

# Namespace NullaryThunk.Funky.Core. Options

## Classes

### [Nothing<T>](#)

An [Option<T>](#) that has no meaningful value.

### [O](#)

### [OptionFunctions](#)

### [Something<T>](#)

An [Option<T>](#) that has a meaningful value.

## Interfaces

### [ISomething<T>](#)

An [Option<T>](#) that has a meaningful value.

### [Nothing](#)

An object with no meaningful value. Represents the absence of a value.

### [Option<T>](#)

Represents the possibility of a T. An [Option<T>](#) can be either [Something<T>](#) or [Nothing<T>](#). If it is a [Something<T>](#) then it contains a value of . Use pattern matching to match on an [Option<T>](#)

### [Something](#)

An [Option<T>](#) that has no meaningful value.

# Interface ISomething<T>

Namespace: [NullaryThunk.Funky.Core.Options](#)

Assembly: NullaryThunk.Funky.Core.dll

An [Option<T>](#) that has a meaningful value.

```
public interface ISomething<out T> : Option<T>, Something
```

## Type Parameters

T

The type of the meaningful value contained within.

## Extension Methods

[O.ToSomething<T>\(T\)](#) , [OptionFunctions.SomethingOr<T>\(Option<T>, T\)](#) ,  
[OptionFunctions.SomethingOr<T, R>\(Option<T>, Func<T, R>, R\)](#)

## Properties

### Value

```
T Value { get; }
```

### Property Value

T

# Interface Nothing

Namespace: [NullaryThunk.Funky.Core.Options](#)

Assembly: NullaryThunk.Funky.Core.dll

An object with no meaningful value. Represents the absence of a value.

```
public interface Nothing
```

## Extension Methods

[O.ToSomething<T>\(T\)](#)

# Class Nothing<T>

Namespace: [NullaryThunk.Funky.Core.Options](#)

Assembly: NullaryThunk.Funky.Core.dll

An [Option<T>](#) that has no meaningful value.

```
public record Nothing<T> : Option<T>, Nothing, IEquatable<Nothing<T>>
```

## Type Parameters

T

The type of the meaningful value.

## Inheritance

[object](#) ← Nothing<T>

## Implements

[Option<T>](#), [Nothing](#), [IEquatable](#)<[Nothing<T>](#)>

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Extension Methods

[O.ToSomething<T>\(T\)](#) , [OptionFunctions.SomethingOr<T>\(Option<T>, T\)](#) ,  
[OptionFunctions.SomethingOr<T, R>\(Option<T>, Func<T, R>, R\)](#)

# Class O

Namespace: [NullaryThunk.Funky.Core.Options](#)

Assembly: NullaryThunk.Funky.Core.dll

```
public static class O
```

## Inheritance

[object](#) ← O

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Methods

### Nothing<T>()

Creates a [Nothing<T>\(\)](#).

```
public static Nothing<T> Nothing<T>()
```

Returns

[Nothing](#)<T>

An instance of [Nothing<T>\(\)](#)

Type Parameters

T

The underlying type of the meaningful value.

### Somethings<T>(IEnumerable<Option<T>>)

Extracts the underlying values from a sequence of [Option<T>](#) items, returning only the values wrapped in [Something<T>](#)

```
public static IEnumerable<T> Somethings<T>(this IEnumerable<Option<T>> options)
```

Parameters

options [IEnumerable<Option<T>>](#)

A sequence of optional values

Returns

[IEnumerable<T>](#)

A sequence of values.

Type Parameters

T

The underlying type of the meaningful value.

## ToSomething<T>(T)

Lifts a meaningful value into a [Something<T>](#).

```
public static Something<T> ToSomething<T>(this T value)
```

Parameters

value T

The meaningful value.

Returns

[Something<T>](#)

A [Something<T>](#) containing the meaningful value.

## Type Parameters

T

The type of the meaningful value.

# Class OptionFunctions

Namespace: [NullaryThunk.Funky.Core.Options](#)

Assembly: NullaryThunk.Funky.Core.dll

```
public static class OptionFunctions
```

## Inheritance

[object](#) ← OptionFunctions

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Methods

### SomethingOr<T>(Option<T>, T)

```
public static T SomethingOr<T>(this Option<T> opt, T ifNone)
```

#### Parameters

opt [Option](#)<T>

ifNone T

#### Returns

T

#### Type Parameters

T

### SomethingOr<T, R>(Option<T>, Func<T, R>, R)

```
public static R SomethingOr<T, R>(this Option<T> opt, Func<T, R> mapper, R ifNone)
```

## Parameters

opt [Option<T>](#)

mapper [Func<T, R>](#)

ifNone R

## Returns

R

## Type Parameters

T

R

# Interface Option<T>

Namespace: [NullaryThunk.Funky.Core.Options](#)

Assembly: NullaryThunk.Funky.Core.dll

Represents the possibility of a T. An [Option<T>](#) can be either [Something<T>](#) or [Nothing<T>](#). If it is a [Something<T>](#) then it contains a value of . Use pattern matching to match on an [Option<T>](#)

```
public interface Option<out T>
```

## Type Parameters

T

The underlying value type.

