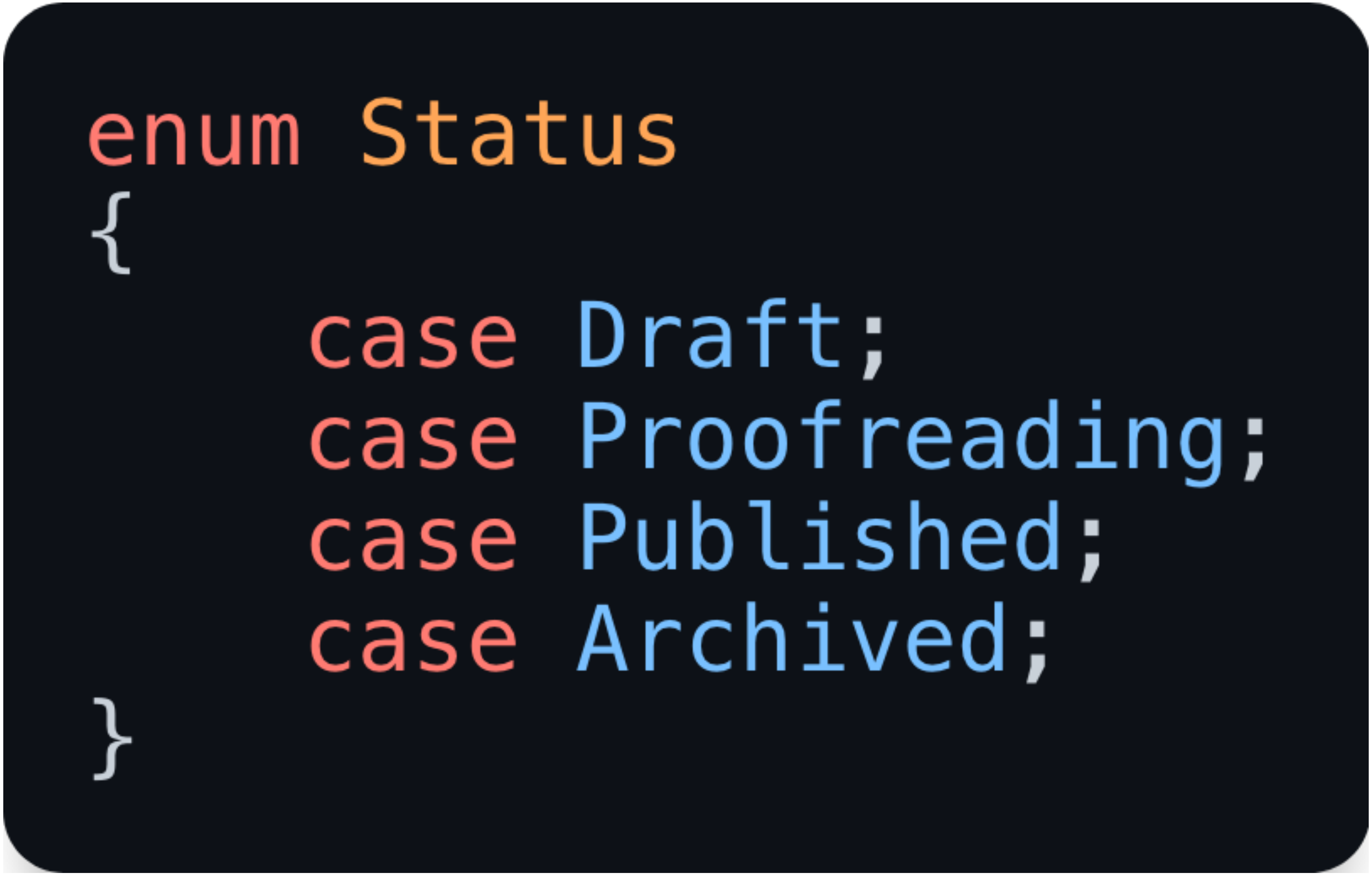
En aften med enumeration

Helt uten matte

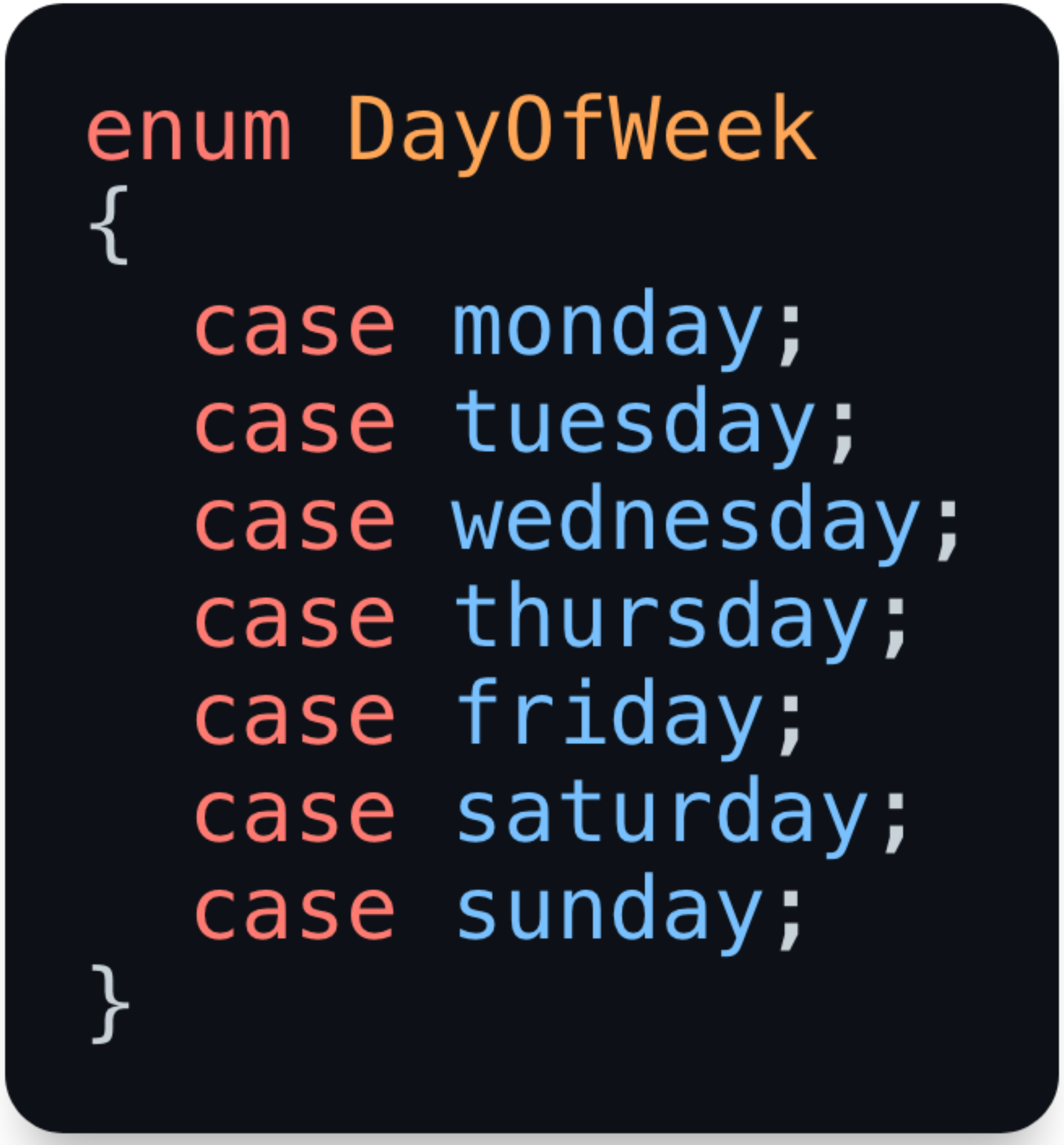
@steinmb 23.02.2023 - Stardate 47634.44

In computer programming, an enumerated type (also called enumeration, enum, or factor in the R programming language, and a categorical variable in statistics) is a data type consisting of a set of named values called elements, members, enumeral, or enumerators of the type. The enumerator names are usually identifiers that behave as constants in the language. An enumerated type can be seen as a degenerate tagged union of unit type. A variable that has been declared as having an enumerated type can be assigned any of the enumerators as a value. In other words, an enumerated type has values that are different from each other, and that can be compared and assigned, but are not specified by the

