PHP

VERSIONS

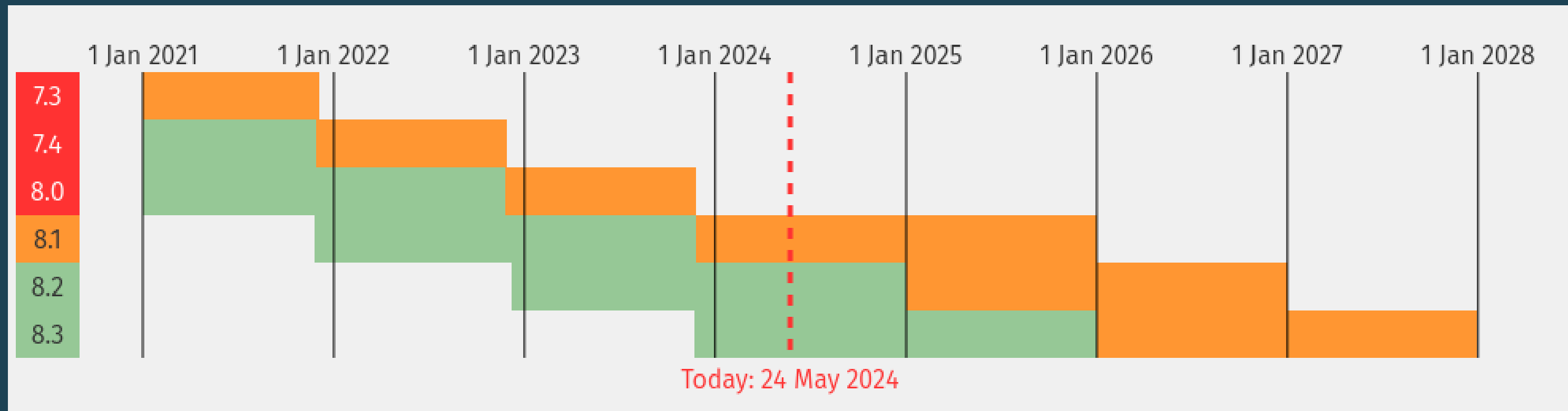|	1 Jan 2021	1 Jan 2022	1 Jan 2023	1 Jan 2024	1 Jan 2025	1 Jan 2026	1 Jan 2027	1 Jan 2028
7.3								
7.4								
8.0								
8.1								
8.2								
8.3								

Today: 24 May 2024

CLASSES – PHP 5

```php
class Exporter5
{
    private $logger;

    public function __construct($logger)
    {
        $this->logger = $logger;
    }

    public function getLogger()
    {
        return $this->logger;
    }
}
```

