Un ensemble de bibliothèques PHP

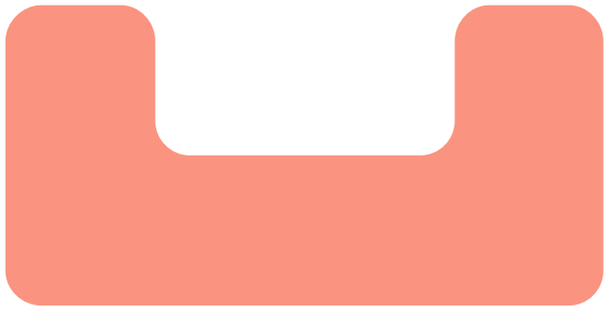
15 Février 2018

Qui suis-je ?

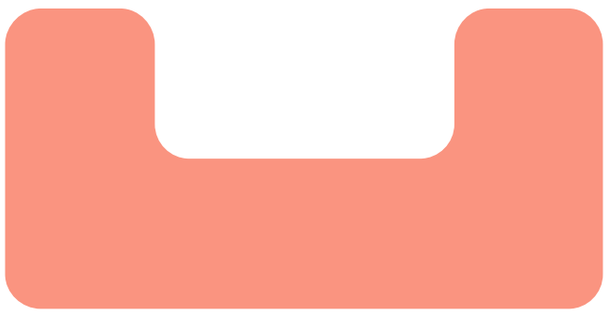
- Alexis von Glasow
-  @ashgenesis
- https://matters.tech
- contributeur:

http://atoum.org 

https://hoa-project.net 








~ 60

Bibliothèques

Acl	Dispatcher	Iterator	Notification	Ruler	Ustring
Bench	Dns	Json	Option	Serialize	View
Cache	Event	Kitab	Praspel	Session	Visitor
Cli	Eventsource	Locale	Promise	Socket	Websocket
Compiler	Exception	Log	Protocol	Stream	Worker
Consistency	Fastcgi	Mail	Prototype	String	Xml
Console	File	Math	Realdom	Stringbuffer	Xmlrpc
Core	Graph	Memory	Regex	Test	Xyl
Database	Http	Mime	Registry	Translate	Zformat
Devtools	Irc	Model	Router	Tree	Zombie

Bibliothèques

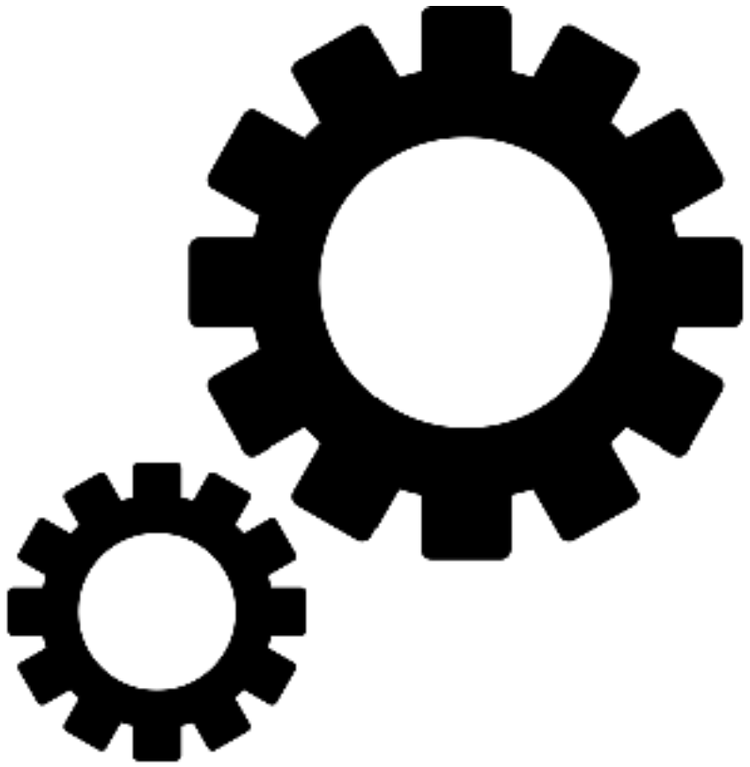
Acl	Dispatcher	Iterator	Notification	**Ruler**	Ustring
Bench	Dns	Json	**Option**	Serialize	View
Cache	Event	**Kitab**	**Praspel**	Session	Visitor
Cli	Eventsource	Locale	Promise	Socket	**Websocket**
Compiler	Exception	Log	Protocol	Stream	Worker
Consistency	Fastcgi	Mail	Prototype	String	Xml
Console	File	Math	Realdom	Stringbuffer	Xmlrpc
Core	Graph	Memory	Regex	Test	Xyl
Database	Http	Mime	Registry	Translate	Zformat
Devtools	Irc	Model	Router	Tree	Zombie

Bibliothèques

Acl	Dispatcher	Iterator	Notification	**Ruler**	Ustring
Bench	Dns	Json	**Option**	Serialize	View
Cache	Event	**Kitab**	**Praspel**	Session	Visitor
Cli	Eventsource	Locale	Promise	Socket	**Websocket**
Compiler	Exception	Log	Protocol	Stream	Worker
Consistency	Fastcgi	Mail	Prototype	String	Xml
Console	File	Math	Realdom	Stringbuffer	Xmlrpc
Core	Graph	Memory	Regex	Test	Xyl
Database	Http	Mime	Registry	Translate	Zformat
Devtools	Irc	Model	Router	Tree	Zombie

