Expression Language

Evaluation & Compiling to PHP

```php
use Symfony\Component\ExpressionLanguage\ExpressionLanguage;

$expressionLanguage = new ExpressionLanguage();

var_dump($expressionLanguage->evaluate('1 + 2')); // displays 3

var_dump($expressionLanguage->compile('1 + 2')); // displays (1 + 2)


$apple = new Apple();
$apple->variety = 'Honeycrisp';

var_dump($expressionLanguage->evaluate(
    'fruit.variety',
    array(
        'fruit' => $apple,
    )
));
```

Parsing & Serialization

```php
# parsing
$expression = $expressionLanguage->parse('1 + 4', array());

var_dump($expressionLanguage->evaluate($expression)); // prints 5

# serialization
$expression = new SerializedParsedExpression(
    '1 + 4',
    serialize($expressionLanguage->parse('1 + 4', array())->getNodes())
);

var_dump($expressionLanguage->evaluate($expression)); // prints 5
```

Overview

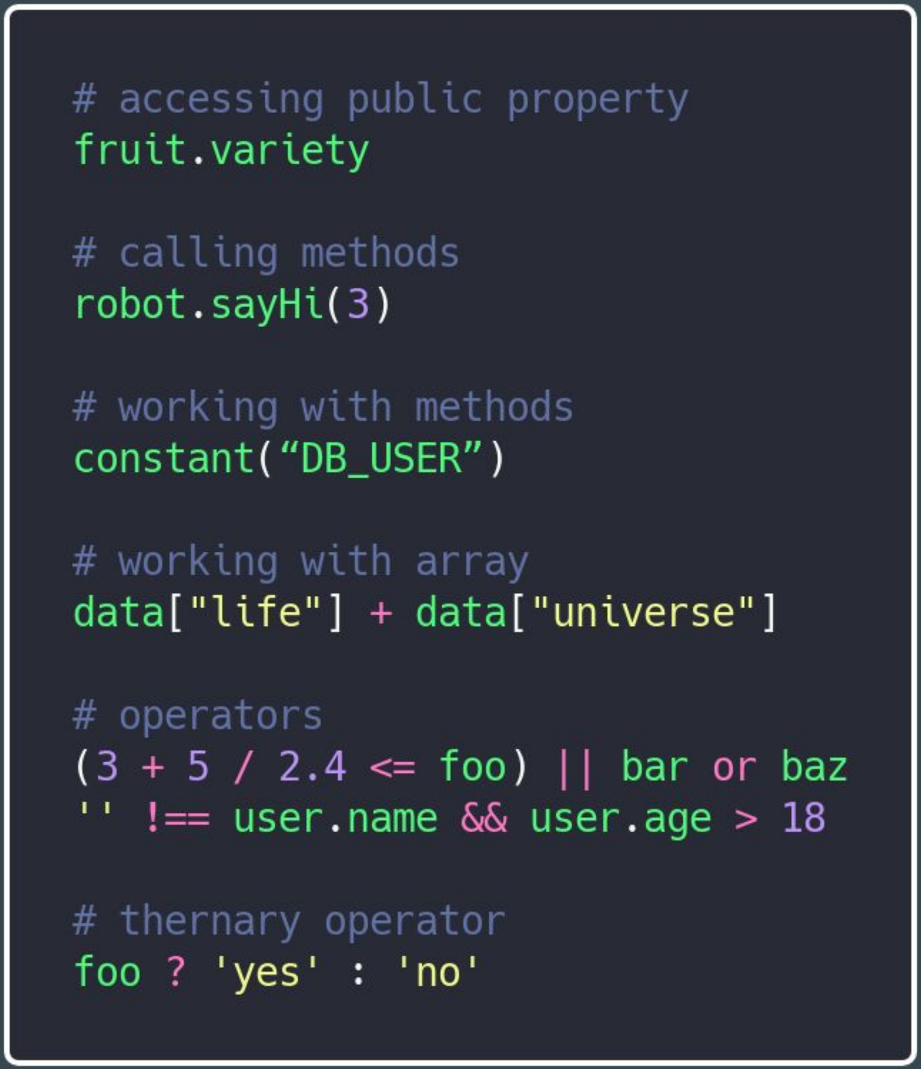
```yaml
# Services
services:
  my_mailer:
    class:     AppBundle\Mailer
    arguments:
    - "@=service('mailer_configuration').getMailerMethod()"
    - "@=container.hasParameter('some_param') ? parameter('some_param') : 'default_value'"

# Validation
AppBundle\Entity\BlogPost:
  constraints:
    - Expression:
        expression: "this.getCategory() in ['php', 'symfony'] or !this.isTechnicalPost()"
        message: "If this is a tech post, the category should be either php or symfony!"

# Routing
contact:
  path:       /contact
  controller: 'App\Controller\DefaultController::contact'
  condition:  "request.headers.get('User-Agent') matches '/firefox/i' and context.getMethod() == 'GET'"

# Layouts
=data["product"]
```