## Extension Methods

[O.ToSomething<T>\(I\)](#) , [OptionFunctions.SomethingOr<T>\(Option<T>, T\)](#) ,  
[OptionFunctions.SomethingOr<T, R>\(Option<T>, Func<T, R>, R\)](#)

# Interface Something

Namespace: [NullaryThunk.Funky.Core.Options](#)

Assembly: NullaryThunk.Funky.Core.dll

An [Option<T>](#) that has no meaningful value.

```
public interface Something
```

## Extension Methods

[O.ToSomething<T>\(T\)](#)

# Class Something<T>

Namespace: [NullaryThunk.Funky.Core.Options](#)

Assembly: NullaryThunk.Funky.Core.dll

An [Option<T>](#) that has a meaningful value.

```
public record Something<T> : ISomething<T>, Option<T>,
    Something, IEquatable<Something<T>>
```

## Type Parameters

T

The type of the meaningful value.

## Inheritance

[object](#) ← Something<T>

## Implements

[ISomething<T>](#), [Option<T>](#), [Something](#), [IEquatable<Something<T>>](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Extension Methods

[OptionFunctions.SomethingOr<T>\(Option<T>, T\)](#) ,  
[OptionFunctions.SomethingOr<T, R>\(Option<T>, Func<T, R>, R\)](#) , [O.ToSomething<T>\(T\)](#)

## Constructors

### Something(T)

An [Option<T>](#) that has a meaningful value.

```
public Something(T Value)
```

## Parameters

### **Value T**

The meaningful value.

## Properties

### Value

The meaningful value.

```
public T Value { get; init; }
```

Property Value

T

## Operators

### implicit operator Something<T>(T)

Implicitly converts a value of type T into a [Something<T>](#).

```
public static implicit operator Something<T>(T value)
```

## Parameters

### **value T**

The value to wrap.

## Returns

### [Something<T>](#)

A [Something<T>](#) containing the provided value.

# Namespace NullaryThunk.Funky.Core. Tests.Options

## Classes

[OptionGenerators](#)

[OptionTests](#)

[SomethingOrTests](#)

# Class OptionGenerators

Namespace: [NullaryThunk.Funky.Core.Tests.Options](#)

Assembly: NullaryThunk.Funky.Core.Tests.dll

```
public static class OptionGenerators
```

## Inheritance

[object](#) ← OptionGenerators

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Properties

### Integers

```
public static Gen<int> Integers { get; }
```

Property Value

Gen<[int](#)>

## Methods

### OptionOfInt()

```
public static Arbitrary<(int Value, Option<int> Option)> OptionOfInt()
```

Returns

Arbitrary<([int](#) [Value](#), [Option<int](#)> [Option](#))>

## OptionsOfInt()

```
public static Arbitrary<Option<int>[]> OptionsOfInt()
```

Returns

Arbitrary<[Option<int>\[\]](#)>

# Class OptionTests

Namespace: [NullaryThunk.Funky.Core.Tests.Options](#)

Assembly: NullaryThunk.Funky.Core.Tests.dll

```
[TestFixture]
public class OptionTests
```

## Inheritance

[object](#) ← OptionTests

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Extension Methods

[O.ToSomething<T>\(T\)](#)

## Methods

### ArbitraryOptionIntTest()

```
[Test]
public static void ArbitraryOptionIntTest()
```

### CastingIntToSomethingHasSameValueAsOriginalInt()

```
[Test]
public static void CastingIntToSomethingHasSameValueAsOriginalInt()
```

### SomethingHasAllSomethingsFromOriginal()

```
[Test]
```

```
public static void SomethingsHasAllSomethingsFromOriginal()
```

# Class SomethingOrTests

Namespace: [NullaryThunk.Funky.Core.Tests.Options](#)

Assembly: NullaryThunk.Funky.Core.Tests.dll

```
[TestFixture]
public class SomethingOrTests
```

## Inheritance

[object](#) ← SomethingOrTests

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Extension Methods

[O.ToSomething<T>\(T\)](#)

## Methods

### Nothing\_Yields\_Value\_Of\_IfNone(int, int)

```
[Test]
public static void Nothing_Yields_Value_Of_IfNone(int value, int ifNone)
```

#### Parameters

value [int](#)

ifNone [int](#)

### Something\_Yields\_Value\_Of\_Something(int, int)

```
[Test]
public static void Something_Yields_Value_Of_Something(int value, int ifNone)
```

Parameters

value [int](#)

ifNone [int](#)

## With\_Nothing\_Nullary\_Yields\_Same\_As\_Without(int, int)

[Test]

```
public static void With_Nothing_Nullary_Yields_Same_As_Without(int value,  
int ifNone)
```

Parameters

value [int](#)

ifNone [int](#)

## With\_Something\_Nullary\_Yields\_Same\_As\_Without(int, int)

[Test]

```
public static void With_Something_Nullary_Yields_Same_As_Without(int value,  
int ifNone)
```

Parameters

value [int](#)

ifNone [int](#)