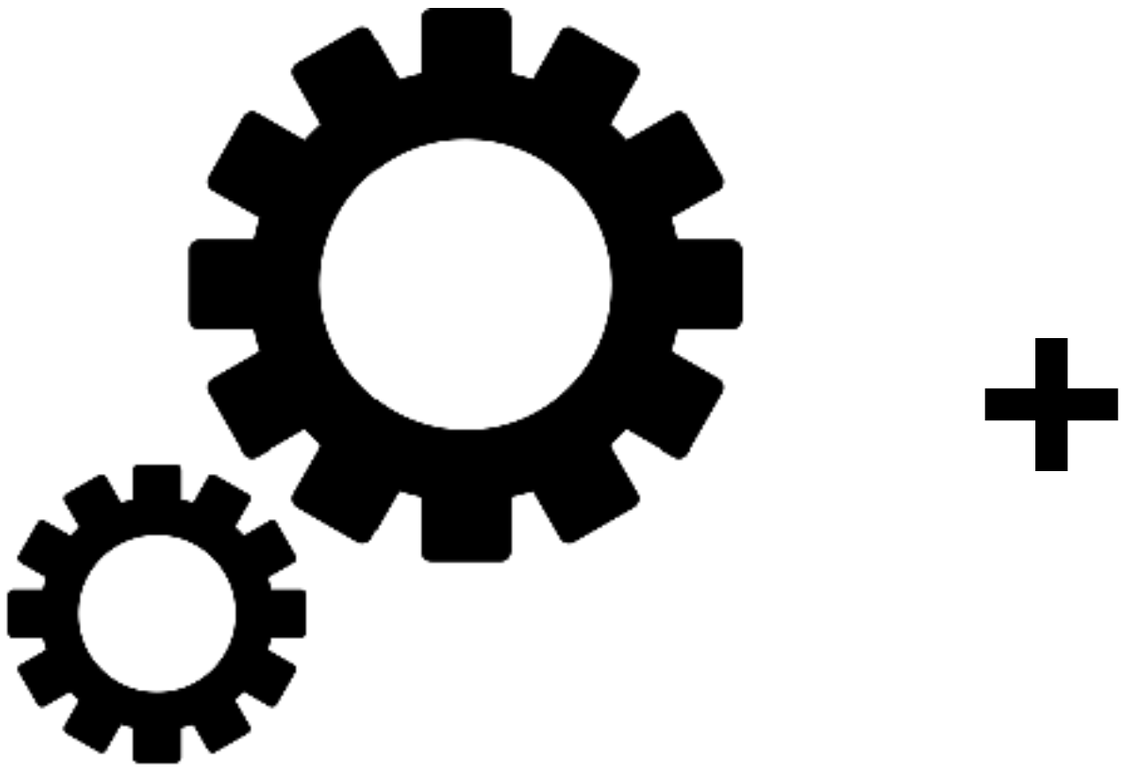
Compiler

Compiler



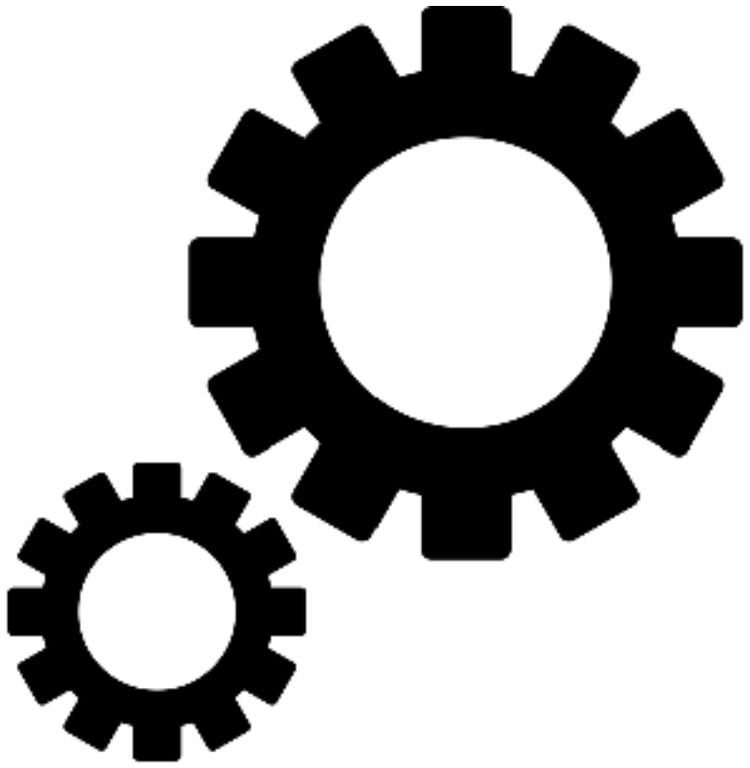
credits: Rodrigo Álvarez Virgós from noun project

Compiler



credits: Rodrigo Álvarez Virgós from noun project

Compiler



+ **Grammaire**

credits: Rodrigo Álvarez Virgós from noun project

Compiler

Grammaire

Compiler

Grammaire

=

Compiler

Grammaire

=

Lexèmes (tokens)

Compiler

Grammaire

=

Lexèmes (tokens)

+

Compiler

Grammaire

=

Lexèmes (tokens)

+

Règles

Compiler

```
%token      T_OP_PLUS       plus
%token      T_OP_MINUS      moins
%token      T_OP_MULTI      multiplié par
%token      T_OP_DIVIDE     divisé par
%token      T_OP_EQUAL      égale
%token      T_FN            \.[a-zA-Z_][a-zA-Z0-9_]*
%token      T_VAR           @[a-zA-Z_][a-zA-Z0-9_]*
%token      T_CONST         [A-Z_][A-Z0-9_]*
%token      T_NUMBER        \-?[1-9][0-9]*
%token      T_OPEN_PAREN    \(
%token      T_CLOSE_PAREN   \)
%token      T_COMMA         ,
```

Compiler

```
%token      T_OP_PLUS       plus
%token      T_OP_MINUS      moins
%token      T_OP_MULTI      multiplié par
%token      T_OP_DIVIDE     divisé par
%token      T_OP_EQUAL      égale
%token      T_FN            \.[a-zA-Z_][a-zA-Z0-9_]*
%token      T_VAR           @[a-zA-Z_][a-zA-Z0-9_]*
%token      T_CONST         [A-Z_][A-Z0-9_]*
%token      T_NUMBER        \-?[1-9][0-9]*
%token      T_OPEN_PAREN    \(
%token      T_CLOSE_PAREN   \)
%token      T_COMMA         ,
```

Compiler

```
%token      T_OP_PLUS       plus
%token      T_OP_MINUS      moins
%token      T_OP_MULTI      multiplié par
%token      T_OP_DIVIDE     divisé par
%token      T_OP_EQUAL      égale
%token      T_FN            \.[a-zA-Z_][a-zA-Z0-9_]*
%token      T_VAR           @[a-zA-Z_][a-zA-Z0-9_]*
%token      T_CONST         [A-Z_][A-Z0-9_]*
%token      T_NUMBER        \-?[1-9][0-9]*
%token      T_OPEN_PAREN    \(
%token      T_CLOSE_PAREN   \)
%token      T_COMMA         ,
```

Compiler

```
%token      T_OP_PLUS      plus
%token      T_OP_MINUS     moins
%token      T_OP_MULTI     multiplié par
%token      T_OP_DIVIDE    divisé par
%token      T_OP_EQUAL     égale
%token      T_FN           \.[a-zA-Z_][a-zA-Z0-9_]*
%token      T_VAR          @[a-zA-Z_][a-zA-Z0-9_]*
%token      T_CONST        [A-Z_][A-Z0-9_]*
%token      T_NUMBER       \-?[1-9][0-9]*
%token      T_OPEN_PAREN   \(
%token      T_CLOSE_PAREN  \)
%token      T_COMMA        ,
```

Compiler

```
#tml:
    ( fn() | expr() | assign() | str() )+
#expr:
    ( <T_NUMBER> | <T_CONST> | rvar() | fn() )
( operator() expr() )?
#assign:
    lvar() ::T_OP_EQUAL:: rval()
#fn:
    <T_FN> ::T_OPEN_PAREN:: arguments() ::T_CLOSE_PAREN::
#str:
    ::T_QUOTE:: <T_CHAR>* ::T_QUOTE::
#rvar:
    <T_VAR>
#lvar:
    <T_VAR>
```

