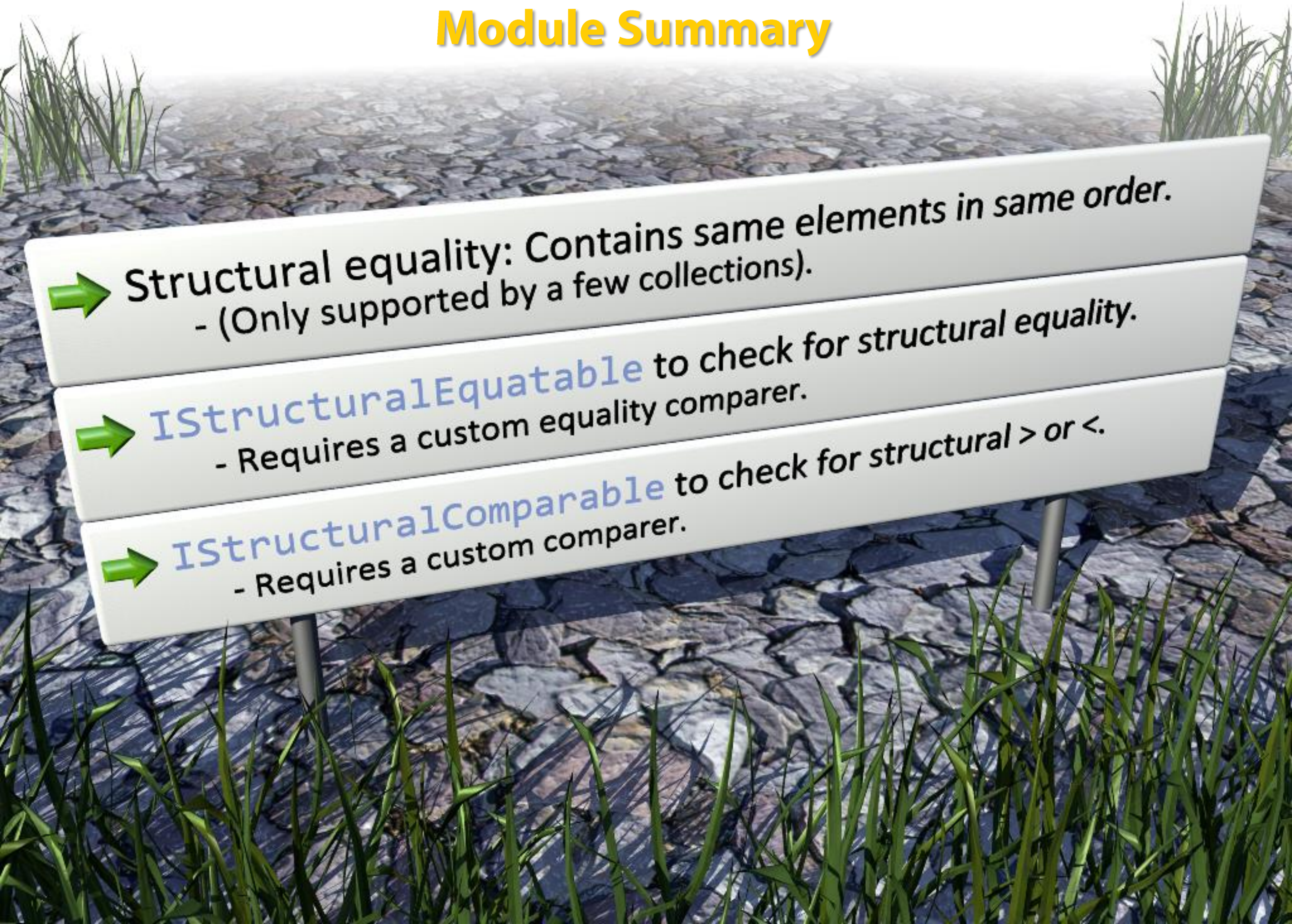
Structural Equality and Comparisons

Simon Robinson
http://TechieSimon.com
@TechieSimon

- Way of checking if collections contain:
  - The same elements.
  - In the same order.
- `IStructuralEquatable`.
- `IStructuralComparable`.
- Limited support:
  - Arrays.
  - Tuples.
- Course overview.

