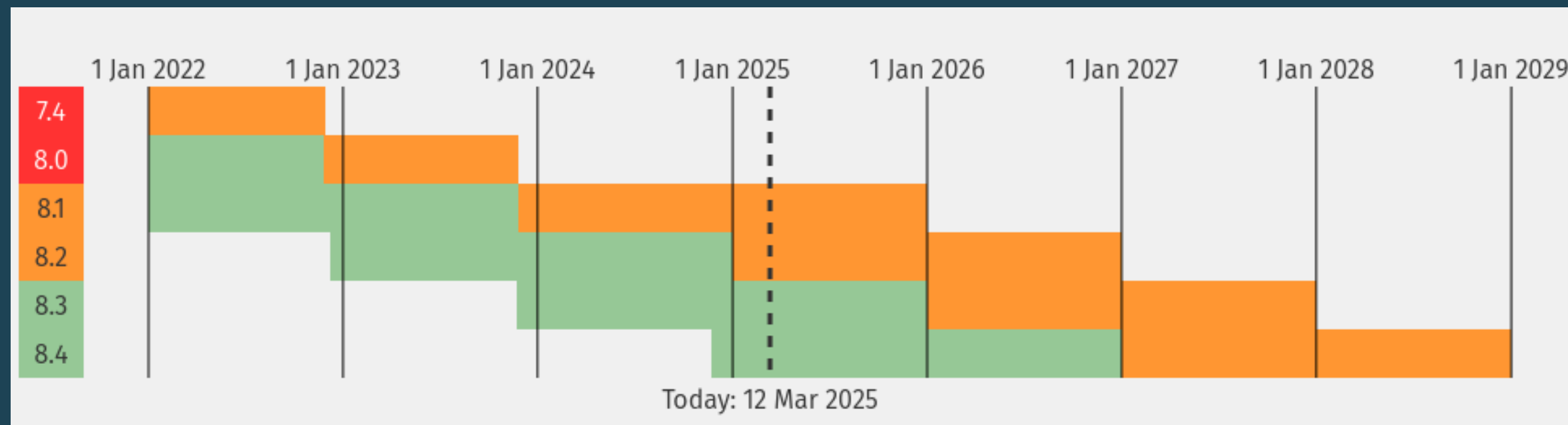


PHP

VERSIONS



CLASSES - PHP 5



```
class Exporter5
{
    private $logger;

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

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

CLASSES - PHP 7

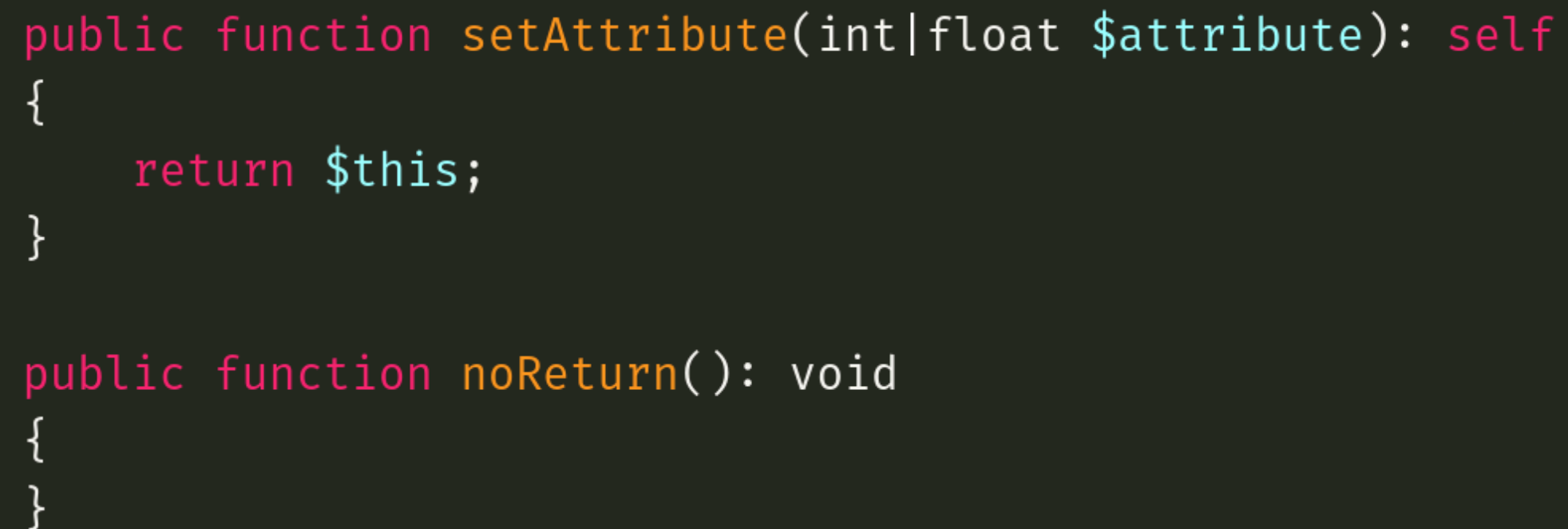