Compiler

```
#tml:
    ( fn() | expr() | assign() | str() )+
#expr:
    ( <T_NUMBER> | <T_CONST> | rvar() | fn() )
( operator() expr() )?
#assign:
    lvar() ::T_OP_EQUAL:: rval()
#fn:
    <T_FN> ::T_OPEN_PAREN:: arguments() ::T_CLOSE_PAREN::
#str:
    ::T_QUOTE:: <T_CHAR>* ::T_QUOTE::
#rvar:
    <T_VAR>
#lvar:
    <T_VAR>
```

Compiler

```
#tml:
     ( fn() | expr() | assign() | str() )+
#expr:
     ( <T_NUMBER> | <T_CONST> | rvar() | fn() )
( operator() expr() )?
#assign:
     lvar() ::T_OP_EQUAL:: rval()
#fn:
     <T_FN> ::T_OPEN_PAREN:: arguments() ::T_CLOSE_PAREN::
#str:
     ::T_QUOTE:: <T_CHAR>* ::T_QUOTE::
#rvar:
     <T_VAR>
#lvar:
     <T_VAR>
```

10 minutes language

```
@neuf égal 4 plus 5
@trois égal @neuf divisé par 3
@mon_age égal @trois multiplié par 10 plus 6

affiche J'ai @mon_age ans 😅
```

10 minutes language

```
@neuf égal 4 plus 5
@trois égal @neuf divisé par 3
@mon_age égal @trois multiplié par 10 plus 6

affiche J'ai @mon_age ans 😅
```

```
J'ai 36 ans 😅
```

Bibliothèques

Acl	Dispatcher	Iterator	Notification	**Ruler**	Ustring
Bench	Dns	Json	**Option**	Serialize	View
Cache	Event	**Kitab**	**Praspel**	Session	Visitor
Cli	Eventsource	Locale	Promise	Socket	**Websocket**
Compiler	Exception	Log	Protocol	Stream	Worker
Consistency	Fastcgi	Mail	Prototype	String	Xml
Console	File	Math	Realdom	Stringbuffer	Xmlrpc
Core	Graph	Memory	Regex	Test	Xyl
Database	Http	Mime	Registry	Translate	Zformat
Devtools	Irc	Model	Router	Tree	Zombie

Bibliothèques

Acl	Dispatcher	Iterator	Notification	**Ruler**	Ustring
Bench	Dns	Json	**Option**	Serialize	View
Cache	Event	**Kitab**	**Praspel**	Session	Visitor
Cli	Eventsource	Locale	Promise	Socket	**Websocket**
Compiler	Exception	Log	Protocol	Stream	Worker
Consistency	Fastcgi	Mail	Prototype	String	Xml
Console	File	Math	Realdom	Stringbuffer	Xmlrpc
Core	Graph	Memory	Regex	Test	Xyl
Database	Http	Mime	Registry	Translate	Zformat
Devtools	Irc	Model	Router	Tree	Zombie

Console

```
$functions = get_defined_functions();

$readline->setAutocompleter(
    new Hoa\Console\Readline\Autocompleter\Aggregate([
        new Hoa\Console\Readline\Autocompleter\Path(),
        new Hoa\Console\Readline\Autocompleter\Word(
            $functions['internal'])
        ]
    )
);
```

Console

```
$functions = get_defined_functions();

$readline->setAutocompleter(
    new Hoa\Console\Readline\Autocompleter\Aggregate([
        new Hoa\Console\Readline\Autocompleter\Path(),
        new Hoa\Console\Readline\Autocompleter\Word(
            $functions['internal'])
        ]
    )
);
```

Console

```
$functions = get_defined_functions();
```

```
$readline->setAutocompleter(
    new Hoa\Console\Readline\Autocompleter\Aggregate([
        new Hoa\Console\Readline\Autocompleter\Path(),
        new Hoa\Console\Readline\Autocompleter\Word(
            $functions['internal'])
        ]
    )
);
```

Console

https://github.com/hoaproject/Console#readline

Console

https://github.com/hoaproject/Console#readline

Console

```
$options = new Hoa\Console\GetOption(
    array(
        array('colonnes',  Hoa\Console\GetOption::REQUIRED_ARGUMENT, 'y'),
        array('lines',  Hoa\Console\GetOption::REQUIRED_ARGUMENT, 'x'),
        array('random',   Hoa\Console\GetOption::OPTIONAL_ARGUMENT, 'r'),
        array('glider-gun', Hoa\Console\GetOption::OPTIONAL_ARGUMENT, 'g'),
        array('help',   Hoa\Console\GetOption::OPTIONAL_ARGUMENT, 'h'),
    ),
    $parser
);
```

Console

```
Hoa\Console\Cursor::clear('↕');
Hoa\Console\Cursor::hide();
Hoa\Console\Cursor::move('↓', 1);

do {
    $hash = $this->computeHash();
    $this->displayWorld();
    $this->computeNewState();
} while ($hash !== $this->computeHash());

Hoa\Console\Cursor::show();
Hoa\Console\Cursor::colorize('default');
Hoa\Console\Cursor::move(
    '↓',
    $this->universe->getWidth()
);
```

Console

```
Hoa\Console\Cursor::clear('↕');
Hoa\Console\Cursor::hide();
Hoa\Console\Cursor::move('↓', 1);

do {
    $hash = $this->computeHash();
    $this->displayWorld();
    $this->computeNewState();
} while ($hash !== $this->computeHash());

Hoa\Console\Cursor::show();
Hoa\Console\Cursor::colorize('default');
Hoa\Console\Cursor::move(
    '↓',
    $this->universe->getWidth()
);
```

Console

```
Hoa\Console\Cursor::clear('↕');
Hoa\Console\Cursor::hide();
Hoa\Console\Cursor::move('↓', 1);

do {
    $hash = $this->computeHash();
    $this->displayWorld();
    $this->computeNewState();
} while ($hash !== $this->computeHash());

Hoa\Console\Cursor::show();
Hoa\Console\Cursor::colorize('default');
Hoa\Console\Cursor::move(
    '↓',
    $this->universe->getWidth()
);
```

