PHP Traits and Class Relationship

Daryl K Wood
CTO DATASHUTTLE.NET

@datashuttle | www.datashuttle.net | www.linkedin.com/in/datashuttle

Module Overview

Introduction to PHP traits

The trait construct

Trait issues

Trait visibilities

Trait multiples

The Trait Construct

Introduced in PHP 5.4

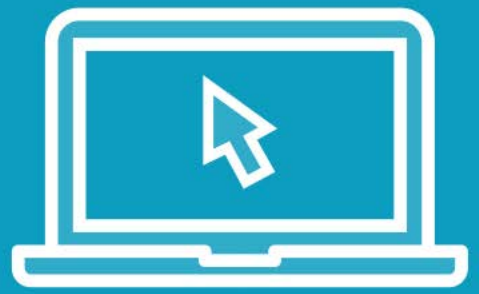
Used in two or more subclass constructs

Not needed if all subclasses need the base class code

An answer for carrying un-needed or duplicate code

```
trait <Name> {}
```

Demo



An airliner trait

PHP Traits

Code written once / used many

A very cool addition to the PHP language

Ahead to trait issues

Trait Issues

Oh NO! Issues, now what?

Understand the full extent of traits

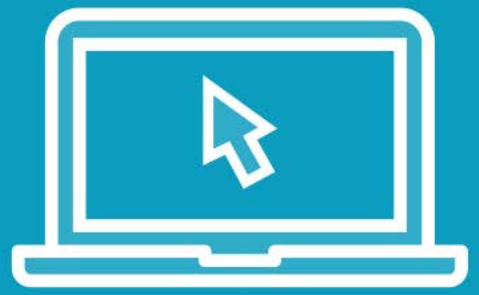
Trait Method Overrides

Subclass methods override superclass methods

Trait methods override superclass methods

Trait code is used with “use” in the subclass

Demo



Trait method overrides in action

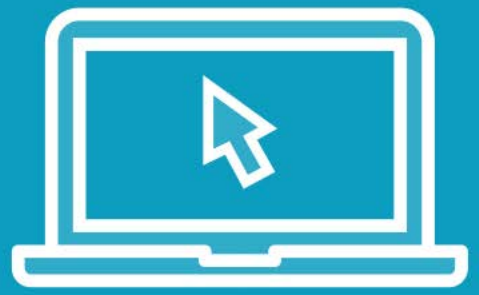
Subclass Method Overrides

What happens with same-named subclass methods?

Subclass same-named methods override trait methods.