```
class Exporter7
{
    private LoggerInterface $logger;

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

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

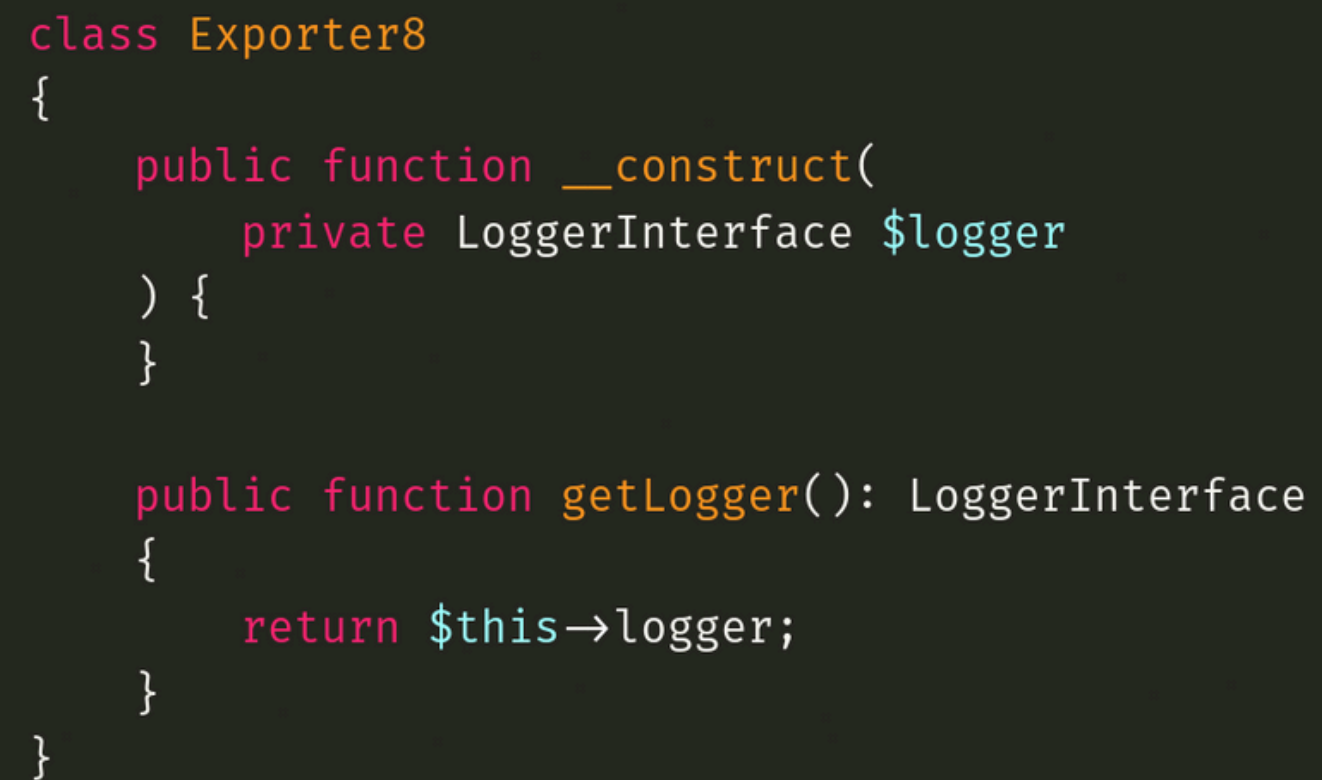
TYPE HINTING



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

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

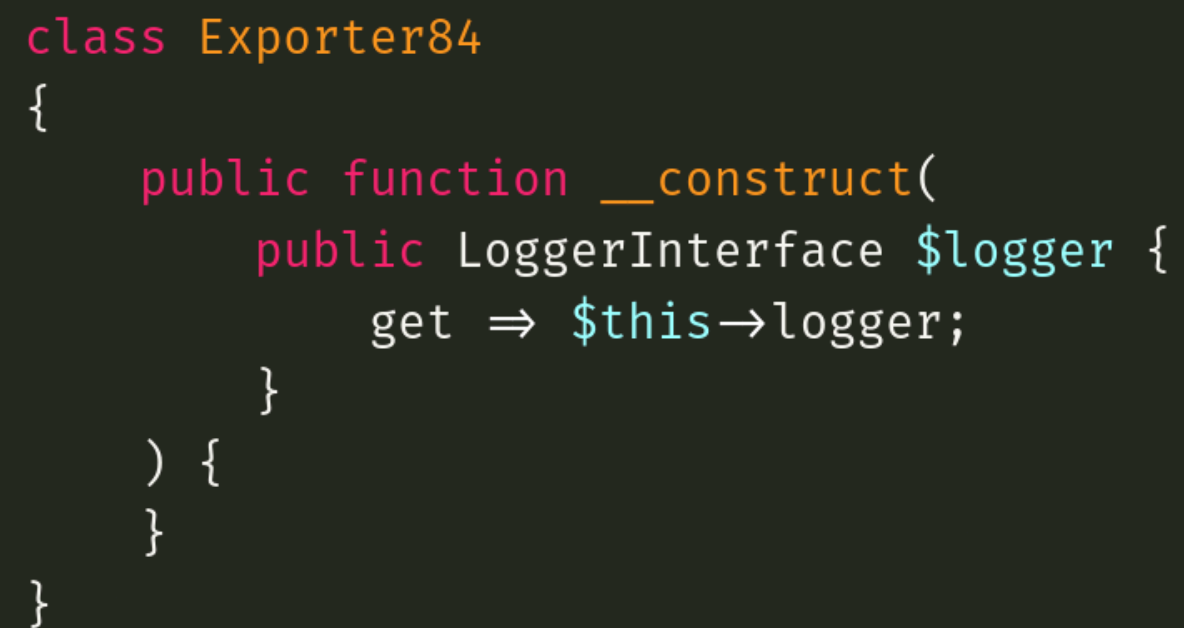
CONSTRUCTOR PROPERTY PROMOTION - PHP 8



```
class Exporter8
{
    public function __construct(
        private LoggerInterface $logger
    ) {
    }

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

PROPERTY HOOKS – PHP 8.4



```
class Exporter84
{
    public function __construct(
        public LoggerInterface $logger {
            get => $this->logger;
        }
    ) {
    }
}
```

ATTRIBUTS - PHP 8



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


NAMESPACES



PSR 4: cette classe se trouve dans le fichier src/Services/Exporter.php

```
namespace App\Services;
```

```
class Exporter  
{  
}
```

FULLY QUALIFIED CLASS NAME



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


HERITAGE

```
// 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



```
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

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

INTERFACES



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

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

EXCEPTIONS



```
class ExporterException extends Exception
{
}

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