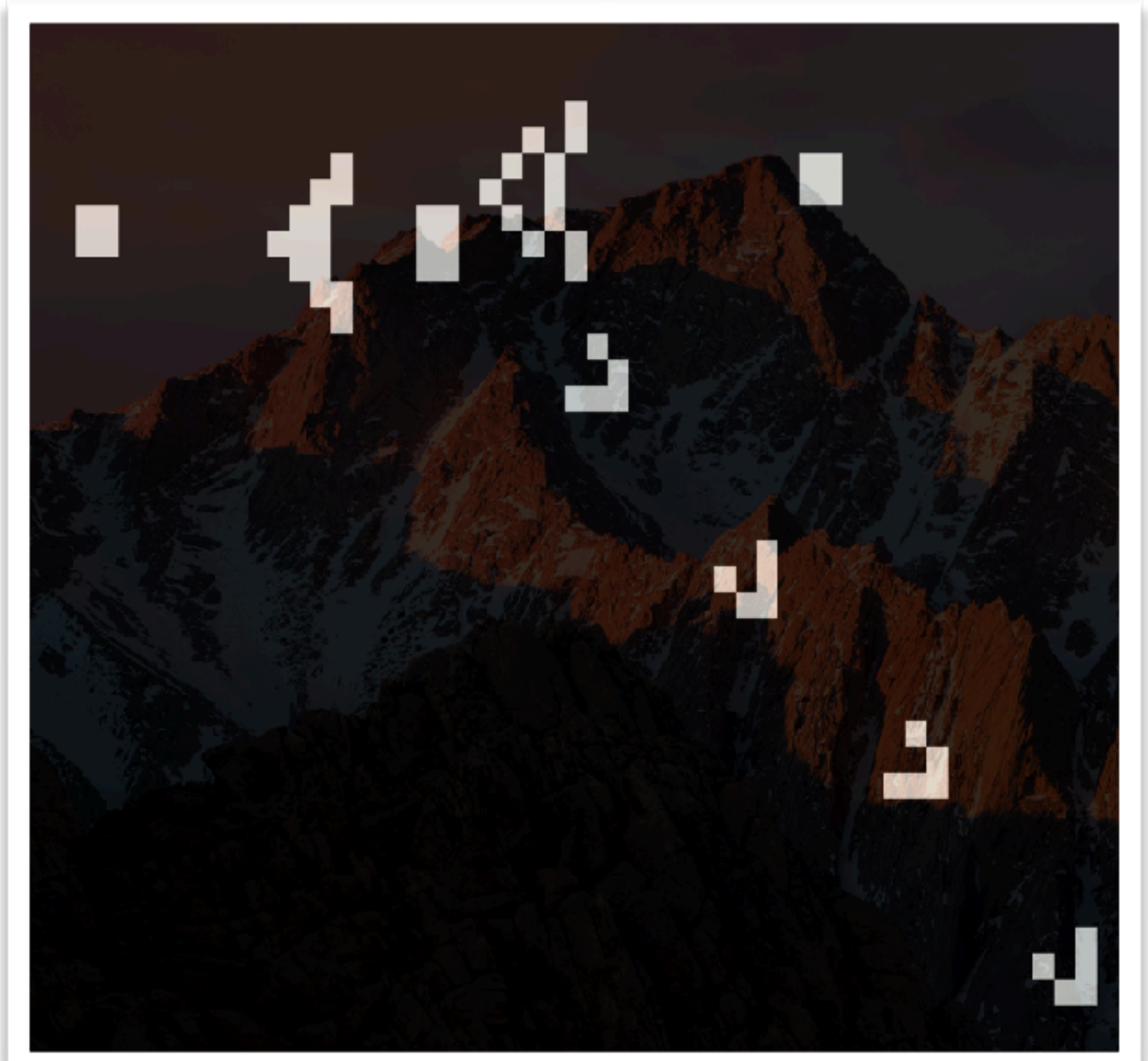
Console

```
Hoa\Console\Cursor::clear('↕');
Hoa\Console\Cursor::hide();
Hoa\Console\Cursor::move('↓', 1);

do {
    $hash = $this->computeHash();
    $this->displayWorld();
    $this->computeNewState();
} while ($hash !== $this->computeHash());

Hoa\Console\Cursor::show();
Hoa\Console\Cursor::colorize('default');
Hoa\Console\Cursor::move(
    '↓',
    $this->universe->getWidth()
);
```