Trait methods override superclass methods

Demo



Subclass method overrides in action

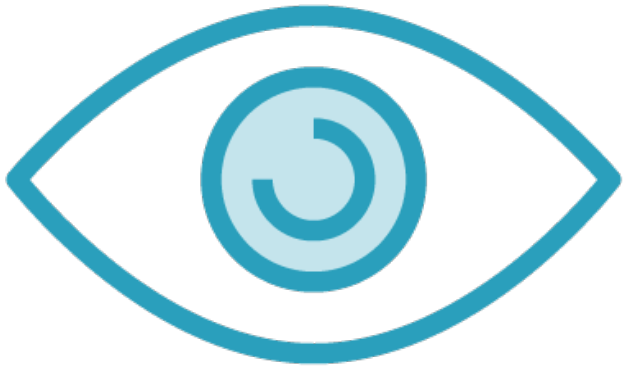
Overrides

Subclass methods override trait methods

Trait methods override superclass methods

Next up: Trait property and method visibilities

Trait Property and Method Visibilities

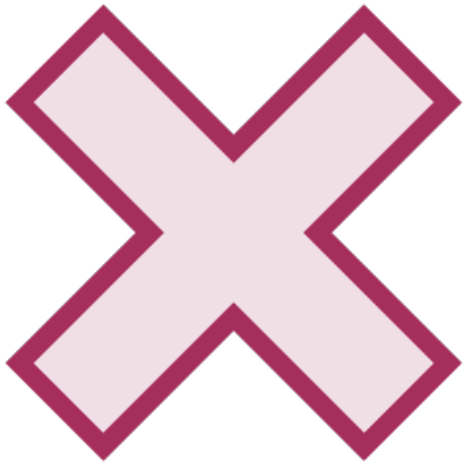


Set to a desired use case

But, it really doesn't matter

Ordinarily, methods overrides can loosen, not tighten visiblities

Method Visibility Restrictions

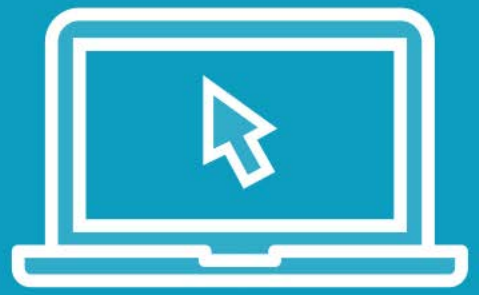


Trait method overrides allow for tightening visibilities

Trait method visibility tightening is desirable

Assigned at the “use” statement

Demo



Trait method visibility tightening in action

Trait Usage

Use traits to remove code duplication

Define your classes narrow

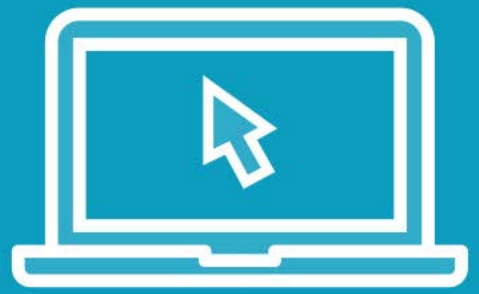
Next up: Trait multiples

Trait Multiples

Use more than one trait in a class

Simple syntax implementation

Demo



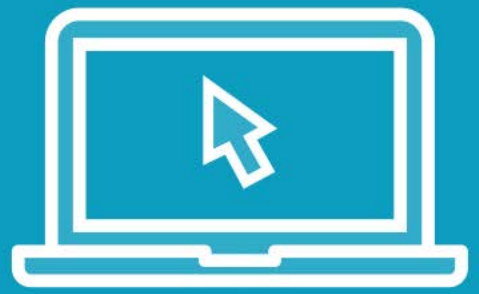
The “BusinessJet” trait

Use with the “Airliner” trait

Instead of Operator

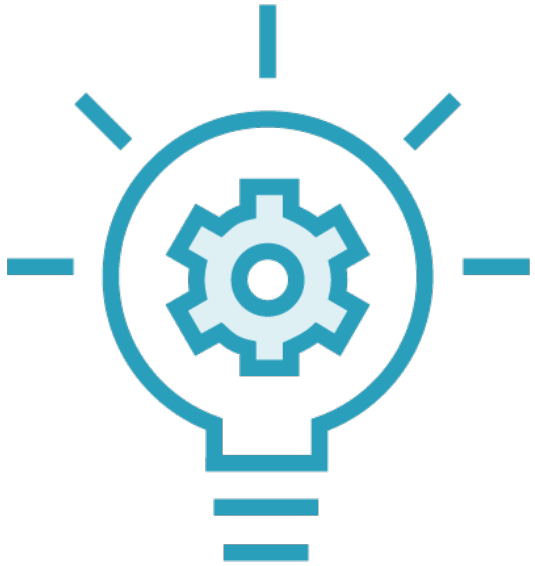
Defines the preferred method

Demo



Back to the “Plane” class

But, Wait!



What if the other trait has valid functionality?

Alias the method

Convoluted?

Trait Use

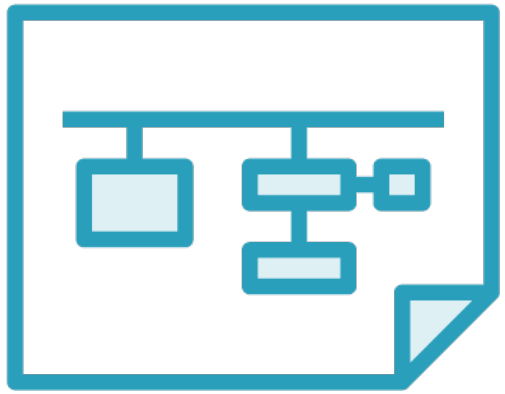
Consider their use carefully

Use to reduce code duplication

Next: traits in traits

Trait static properties and methods

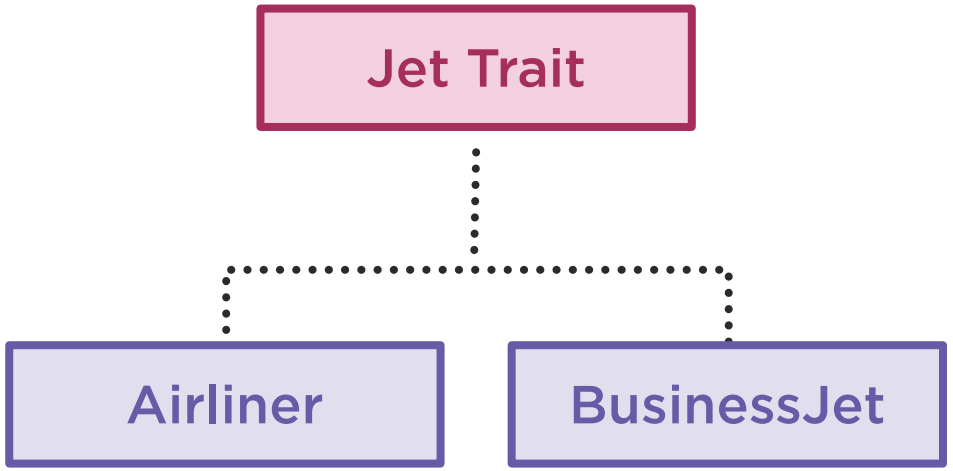
Trait Inheritance



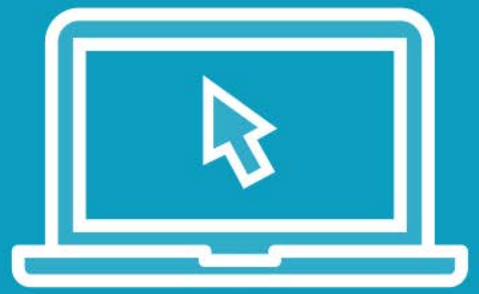
A trait inheritance structure

Traits integrated within other traits

Traits Within Traits



Demo



Integrate the Airliner and BusinessJet traits

Into “Jets”

Care in Trait Use

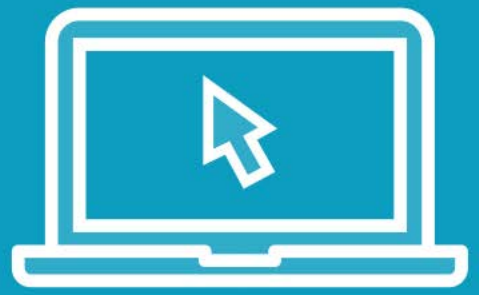
You be the judge

Trait Static Members

Static properties and methods available

No constants – not allowed

Demo



Static members in action

Module Summary

- Eliminate code duplication with traits
- Trait method precedence
- Change trait member visibilities
- Implementing trait multiples
- Trait static properties and methods