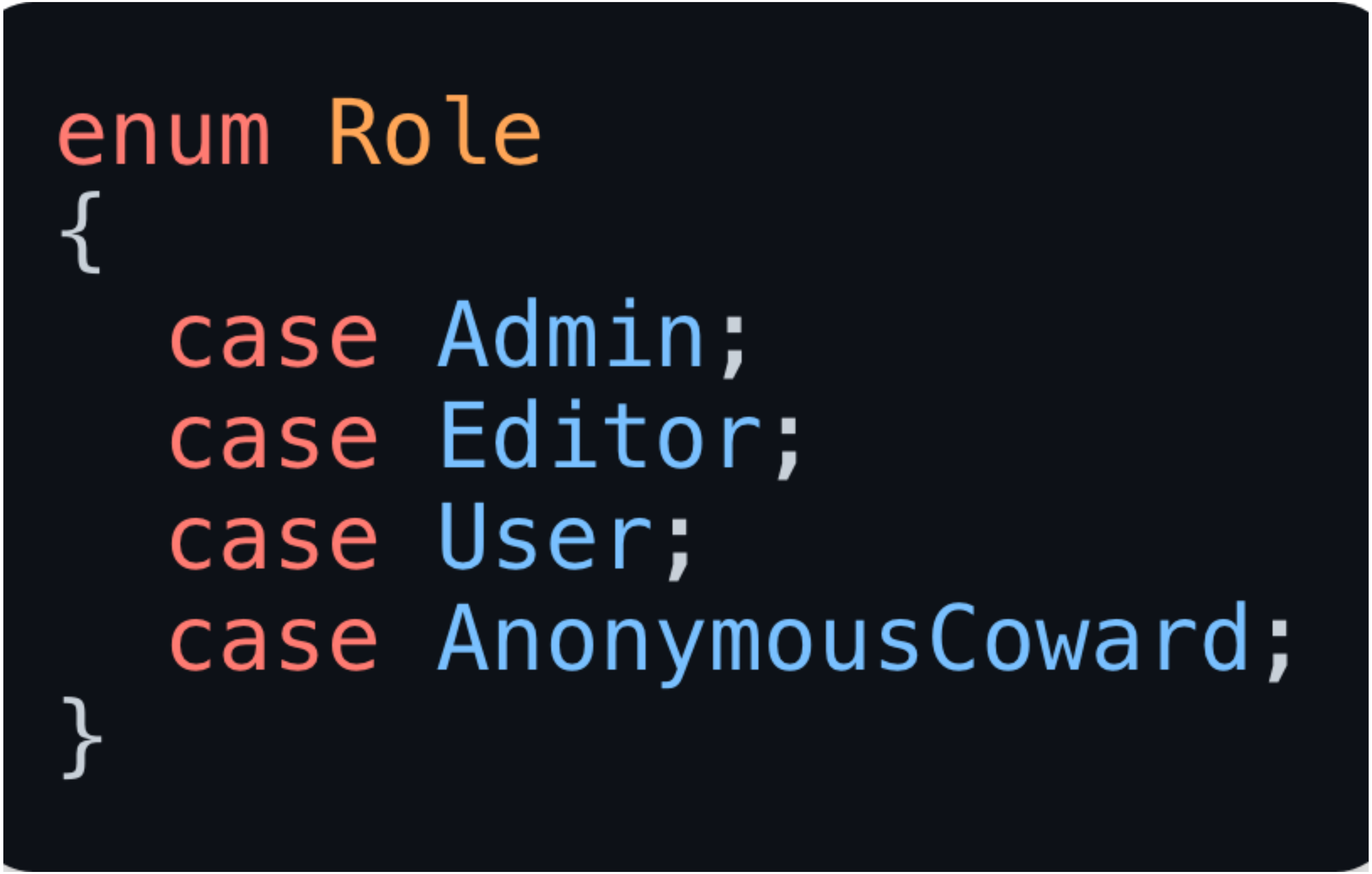
https://en.wikipedia.org/wiki/Enumerated_type



```php
enum Status
{
    case Draft;
    case Proofreading;
    case Published;
    case Archived;
}
```



```
enum DayOfWeek
{
    case monday;
    case tuesday;
    case wednesday;
    case thursday;
    case friday;
    case saturday;
    case sunday;
}
```