Console



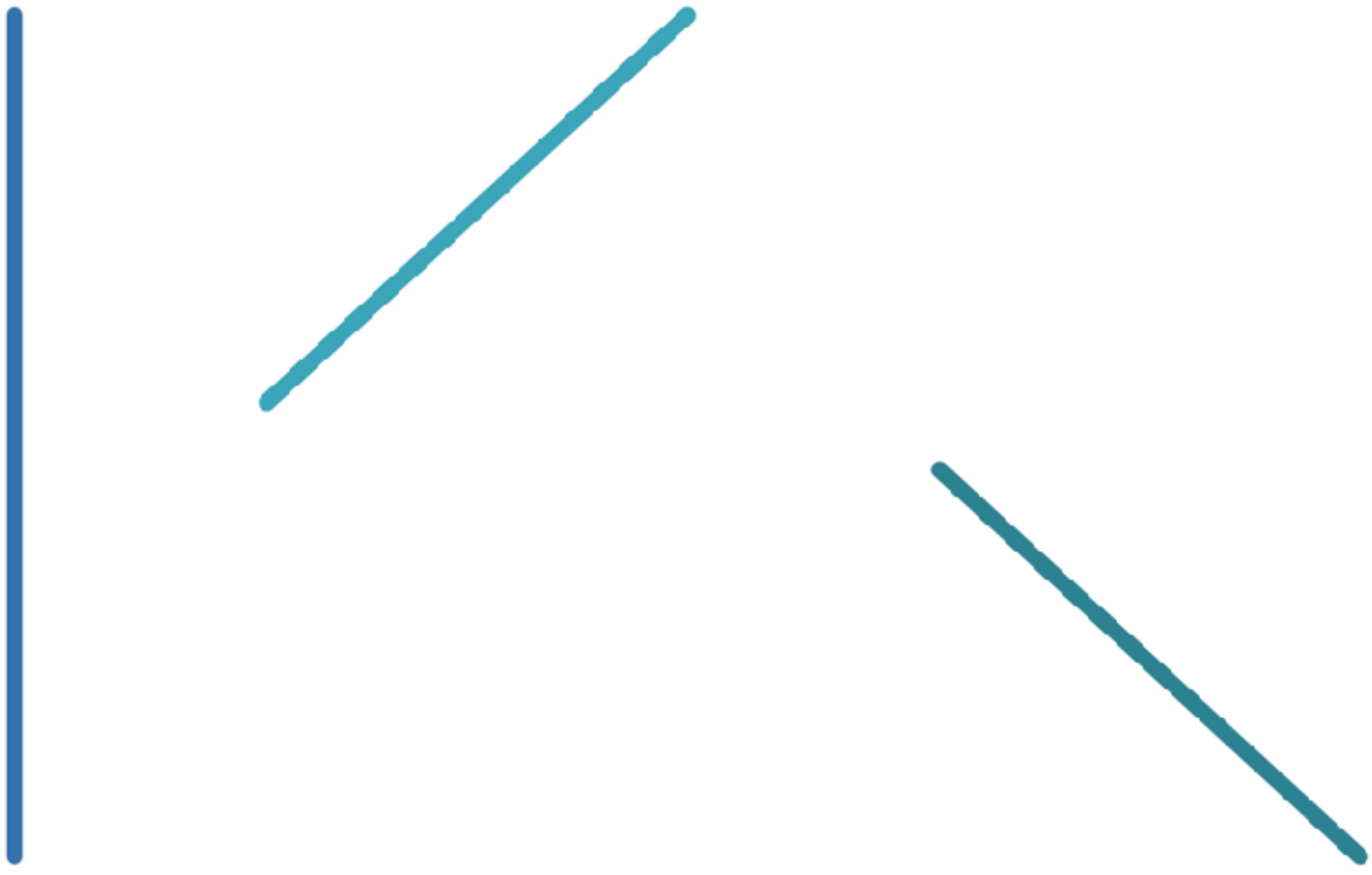
Bibliothèques

Acl	Dispatcher	Iterator	Notification	**Ruler**	Ustring
Bench	Dns	Json	**Option**	Serialize	View
Cache	Event	**Kitab**	**Praspel**	Session	Visitor
Cli	Eventsource	Locale	Promise	Socket	**Websocket**
Compiler	Exception	Log	Protocol	Stream	Worker
Consistency	Fastcgi	Mail	Prototype	String	Xml
Console	File	Math	Realdom	Stringbuffer	Xmlrpc
Core	Graph	Memory	Regex	Test	Xyl
Database	Http	Mime	Registry	Translate	Zformat
Devtools	Irc	Model	Router	Tree	Zombie

Bibliothèques

Acl	Dispatcher	Iterator	Notification	**Ruler**	Ustring
Bench	Dns	Json	**Option**	Serialize	View
Cache	Event	**Kitab**	**Praspel**	Session	Visitor
Cli	Eventsource	Locale	Promise	Socket	**Websocket**
Compiler	Exception	Log	Protocol	Stream	Worker
Consistency	Fastcgi	Mail	Prototype	String	Xml
Console	File	Math	Realdom	Stringbuffer	Xmlrpc
Core	Graph	Memory	Regex	Test	Xyl
Database	Http	Mime	Registry	Translate	Zformat
Devtools	Irc	Model	Router	Tree	Zombie

Kitab



Kitab

```
/**
 * This is a block of documentation, attached to a
PHP class.
 *
 * # Examples
 *
 * An example illustrates how to use the documented
entity, here the
 * class `C`.
 *
 * ```php
 * $c = new C();
 * ```
 */
class C { }
```

Kitab

```php
namespace Kitab;
/**
 * Configuration structure for the Kitab project.
 *
 * This structure contains all the configuration items used by Kitab.
 * All these items are public. Use it as a regular structure.
 *
 * # Examples
 *
 * Assuming the `foo` item is correctly declared:
 *
 * ```php
 * $configuration = new class () extends Kitab\Configuration {
 *     public $foo = null;
 * };
 * $configuration->foo = 'bar';
 *
 * assert('bar' === $configuration->foo);
 * ```
 */
abstract class Configuration
{
    /**
     * Get a configuration item value, or â€šÃÃ®if `null`â€šÃÃ® return a default value.
     * If the configuration item is not set, it will be considered as `null`.
     *
     * # Examples
     *
     * ```php
     * $configuration = new class () extends Kitab\Configuration {
     *     public $foo = null;
     * };
     *
     * assert($configuration->foo === null);
     * assert($configuration->getOr('foo', 'foobar') === 'foobar');
     * ```
     */
    public function getOr(string $item, $default)
    {
        if (null === $value = $this->$item) {
            return $default;
        }
        return $value;
    }
}
```

Kitab

```php
namespace Kitab;
/**
 * Configuration structure for the Kitab project.
 *
 * This structure contains all the configuration items used by Kitab.
 * All these items are public. Use it as a regular structure.
 *
 * # Examples
 *
 * Assuming the `foo` item is correctly declared:
 *
 * ```php
 * $configuration = new class () extends Kitab\Configuration {
 *     public $foo = null;
 * };
 * $configuration->foo = 'bar';
 *
 * assert('bar' === $configuration->foo);
 * ```
 */
abstract class Configuration
{
    /**
     * Get a configuration item value, or â€šÃÃ®if `null`â€šÃÃ® return a default value.
     * If the configuration item is not set, it will be considered as `null`.
     *
     * # Examples
     *
     * ```php
     * $configuration = new class () extends Kitab\Configuration {
     *     public $foo = null;
     * };
     *
     * assert($configuration->foo === null);
     * assert($configuration->getOr('foo', 'foobar') === 'foobar');
     * ```
     */
    public function getOr(string $item, $default)
    {
        if (null === $value = $this->$item) {
            return $default;
        }
        return $value;
    }
}
```

Class `Kitab\Configuration`

[src]

```
abstract class Configuration
{
    pub fn getOr(string $item, $default);
}
```


Configuration structure for the Kitab project.

This structure contains all the configuration items used by Kitab. All these items are public. Use it as a regular structure.

Examples

Assuming the `foo` item is correctly declared:

```
$configuration = new class () extends Kitab\Configuration {
    pub $foo = null;
};
$configuration->foo = 'bar';

assert('bar' === $configuration->foo);
```

Search anything...

Class `Kitab\Configuration`

[src]

```
abstract class Configuration
{
    pub fn getOr(string $item, $default);
}
```

Kitab

Classes

`Finder`
`Configuration`

Configuration structure for the Kitab project.

This structure contains all the configuration items used by Kitab. All these items are public. Use it as a regular structure.

Examples

Assuming the `foo` item is correctly declared:

```
$configuration = new class () extends Kitab\Configuration {
    pub $foo = null;
};
$configuration->foo = 'bar';

assert('bar' === $configuration->foo);
```

Bibliothèques

Acl	Dispatcher	Iterator	Notification	**Ruler**	Ustring
Bench	Dns	Json	**Option**	Serialize	View
Cache	Event	**Kitab**	**Praspel**	Session	Visitor
Cli	Eventsource	Locale	Promise	Socket	**Websocket**
Compiler	Exception	Log	Protocol	Stream	Worker
Consistency	Fastcgi	Mail	Prototype	String	Xml
Console	File	Math	Realdom	Stringbuffer	Xmlrpc
Core	Graph	Memory	Regex	Test	Xyl
Database	Http	Mime	Registry	Translate	Zformat
Devtools	Irc	Model	Router	Tree	Zombie

Bibliothèques

Acl	Dispatcher	Iterator	Notification	**Ruler**	Ustring
Bench	Dns	Json	**Option**	Serialize	View
Cache	Event	**Kitab**	**Praspel**	Session	Visitor
Cli	Eventsource	Locale	Promise	Socket	**Websocket**
Compiler	Exception	Log	Protocol	Stream	Worker
Consistency	Fastcgi	Mail	Prototype	String	Xml
Console	File	Math	Realdom	Stringbuffer	Xmlrpc
Core	Graph	Memory	Regex	Test	Xyl
Database	Http	Mime	Registry	Translate	Zformat
Devtools	Irc	Model	Router	Tree	Zombie

