี้

ตารางที่ 2.3.4 ตารางแสดงฟังก์ชันเกี่ยวกับเวลา

เมทอด	ความหมาย	ตัวอย่าง
DateTime.Now	ค่าวันเดือนปี และเวลา ปัจจุบันตามนาฬิกาของเครื่อง	DateTime dt; dt=DateTime.Now; คือ กำหนดให้ dt เก็บค่าวันเดือนปีและเวลาปัจจุบันของเครื่อง เช่น 16/6/2555 08:30:10
DateTime.Now.ToShortDateString()	ค่าวันเดือนปีปัจจุบันตามนาฬิกาของเครื่อง แบบ ShortDate	ถ้าวันเดือนปีและเวลาปัจจุบัน คือ 16/6/2555 08:30:10 จะได้ผลลัพธ์เป็น 16/6/2555
DateTime.Now.ToLongDateString()	ค่าวันเดือนปีปัจจุบันตามนาฬิกาของเครื่อง แบบ LongDate	ถ้าวันเดือนปีและเวลาปัจจุบัน คือ 16/6/2555 08:30:10 จะได้ผลลัพธ์เป็น 16 มิถุนายน 2555
DateTime.Now.ToShortTimeString()	ค่าเวลาปัจจุบันตามนาฬิกาของเครื่อง แบบ ShortTime	ถ้าวันเดือนปีและเวลาปัจจุบัน คือ 16/6/2555 08:30:10 จะได้ผลลัพธ์เป็น 08:30
DateTime.Now.ToLongTimeString()	ค่าเวลาปัจจุบันตามนาฬิกาของเครื่อง แบบ LongTime	ถ้าวันเดือนปีและเวลาปัจจุบัน คือ 16/6/2555 08:30:10 จะได้ผลลัพธ์เป็น 08:30:10