Module Summary



- Structural equality: Contains same elements in same order.
  - (Only supported by a few collections).
- `IStructuralEquatable` to check for structural equality.
  - Requires a custom equality comparer.
- `IStructuralComparable` to check for structural > or <.
  - Requires a custom comparer.

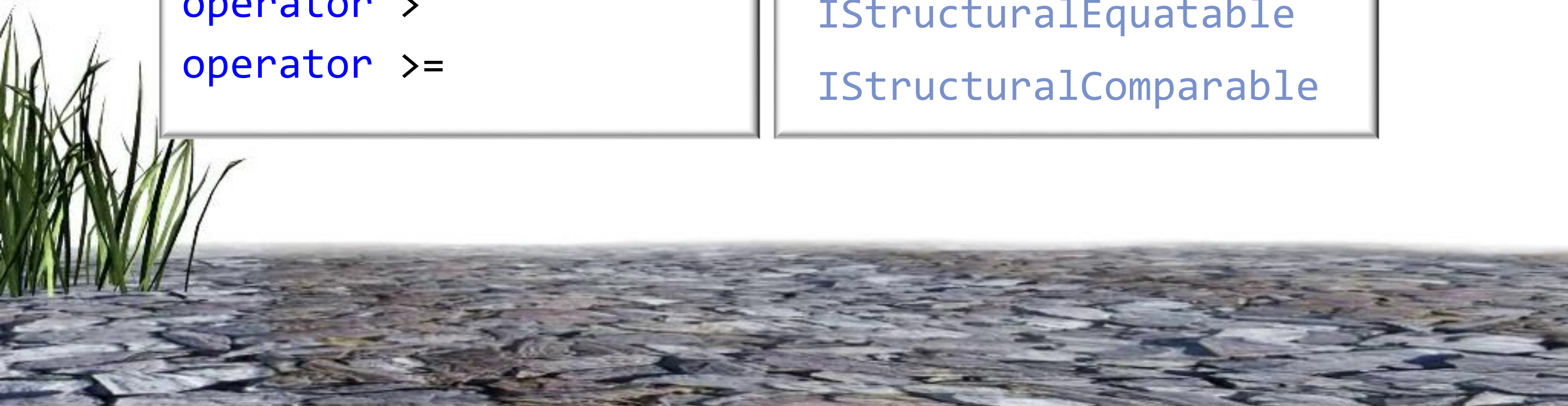
The Methods and Interfaces…

Methods/Operators

```
object.Equals()
static object.Equals()
object.ReferenceEquals()
object.GetHashCode()
operator ==
operator !=
operator <
operator <=
operator >
operator >=
```

Interfaces

```
IEquatable<T>
IComparable<T>
IComparable
IComparer
IComparer<T>
IEqualityComparer
IEqualityComparer<T>
IStructuralEquatable
IStructuralComparable
```



The Interfaces

	Equality	Comparisons

"Natural"	`IEquatable<T>`	`IComparable<T>`
		`IComparable`

"Plugged-in"	`IEqualityComparer`
	`IEqualityComparer<T>`
		`IComparer`
		`IComparer<T>`