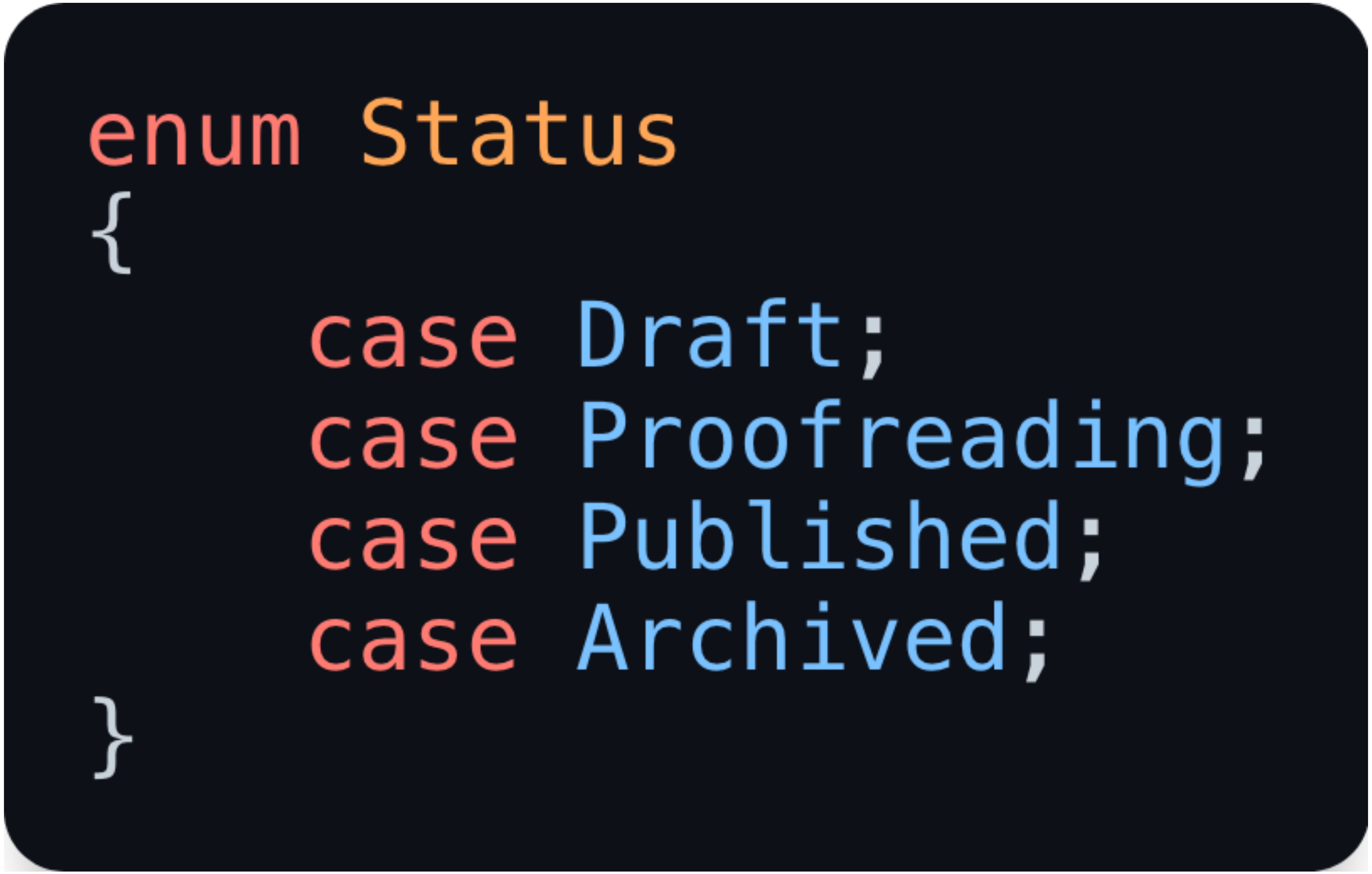
```
enum Role
{
    case Admin;
    case Editor;
    case User;
    case AnonymousCoward;
}
```

```php
enum Role
{
  case Admin;
  case Editor;
  case User;
  case AnonymousCoward;
}

function check_role(Role $neededRole, User $user): bool
{
  return in_array($neededRole->name, $user->roles, true);
}
```

```php
class ClassicStatus
{
    public const Draft = 'Draft';
    public const Proofreading = 'Proofreading';
    public const Published = 'Proofreading';
    public const Archived = 'Proofreading';
}

class BlogPost
{
    public string $title;
    public function __construct(public $status) {}
}

$newPost = new BlogPost(ClassicStatus::Draft);
$myNewPost = new BlogPost('Drfat');
```

```php
enum Status
{
    case Draft;
    case Proofreading;
    case Published;
    case Archived;
}

class BlogPost
{
    public string $title;
    public function __construct(readonly public Status $status) {}
}

$newPost = new BlogPost(Status::Draft);
$myNewPost = new BlogPost('Drfat');
// Uncaught TypeError:
// BlogPost::__construct():
// Argument #1 ($status) must be of type Status, string given
```

