

# C# 泛型

Ian Chen  
2019/11/11

# 繼承 vs 介面

- ▶ 什麼情境下適合用哪個？



# 明確轉換 vs “as”

- ▶ 什麼情境下適合用哪個？



# Generics

泛型



# 什麼是泛型？

GenericSample\_1

The background of the slide features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern and dynamic visual effect.

如果沒有泛型？

GenericSample\_2

The right side of the slide features a decorative graphic consisting of several overlapping, semi-transparent green triangles and polygons of various shades, creating a modern, abstract background element.

# ArrayList

## ArrayList.Add(Object) 方法

命名空間: [System.Collections](#)

組件: System.Collections.NonGeneric.dll, mscorlib.dll, netstandard.dll, System.Runtime.Extensions.dll

將物件加入至 [ArrayList](#) 的末端。

C#

複製

```
public virtual int Add (object value);
```

GenericSample\_3

可能造成什麼問題？

GenericSample\_4

The background of the slide is composed of several overlapping, semi-transparent green triangles and polygons of various shades, ranging from a light lime green to a dark forest green. These shapes are primarily located on the right side of the slide, with some extending towards the center. A thin, light gray line originates from the bottom left of the text 'GenericSample\_4' and extends diagonally upwards and to the right, passing through the green geometric shapes.



# ArrayList

- ▶ Boxing
- ▶ Unboxing



# Boxing 、 Unboxing

## Reference Type

Type Object Pointer

Sync Block Index

Instance Fields

## Value Type

Instance Fields



# Boxing 、 Unboxing

On the stack

On the heap

**i**



`int i=123;`

**o**



`object o=i;`

**(i boxed)**



**j**



`int j=(int) o;`

# Boxing 、 Unboxing

- ▶ 造成的效能問題？
- ▶ 使用上的問題？

GenericSample\_5

# 泛型

- ▶ Member
  - ▶ Class
  - ▶ Interface
  - ▶ Method
  - ▶ Delegate
  - ▶ Field
  - ▶ Property

GenericSample\_6

# 泛型

- ▶ `<T>`
- ▶ Type Parameter
- ▶ Type Argument

GenericSample\_7



# 泛型

- ▶ 自動推斷<T>的類型

GenericSample\_8

# 不同type的<T>是相同型別嗎？

- ▶ Ex:
- ▶ `MyClass<int>`
- ▶ `MyClass<string>`





# 泛型

## ► Static Constructors

GenericSample\_9



如果方法裡面會用到特定Type的Member...

GenericSample\_13

# 泛型條件約束

► where



# 泛型條件約束

► new()

GenericSample\_10



# 泛型條件約束

- ▶ where T : class
- ▶ where T : <Base Class Name>
- ▶ where T : <Interface Name>
- ▶ where T : U

GenericSample\_11

# 泛型條件約束

- ▶ where T : struct
- ▶ where T : enum (C# 7.3+)
- ▶ where T : delegate
- ▶ where T : notnull

GenericSample\_11

# Func

## Func<TResult> 代理人

命名空間: [System](#)

組件: System.Runtime.dll, mscorlib.dll, netstandard.dll, System.Core.dll

封裝沒有參數並傳回 `TResult` 參數所指定之型別值的方法。

C#

```
public delegate TResult Func<out TResult>();
```

### 類型參數

**TResult**

這個委派所封裝之方法的傳回值之型別。

### 傳回值

**TResult**

這個委派所封裝之方法的傳回值。

繼承 [Object](#) → [Delegate](#) → `Func<TResult>`

FuncSample\_1

# Func

► Method

FuncSample\_2

An abstract graphic design featuring overlapping translucent green triangles and polygons of various shades, creating a layered, geometric effect on the right side of the slide.



# Func

## ► 方法參數

FuncSample\_3

A decorative graphic on the right side of the slide, consisting of several overlapping, semi-transparent green triangles and polygons of various shades, creating a modern, abstract background element.

怎麼實做一個Where ?

FuncSample\_3

The background of the slide features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern, layered effect. A thin, light gray line extends from the bottom left towards the right, passing behind the 'FuncSample\_3' text.

# Action

## Action<T> 代理人

命名空間: [System](#)

組件: System.Runtime.dll, mscorlib.dll, netstandard.dll

封裝具有單一參數的方法，並且不會傳回值。

C#

```
public delegate void Action<in T>(T obj);
```

### 類型參數

**T**

這個委派所封裝之方法的參數型別。

### 參數

**obj**

這個委派所封裝之方法的參數。

繼承 [Object](#) → [Delegate](#) → Action<T>

ActionSample\_1

End