Structural	`IStructuralEquatable`	`IStructuralComparable`

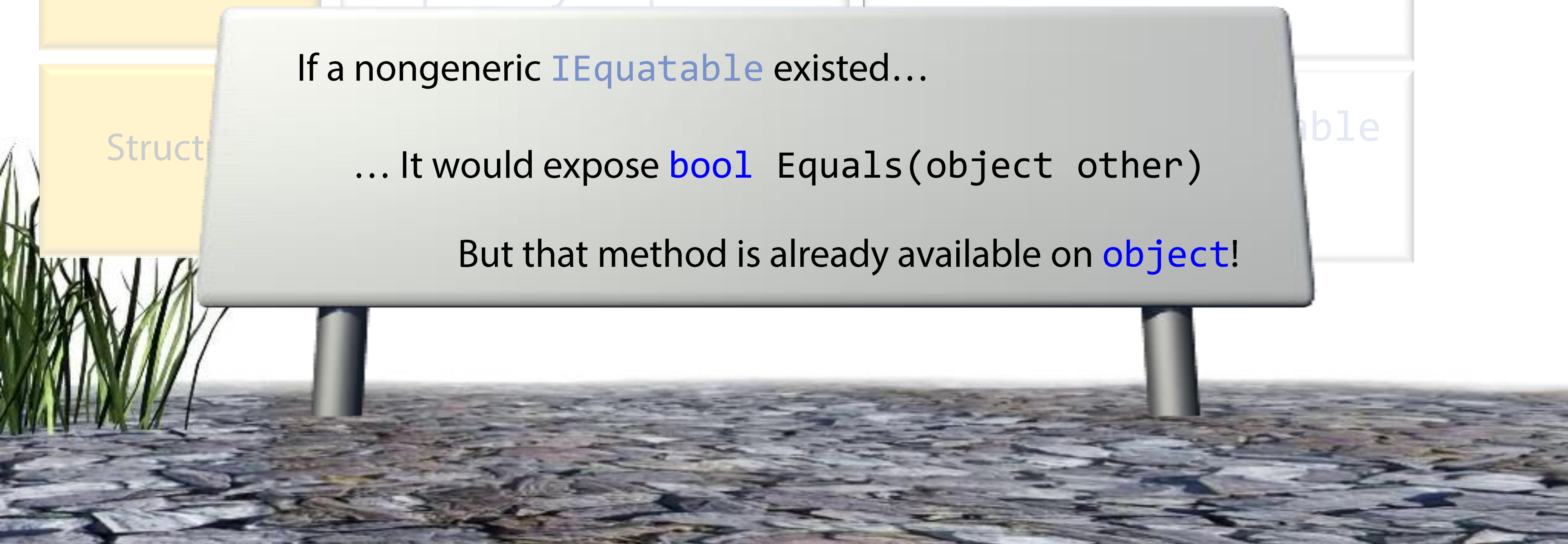
Why is There No IEquatable?

	Equality	Comparisons

"Natural"	`IEquatable<T>`	`IComparable<T>`
		`IComparable`

"Plugged-in"	`IEqualityComparer`
	`IEqualityComparer<T>`
		`IComparer`
		`IComparer<T>`

Struct		able



If a nongeneric `IEquatable` existed…

… It would expose `bool Equals(object other)`

But that method is already available on `object`!