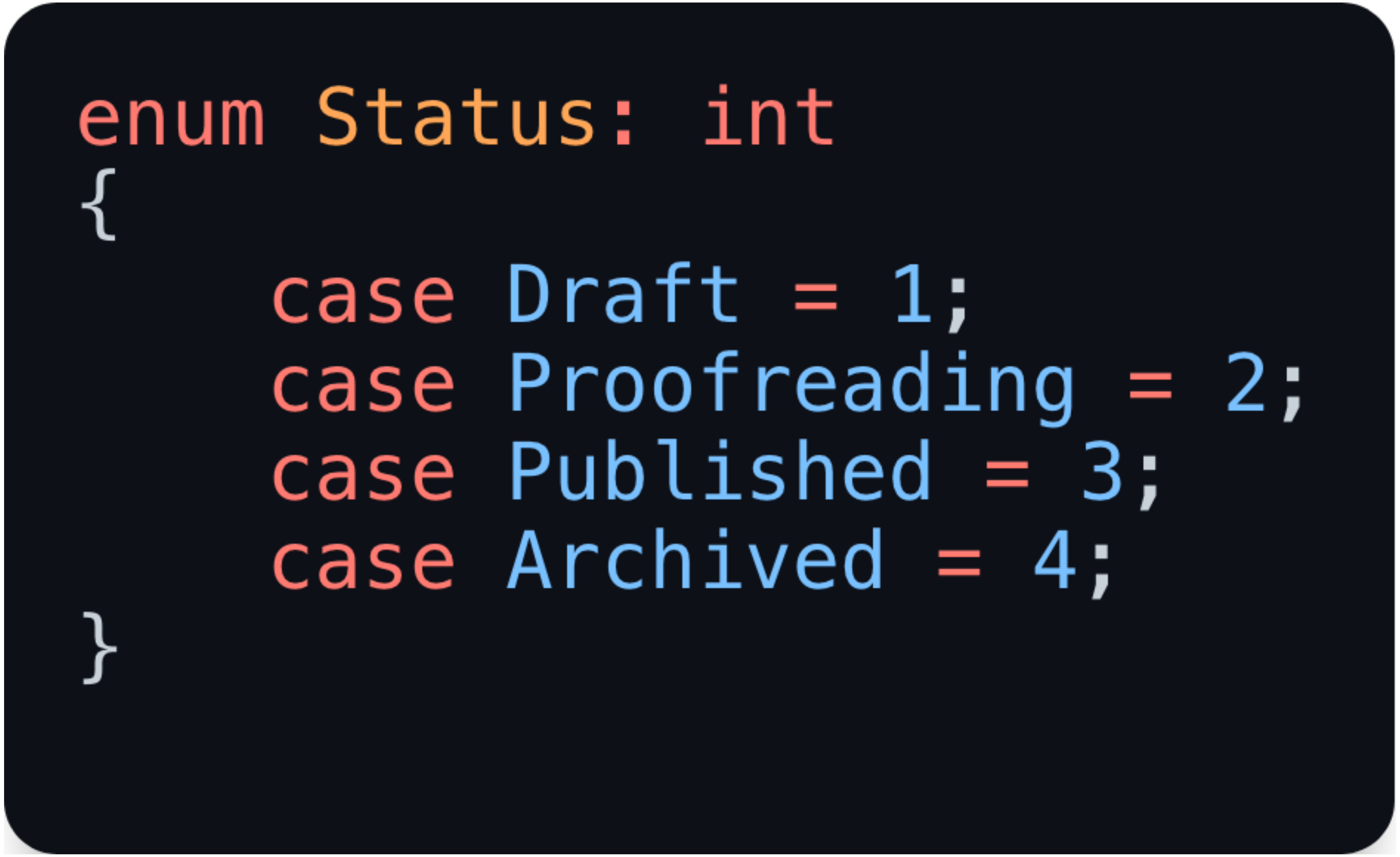
Pure Enums



```
enum Status
{
    case Draft;
    case Proofreading;
    case Published;
    case Archived;
}
```

```php
enum Status: int
{
    case Draft = 1;
    case Proofreading = 2;
    case Published = 3;
    case Archived = 4;
}
```