Option

This library is an implementation of the famous `Option` polymorphic type (also called `Maybe`)

Option

```
$x = Some(42);
$y = None();

assert($x->isSome());
assert($y->isNone());
```

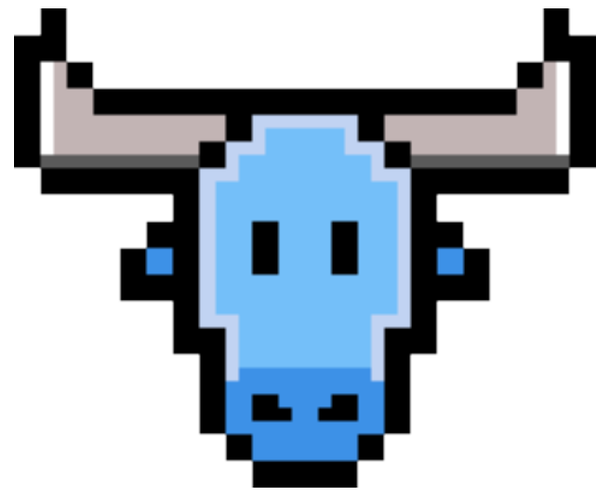
Bibliothèques

Acl	Dispatcher	Iterator	Notification	**Ruler**	Ustring
Bench	Dns	Json	**Option**	Serialize	View
Cache	Event	**Kitab**	**Praspel**	Session	Visitor
Cli	Eventsource	Locale	Promise	Socket	**Websocket**
Compiler	Exception	Log	Protocol	Stream	Worker
Consistency	Fastcgi	Mail	Prototype	String	Xml
Console	File	Math	Realdom	Stringbuffer	Xmlrpc
Core	Graph	Memory	Regex	Test	Xyl
Database	Http	Mime	Registry	Translate	Zformat
Devtools	Irc	Model	Router	Tree	Zombie

Bibliothèques

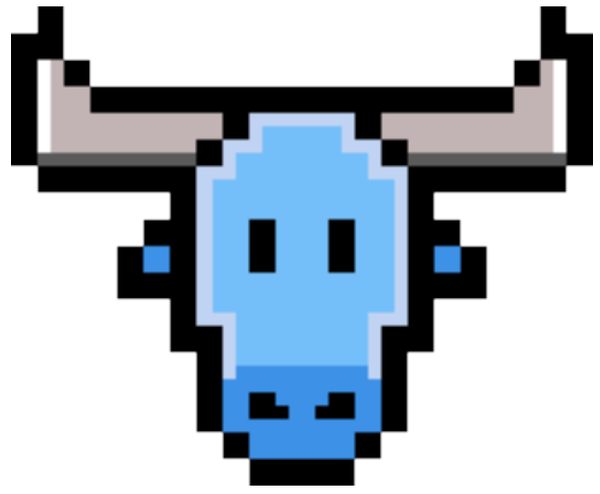
Acl	Dispatcher	Iterator	Notification	**Ruler**	Ustring
Bench	Dns	Json	**Option**	Serialize	View
Cache	Event	**Kitab**	**Praspel**	Session	Visitor
Cli	Eventsource	Locale	Promise	Socket	**Websocket**
Compiler	Exception	Log	Protocol	Stream	Worker
Consistency	Fastcgi	Mail	Prototype	String	Xml
Console	File	Math	Realdom	Stringbuffer	Xmlrpc
Core	Graph	Memory	Regex	Test	Xyl
Database	Http	Mime	Registry	Translate	Zformat
Devtools	Irc	Model	Router	Tree	Zombie

Praspel



Praspel

+



Praspel

```
$this->sample(
    $this->realdom->boundinteger(7, 13)
        ->or->boundinteger(42, 153)
)
```

```
$data = $this->realdom
    ->regex('/[\w\-_]+(\.[\w\-\_]+)*@\w\.(net|org)/');
$this->string($this->sample($data))
     ->contains(…)->…;
```

Praspel

```
$this->sample(
    $this->realdom->boundinteger(7, 13)
        ->or->boundinteger(42, 153)
)
```

```
$data = $this->realdom
    ->regex('/[\w\-_]+(\.[\w\-\_]+)*@\w\.(net|org)/');
$this->string($this->sample($data))
     ->contains(…)->…;
```

Praspel

```
$this->sample(
    $this->realdom->boundinteger(7, 13)
        ->or->boundinteger(42, 153)
)
```

```
$data = $this->realdom
    ->regex('/[\w\-_]+(\.[\w\-\_]+)*@\w\.(net|org)/');
$this->string($this->sample($data))
    ->contains(…)->…;
```

Praspel

```
%token  true            true
%token  false           false
%token  null            null
//…
%token  brace_          {
%token _brace           }
%token  bracket_        \[
%token  number          \-?(0|[1-9]\d*)(\.\d+)?([eE][\+\-]?\d+)?

value:
    <true> | <false> | <null> | string() | object() | array() | number()

number:
    <number>

#object:
    ::brace_:: pair() ( ::comma:: pair() )* ::_brace::

#pair:
    string() ::colon:: value()

#array:
    ::bracket_:: value() ( ::comma:: value() )* ::_bracket::

//…
```

Praspel

```
%token  true            true
%token  false           false
%token  null            null
//…
%token  brace_          {
%token _brace           }
%token  bracket_        \[
%token  number          \-?(0|[1-9]\d*)(\.\d+)?([eE][\+\-]?\d+)?
```

```
value:
    <true> | <false> | <null> | string() | object() | array() | number()

number:
    <number>

#object:
    ::brace_:: pair() ( ::comma:: pair() )* ::_brace::

#pair:
    string() ::colon:: value()

#array:
    ::bracket_:: value() ( ::comma:: value() )* ::_bracket::

//…
```

Praspel

```
%token  true            true
%token  false           false
%token  null            null
//…
%token  brace_          {
%token _brace           }
%token  bracket_        \[
%token  number          \-?(0|[1-9]\d*)(\.\d+)?([eE][\+\-]?\d+)?
```

```
value:
    <true> | <false> | <null> | string() | object() | array() | number()

number:
    <number>

#object:
    ::brace_:: pair() ( ::comma:: pair() )* ::_brace::

#pair:
    string() ::colon:: value()

#array:
    ::bracket_:: value() ( ::comma:: value() )* ::_bracket::

//…
```

Praspel

```
public function testJsonExhaustively()
{
    $compiler = \Hoa\Compiler\Llk\Llk::load(
       new \Hoa\File\Read(__DIR__ . DS .  'JsonGrammar.pp')
    );
    $sampler  = new \Hoa\Compiler\Llk\Sampler\BoundedExhaustive(
        $compiler,
        new \Hoa\Regex\Visitor\Isotropic(new \Hoa\Math\Sampler\Random()),
        5
    );

    foreach($sampler as $json) {

        json_decode($json);

        $this->integer(json_last_error())
             ->isEqualTo(JSON_ERROR_NONE);
    }
}
```