CLASSES – PHP 7

```php
class Exporter7
{
    private LoggerInterface $logger;

    public function __construct(LoggerInterface $logger)
    {
        $this→logger = $logger;
    }

    public function getLogger(): LoggerInterface
    {
        return $this→logger;
    }
}
```
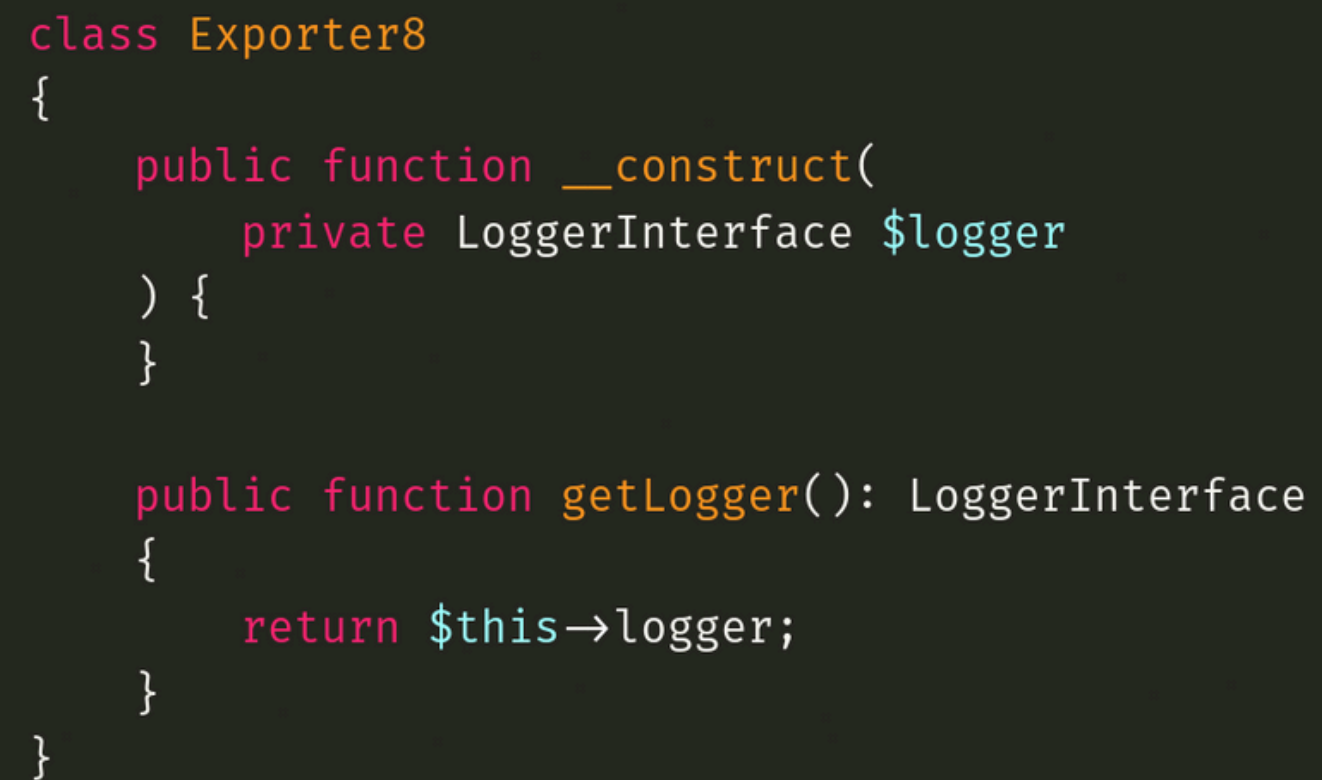
TYPE HINTING

```php
    public function setAttribute(int|float $attribute): self
    {
        return $this;
    }


    public function noReturn(): void
    {
    }
```

CONSTRUCTOR PROPERTY PROMOTION – PHP 8

```php
class Exporter8
{

    public function __construct(
        private LoggerInterface $logger
    ) {
    }


    public function getLogger(): LoggerInterface
    {

        return $this->logger;

    }
}
```

ATTRIBUTS - PHP 8

```php
#[Route('/api/login', name: 'api_login', methods: ['POST'])]
public function index(
    #[CurrentUser]
    ?User $user
): Response {
}
```

NAMESPACES

```php
# PSR 4: cette classe se trouve dans le fichier src/Services/Exporter.php
namespace App\Services;

class Exporter
{
}
```