Backed Enums



```
enum Status: int
{
    case Draft = 1;
    case Proofreading = 2;
    case Published = 3;
    case Archived = 4;
}
```

```php
enum Status: int
{
    case Draft = 1;
    case Proofreading = 2;
    case Published = 3;
    case Archived = 4;
}

var_dump(Status::Archived); // enum(Status::Archived)
var_dump(Status::Archived->value); // int(4)
var_dump(Status::from(4)); // int(4)
var_dump(Status::from(5)); // Uncaught ValueError: 5 is not a valid backing value for enum Status
var_dump(Status::tryFrom(5)); // NULL
```

```php
enum Apple: string {
  case Macintosh = 'M';
  case RedDelicious = 'RD';
  case Honeycrisp = 'H';
}

enum Orange: string {
  case CaraCara = 'CC';
  case Navel = 'N';
  case Mandarin = 'M';
}

var_dump(Apple::Macintosh === Orange::Mandarin); // false
var_dump(Apple::Macintosh->value); // string(1) "M"
var_dump(Apple::Macintosh->name); // string(9) "Macintosh"
var_dump(Apple::cases());
// array(3) {
//   [0]=>
//   enum(Apple::Macintosh)
//   [1]=>
//   enum(Apple::RedDelicious)
//   [2]=>
//   enum(Apple::Honeycrisp)
//}
```

Some theory

Enumerations are a feature of many programming languages. However, we should perhaps say "many languages have a feature called enumerations." What that means in practice has a lot of language-specific semantic nuance

3 types of enums

https://github.com/Crell/enum-comparison

- Syntax sugar over constants
- A restricted-value type unto itself (usually built on top of objects)
- Parameterized algebraic data types