Praspel

```
public function testJsonExhaustively()
{
    $compiler = \Hoa\Compiler\Llk\Llk::load(
        new \Hoa\File\Read(__DIR__ . DS .  'JsonGrammar.pp')
    );
    $sampler  = new \Hoa\Compiler\Llk\Sampler\BoundedExhaustive(
        $compiler,
        new \Hoa\Regex\Visitor\Isotropic(new \Hoa\Math\Sampler\Random()),
        5
    );

    foreach($sampler as $json) {

        json_decode($json);

        $this->integer(json_last_error())
             ->isEqualTo(JSON_ERROR_NONE);
    }
}
```

Praspel

```
public function testJsonExhaustively()
{
    $compiler = \Hoa\Compiler\Llk\Llk::load(
        new \Hoa\File\Read(__DIR__ . DS .  'JsonGrammar.pp')
    );
    $sampler  = new \Hoa\Compiler\Llk\Sampler\BoundedExhaustive(
        $compiler,
        new \Hoa\Regex\Visitor\Isotropic(new \Hoa\Math\Sampler\Random()),
        5
    );

    foreach($sampler as $json) {

        json_decode($json);

        $this->integer(json_last_error())
             ->isEqualTo(JSON_ERROR_NONE);
    }
}
```

Praspel

```
public function testJsonExhaustively()
{
    $compiler = \Hoa\Compiler\Llk\Llk::load(
        new \Hoa\File\Read(__DIR__ . DS .  'JsonGrammar.pp')
    );
    $sampler  = new \Hoa\Compiler\Llk\Sampler\BoundedExhaustive(
        $compiler,
        new \Hoa\Regex\Visitor\Isotropic(new \Hoa\Math\Sampler\Random()),
        5
    );

    foreach($sampler as $json) {

        json_decode($json);

        $this->integer(json_last_error())
             ->isEqualTo(JSON_ERROR_NONE);
    }
}
```

Praspel

```
> atoum path: /private/tmp/praspel/vendor/atoum/atoum/vendor/bin/atoum
> atoum version: 2.6.1
> PHP path: /usr/local/Cellar/php70/7.0.12_5/bin/php
> PHP version:
=> PHP 7.0.12 (cli) (built: Oct 14 2016 09:56:59) ( NTS )
=> Copyright (c) 1997-2016 The PHP Group
=> Zend Engine v3.0.0, Copyright (c) 1998-2016 Zend Technologies
=>     with Zend OPcache v7.0.12, Copyright (c) 1999-2016, by Zend Technologies
=>     with Xdebug v2.4.1, Copyright (c) 2002-2016, by Derick Rethans
> Praspel\tests\units\Example...
[SSS_____________________________________________________________][3/3]
=> Test duration: 0.84 second.
=> Memory usage: 4.00 Mb.
> Total test duration: 0.84 second.
> Total test memory usage: 4.00 Mb.
> Running duration: 0.41 second.
Success (1 test, 3/3 methods, 0 void method, 0 skipped method, 2129 assertions)!
```

Praspel

```
> atoum path: /private/tmp/praspel/vendor/atoum/atoum/vendor/bin/atoum
> atoum version: 2.6.1
> PHP path: /usr/local/Cellar/php70/7.0.12_5/bin/php
> PHP version:
=> PHP 7.0.12 (cli) (built: Oct 14 2016 09:56:59) ( NTS )
=> Copyright (c) 1997-2016 The PHP Group
=> Zend Engine v3.0.0, Copyright (c) 1998-2016 Zend Technologies
=>     with Zend OPcache v7.0.12, Copyright (c) 1999-2016, by Zend Technologies
=>     with Xdebug v2.4.1, Copyright (c) 2002-2016, by Derick Rethans
> Praspel\tests\units\Example...
[SSS_____________________________________________________________][3/3]
=> Test duration: 0.84 second.
=> Memory usage: 4.00 Mb.
> Total test duration: 0.84 second.
> Total test memory usage: 4.00 Mb.
> Running duration: 0.41 second.
Success (1 test, 3/3 methods, 0 void method, 0 skipped method, 2129 assertions)!
```

Praspel

```
> atoum path: /private/tmp/praspel/vendor/atoum/atoum/vendor/bin/atoum
> atoum version: 2.6.1
> PHP path: /usr/local/Cellar/php70/7.0.12_5/bin/php
> PHP version:
=> PHP 7.0.12 (cli) (built: Oct 14 2016 09:56:59) ( NTS )
=> Copyright (c) 1997-2016 The PHP Group
=> Zend Engine v3.0.0, Copyright (c) 1998-2016 Zend Technologies
=>     with Zend OPcache v7.0.12, Copyright (c) 1999-2016, by Zend Technologies
=>     with Xdebug v2.4.1, Copyright (c) 2002-2016, by Derick Rethans
> Praspel\tests\units\Example...
[SSS_____________________________________________________________][3/3]
=> Test duration: 0.84 second.
=> Memory usage: 4.00 Mb.
> Total test duration: 0.84 second.
> Total test memory usage: 4.00 Mb.
> Running duration: 0.41 second.
Success (1 test, 3/3 methods, 0 void method, 0 skipped method, 2129 assertions)!
```

Bibliothèques

Acl	Dispatcher	Iterator	Notification	**Ruler**	Ustring
Bench	Dns	Json	**Option**	Serialize	View
Cache	Event	**Kitab**	**Praspel**	Session	Visitor
Cli	Eventsource	Locale	Promise	Socket	**Websocket**
Compiler	Exception	Log	Protocol	Stream	Worker
Consistency	Fastcgi	Mail	Prototype	String	Xml
Console	File	Math	Realdom	Stringbuffer	Xmlrpc
Core	Graph	Memory	Regex	Test	Xyl
Database	Http	Mime	Registry	Translate	Zformat
Devtools	Irc	Model	Router	Tree	Zombie

Bibliothèques

Acl	Dispatcher	Iterator	Notification	**Ruler**	Ustring
Bench	Dns	Json	**Option**	Serialize	View
Cache	Event	**Kitab**	**Praspel**	Session	Visitor
Cli	Eventsource	Locale	Promise	Socket	**Websocket**
Compiler	Exception	Log	Protocol	Stream	Worker
Consistency	Fastcgi	Mail	Prototype	String	Xml
Console	File	Math	Realdom	Stringbuffer	Xmlrpc
Core	Graph	Memory	Regex	Test	Xyl
Database	Http	Mime	Registry	Translate	Zformat
Devtools	Irc	Model	Router	Tree	Zombie