The Methods and Operators

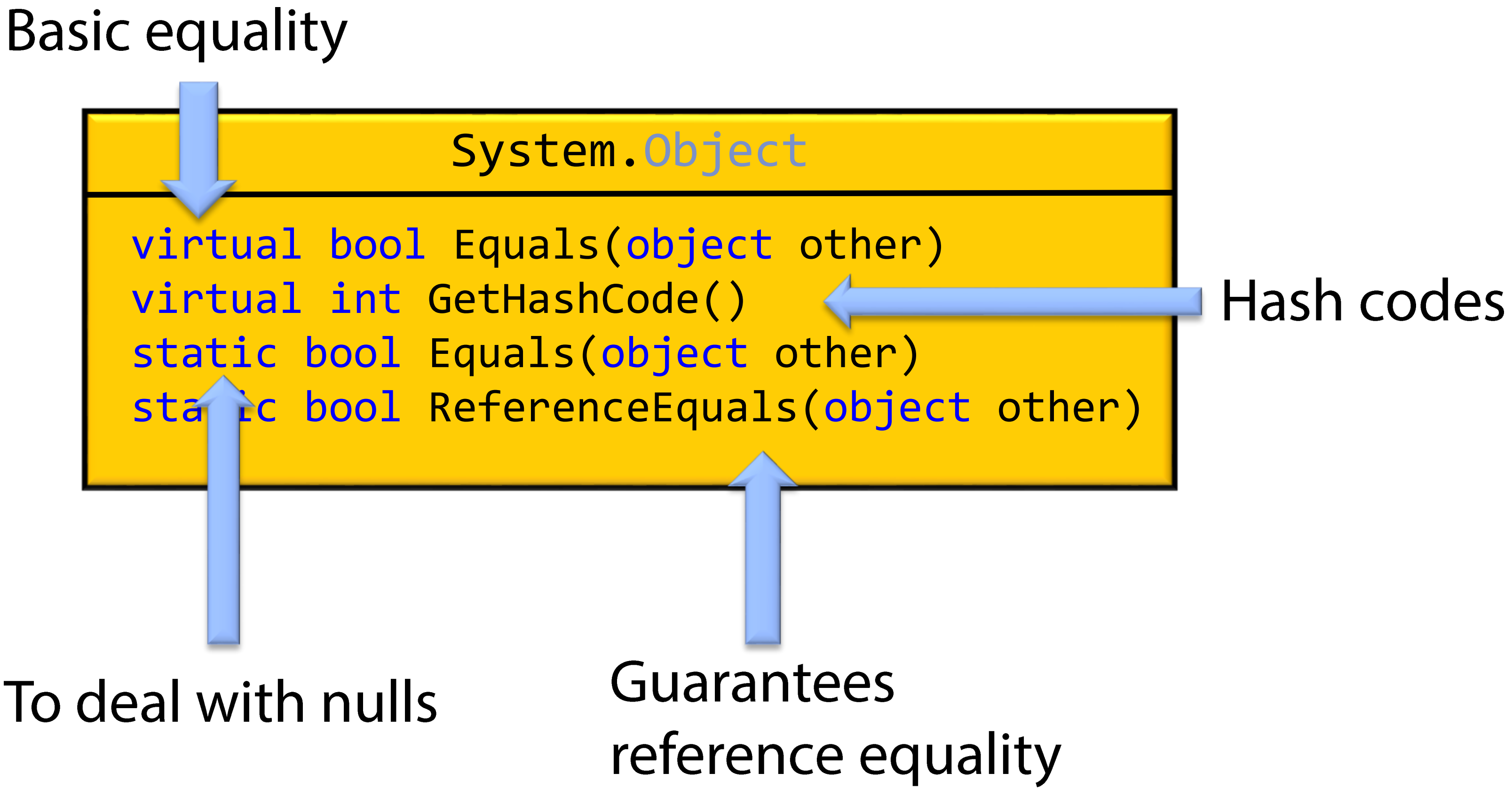
Methods/Operators

```
object.Equals()
static object.Equals()
object.ReferenceEquals()
object.GetHashCode()
operator ==
operator !=
operator <
operator <=
operator >
operator >=
```

Interfaces

```
IEquatable<T>
IComparable<T>
IComparable
IComparer
IComparer<T>
IEqualityComparer
IEqualityComparer<T>
IStructuralEquatable
IStructuralComparable
```

The Methods



Basic equality

```
System.Object
```

```
virtual bool Equals(object other)
virtual int GetHashCode()
static bool Equals(object other)
static bool ReferenceEquals(object other)
```