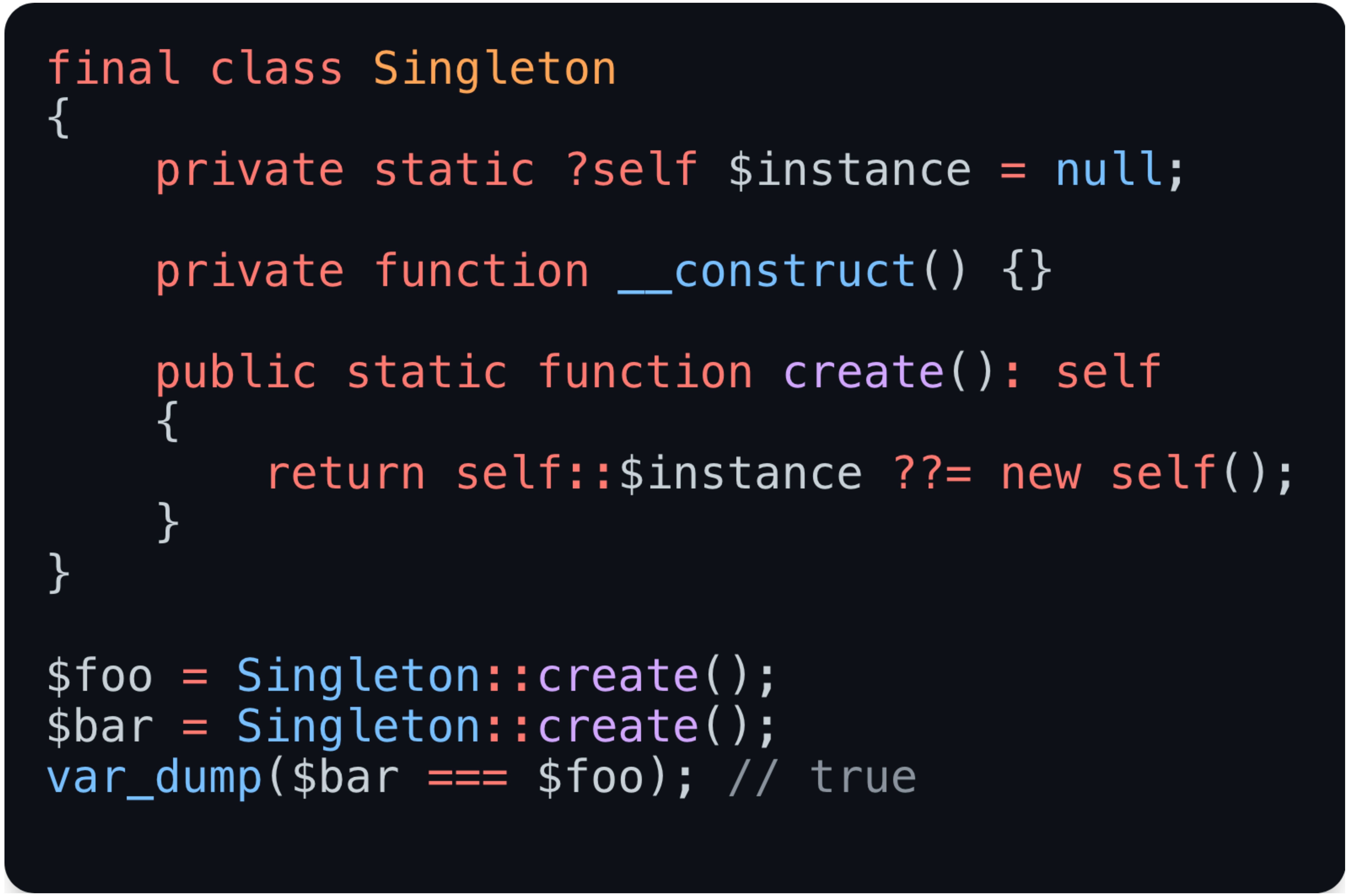
3 types of enums

https://github.com/Crell/enum-comparison

- Fancy Constants: C, Typescript, F#
- Fancy Objects: Python, Java, C#, Scala
- Algebraic Data Types: Haskell, Swift, Rust, Kotlin