The Expression Syntax

Supported literals

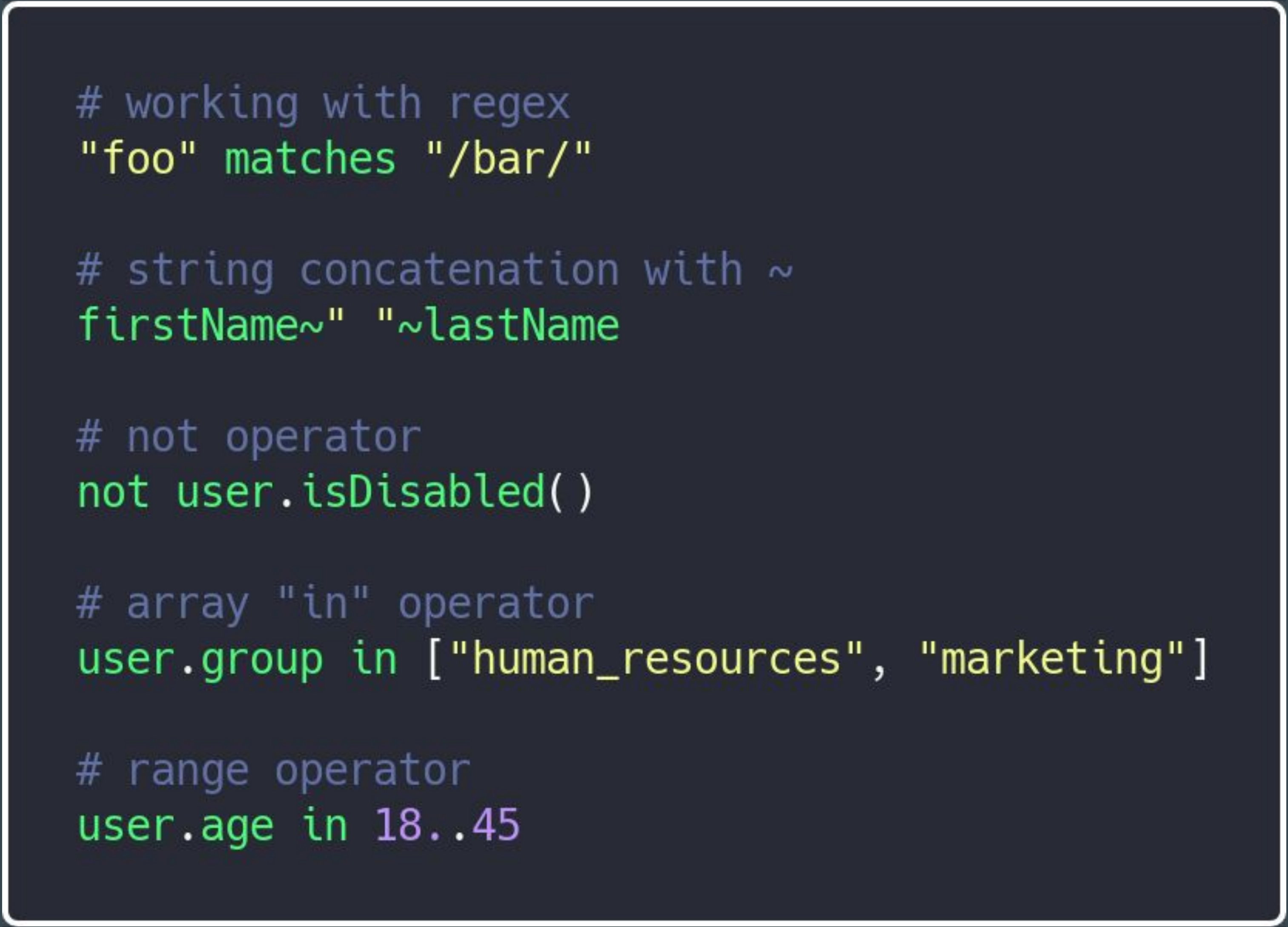
- **strings** - single and double quotes (e.g. 'hello')
- **numbers** - e.g. 103
- **arrays** - using JSON-like notation (e.g. [1, 2])
- **hashes** - using JSON-like notation (e.g. { foo: 'bar' })
- **booleans** - true and false
- **null** - null

Very similar to the JavaScript



```
# accessing public property
fruit.variety

# calling methods
robot.sayHi(3)

# working with methods
constant(“DB_USER”)

# working with array
data["life"] + data["universe"]

# operators
(3 + 5 / 2.4 <= foo) || bar or baz
'' !== user.name && user.age > 18

# thernary operator
foo ? 'yes' : 'no'
```

Sometimes it’s different from the JavaScript



```
# working with regex
"foo" matches "/bar/"

# string concatenation with ~
firstName~" "~lastName

# not operator
not user.isDisabled()

# array "in" operator
user.group in ["human_resources", "marketing"]

# range operator
user.age in 18..45
```

Extending

- Extending the Context
  - Almost not possible
- Extending with Functions
  - LoggedInExpressionLanguageProvider
  - Services.yml
  - ExpressionFunction::fromPhp('strtoupper');

routing.expression_language_provider

security.expression_language_provider

$container->addExpressionLanguageProvider($provider);