Hash codes

To deal with nulls

Guarantees reference equality

The Operators

These all compile to static methods

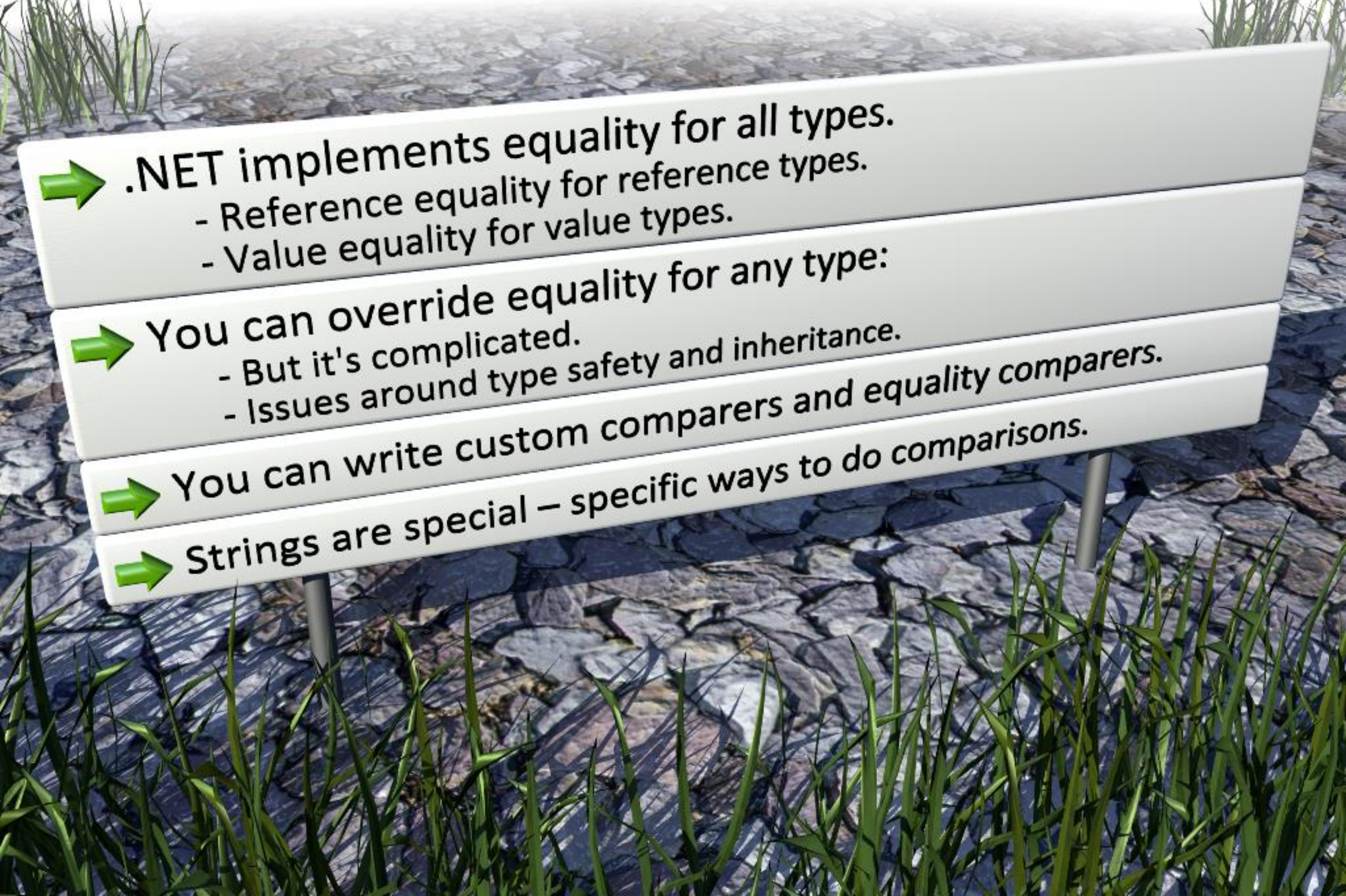
Available on `object`

```
operator ==
operator !=
operator <
operator <=
operator >
operator >=
```

All can be used with primitive numeric types

You can implement these as required on your types

Course Summary



- .NET implements equality for all types.
  - Reference equality for reference types.
  - Value equality for value types.
- You can override equality for any type:
  - But it's complicated.
  - Issues around type safety and inheritance.
- You can write custom comparers and equality comparers.
- Strings are special – specific ways to do comparisons.

C# Equality and Comparisons

Simon Robinson
http://TechieSimon.com
@TechieSimon

pluralsight
hardcore dev and IT training