PHP Enums in details

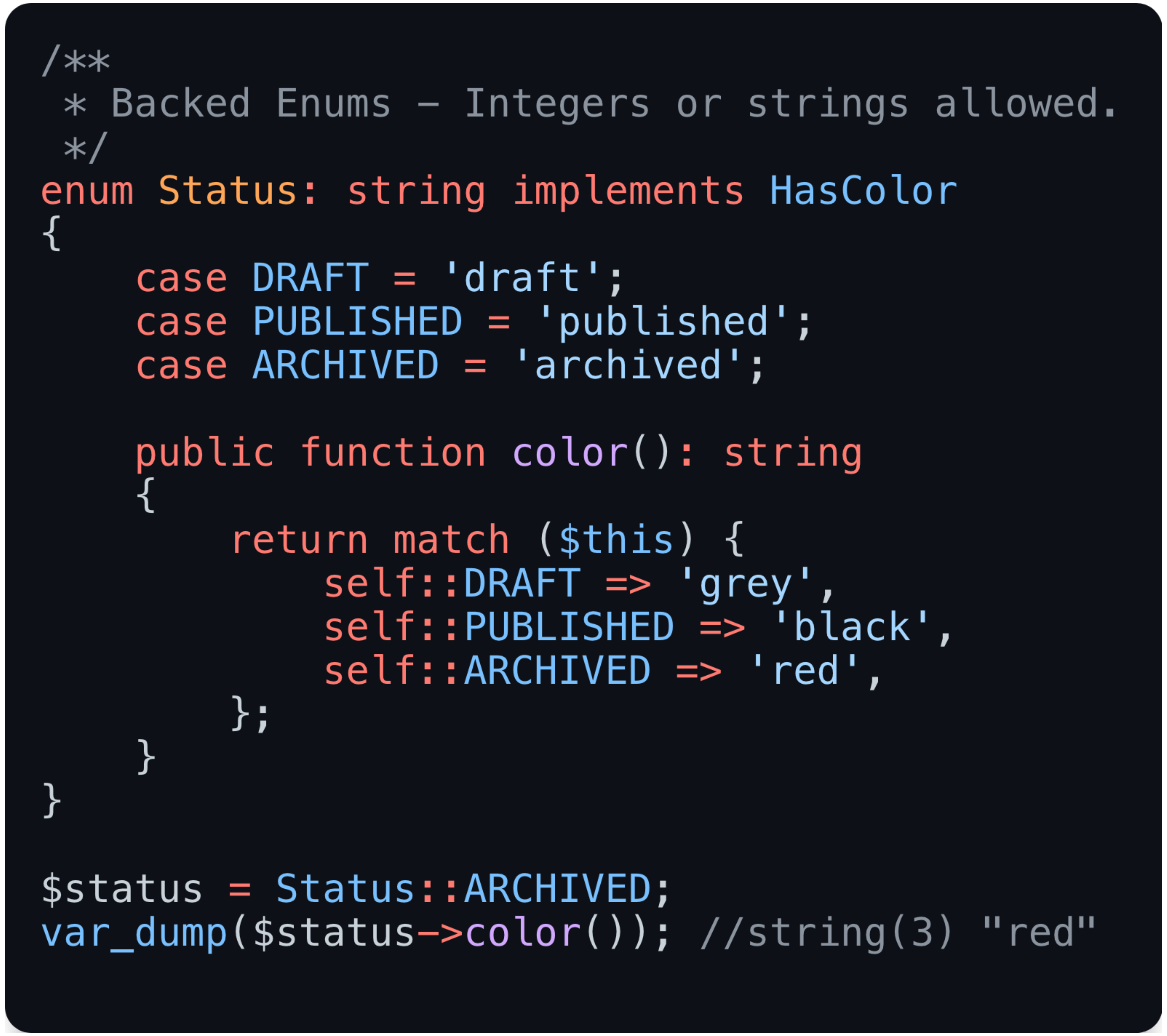
Singletons



```php
final class Singleton
{
    private static ?self $instance = null;

    private function __construct() {}

    public static function create(): self
    {
        return self::$instance ??= new self();
    }
}

$foo = Singleton::create();
$bar = Singleton::create();
var_dump($bar === $foo); // true
```

PHP Enums in details

Support methods



```php
/**
 * Backed Enums - Integers or strings allowed.
 */
enum Status: string implements HasColor
{
    case DRAFT = 'draft';
    case PUBLISHED = 'published';
    case ARCHIVED = 'archived';

    public function color(): string
    {
        return match ($this) {
            self::DRAFT => 'grey',
            self::PUBLISHED => 'black',
            self::ARCHIVED => 'red',
        };
    }
}

$status = Status::ARCHIVED;
var_dump($status->color()); //string(3) "red"
```

When not to use?

Next steps for PHP

https://wiki.php.net/rfc/adts - From Fancy Objects

- Enumerations - The basic unit enumeration type.
- Tagged Unions - Associating values with enum cases to create tagged unions.
- Pattern Matching ''is'' keyword - Pattern matching for objects, enumerations, and similar structures. Mainly to condense conditional logic around the structure of enumerations and value objects.

Spørsmål?

PHP 8.3 - Nov 23 2023

Even more examples

SQL

```php
// These serialized value equivalents
// can be safely concatenated into an SQL string.
enum Order: string
{
    case Asc = 'ASC';
    case Desc = 'DESC';
}

class Query
{
    public function orderBy(string $field, Order $order = Order::Asc) {}
}
```