FULLY QUALIFIED CLASS NAME

```php
var_dump(Exporter::class);
"App\Services\Exporter"
```
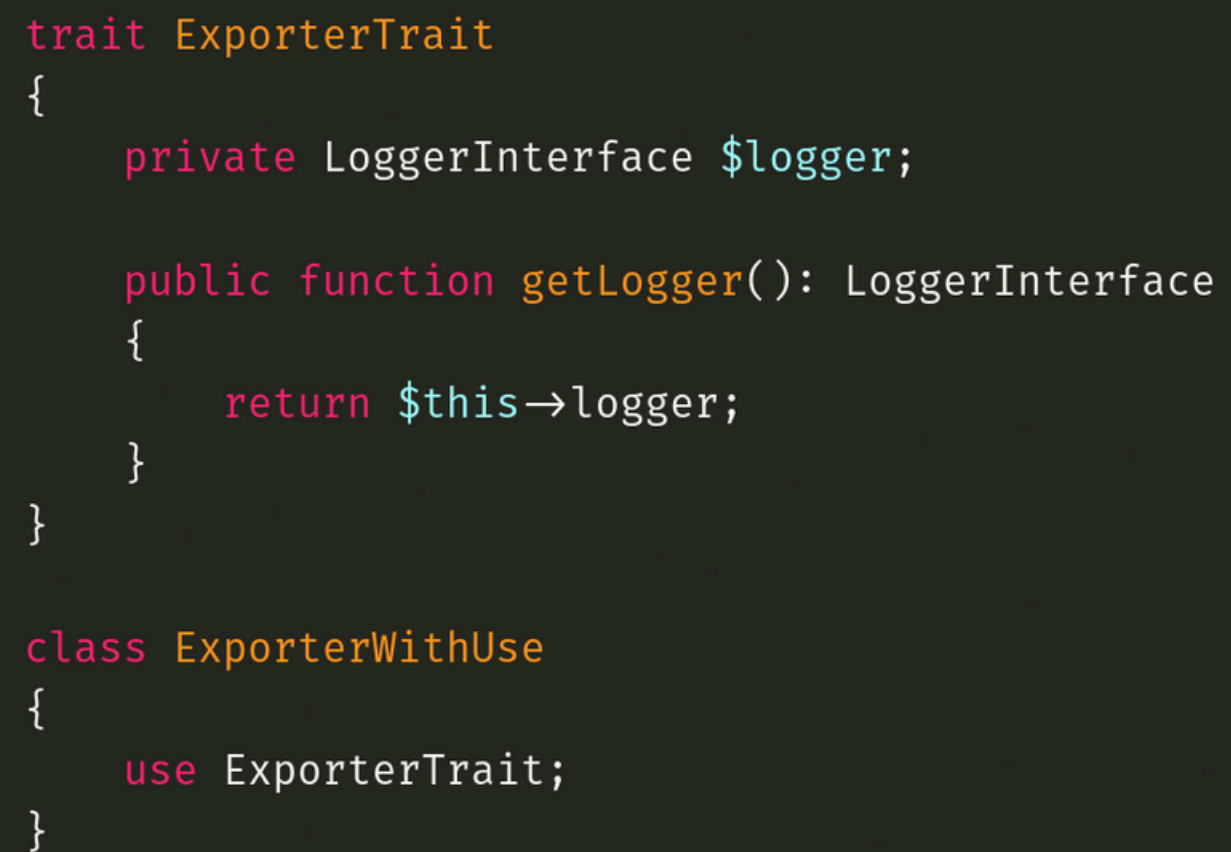
HERITAGE

```php
// Une classe abstraite ne peut être instanciée
abstract class AbstractExporter
{
    protected LoggerInterface $logger;

    public function getLogger(): LoggerInterface
    {
        return $this→logger;
    }
}

// Une classe finale ne peut être étendue
final class ChildExporter extends AbstractExporter
{
}
```

TRAITS

```php
trait ExporterTrait
{
    private LoggerInterface $logger;

    public function getLogger(): LoggerInterface
    {
        return $this→logger;
    }
}


class ExporterWithUse
{
    use ExporterTrait;
}
```

INTERFACES

Contrat: les classes une interface doivent implémenter les
méthodes définies par l'interface

```php
# Interface interne de PHP
interface Stringable {
    public __toString(): string
}
```

INTERFACES

```php
class Article implements Stringable
{
    private string $title;

    public function __toString(): string
    {
        return $this->title;
    }
}
```

EXCEPTIONS

```php
class ExporterException extends Exception
{
}


try {
    // ...
} catch (ExporterException $exception) {
    $this->logger->error($exception->getMessage());
}
```