Ruler

```php
$ruler                = new Hoa\Ruler\Ruler();
$rule                 = 'logged(user) and points > 30';

$context              = new Hoa\Ruler\Context();
$context['user']      = new User();
$context['points']    = 42;

// Declare the `logged` function.
$asserter = new Hoa\Ruler\Visitor\Asserter();
$asserter->setOperator('logged', $logged);

$ruler->setAsserter($asserter);

// Assert!
var_dump(
    $ruler->assert($rule, $context)
);

/**
 * Will output:
 *     bool(true)
 */
```

Ruler

```php
$ruler                = new Hoa\Ruler\Ruler();
$rule                 = 'logged(user) and points > 30';

$context              = new Hoa\Ruler\Context();
$context['user']      = new User();
$context['points']    = 42;

// Declare the `logged` function.
$asserter = new Hoa\Ruler\Visitor\Asserter();
$asserter->setOperator('logged', $logged);

$ruler->setAsserter($asserter);

// Assert!
var_dump(
    $ruler->assert($rule, $context)
);

/**
 * Will output:
 *     bool(true)
 */
```

Ruler

```php
$ruler                = new Hoa\Ruler\Ruler();
$rule                 = 'logged(user) and points > 30';

$context              = new Hoa\Ruler\Context();
$context['user']      = new User();
$context['points']    = 42;

// Declare the `logged` function.
$asserter = new Hoa\Ruler\Visitor\Asserter();
$asserter->setOperator('logged', $logged);

$ruler->setAsserter($asserter);

// Assert!
var_dump(
    $ruler->assert($rule, $context)
);

/**
 * Will output:
 *     bool(true)
 */
```

Ruler

```php
$ruler                = new Hoa\Ruler\Ruler();
$rule                 = 'logged(user) and points > 30';

$context              = new Hoa\Ruler\Context();
$context['user']      = new User();
$context['points']    = 42;

// Declare the `logged` function.
$asserter = new Hoa\Ruler\Visitor\Asserter();
$asserter->setOperator('logged', $logged);

$ruler->setAsserter($asserter);

// Assert!
var_dump(
    $ruler->assert($rule, $context)
);

/**
 * Will output:
 *     bool(true)
 */
```

Ruler

```php
$ruler                = new Hoa\Ruler\Ruler();
$rule                 = 'logged(user) and points > 30';

$context              = new Hoa\Ruler\Context();
$context['user']      = new User();
$context['points']    = 42;

// Declare the `logged` function.
$asserter = new Hoa\Ruler\Visitor\Asserter();
$asserter->setOperator('logged', $logged);

$ruler->setAsserter($asserter);

// Assert!
var_dump(
    $ruler->assert($rule, $context)
);

/**
 * Will output:
 *     bool(true)
 */
```

RulerZ

```
$highRankFemalesRule = 'gender = "F" and points > 9000';
```

```
// or an array of objects
$playersObj = [
    new Player('Joe',   'M', 40, 2500),
    new Player('Moe',   'M', 55, 1230),
    new Player('Alice', 'F', 27, 9001),
];
```

```
$isHighRankFemale = $rulerz->satisfies($playersObj[0], $highRankFemalesRule);
```

RulerZ

```php
$highRankFemalesRule = 'gender = "F" and points > 9000';
```

```php
// or an array of objects
$playersObj = [
    new Player('Joe',   'M', 40, 2500),
    new Player('Moe',   'M', 55, 1230),
    new Player('Alice', 'F', 27, 9001),
];
```

```php
$isHighRankFemale = $rulerz->satisfies($playersObj[0], $highRankFemalesRule);
```

RulerZ

```
$highRankFemalesRule = 'gender = "F" and points > 9000';
```

```
// or an array of objects
$playersObj = [
    new Player('Joe',   'M', 40, 2500),
    new Player('Moe',   'M', 55, 1230),
    new Player('Alice', 'F', 27, 9001),
];
```

```
$isHighRankFemale = $rulerz->satisfies($playersObj[0], $highRankFemalesRule);
```

RulerZ

```
$highRankFemalesRule = 'gender = "F" and points > 9000';
```

```
// or an array of objects
$playersObj = [
    new Player('Joe',   'M', 40, 2500),
    new Player('Moe',   'M', 55, 1230),
    new Player('Alice', 'F', 27, 9001),
];
```

```
$isHighRankFemale = $rulerz->satisfies($playersObj[0], $highRankFemalesRule);
```

Bibliothèques

Acl	Dispatcher	Iterator	Notification	**Ruler**	Ustring
Bench	Dns	Json	**Option**	Serialize	View
Cache	Event	**Kitab**	**Praspel**	Session	Visitor
Cli	Eventsource	Locale	Promise	Socket	**Websocket**
Compiler	Exception	Log	Protocol	Stream	Worker
Consistency	Fastcgi	Mail	Prototype	String	Xml
Console	File	Math	Realdom	Stringbuffer	Xmlrpc
Core	Graph	Memory	Regex	Test	Xyl
Database	Http	Mime	Registry	Translate	Zformat
Devtools	Irc	Model	Router	Tree	Zombie

Bibliothèques

Acl	Dispatcher	Iterator	Notification	**Ruler**	Ustring
Bench	Dns	Json	**Option**	Serialize	View
Cache	Event	**Kitab**	**Praspel**	Session	Visitor
Cli	Eventsource	Locale	Promise	Socket	**Websocket**
Compiler	Exception	Log	Protocol	Stream	Worker
Consistency	Fastcgi	Mail	Prototype	String	Xml
Console	File	Math	Realdom	Stringbuffer	Xmlrpc
Core	Graph	Memory	Regex	Test	Xyl
Database	Http	Mime	Registry	Translate	Zformat
Devtools	Irc	Model	Router	Tree	Zombie

Websocket

```
$server = new Websocket\Server(
    new Socket\Server(
        'ws://127.0.0.1:8080')
    );

$server
    ->getConnection()
    ->setNodeName(Player::class);

$server->on(
    'open',
    function (Event\Bucket $bucket) {
        echo 'New player', "\n";
    }
);
```

Websocket

```
$server = new Websocket\Server(
    new Socket\Server(
        'ws://127.0.0.1:8080')
    );

$server
    ->getConnection()
    ->setNodeName(Player::class);

$server->on(
    'open',
    function (Event\Bucket $bucket) {
        echo 'New player', "\n";
    }
);
```

Websocket

```
$server = new Websocket\Server(
    new Socket\Server(
        'ws://127.0.0.1:8080')
    );

$server
    ->getConnection()
    ->setNodeName(Player::class);

$server->on(
    'open',
    function (Event\Bucket $bucket) {
        echo 'New player', "\n";
    }
);
```

Websocket

```
$server = new Websocket\Server(
    new Socket\Server(
        'ws://127.0.0.1:8080')
    );

$server
    ->getConnection()
    ->setNodeName(Player::class);

$server->on(
    'open',
    function (Event\Bucket $bucket) {
        echo 'New player', "\n";
    }
);
```

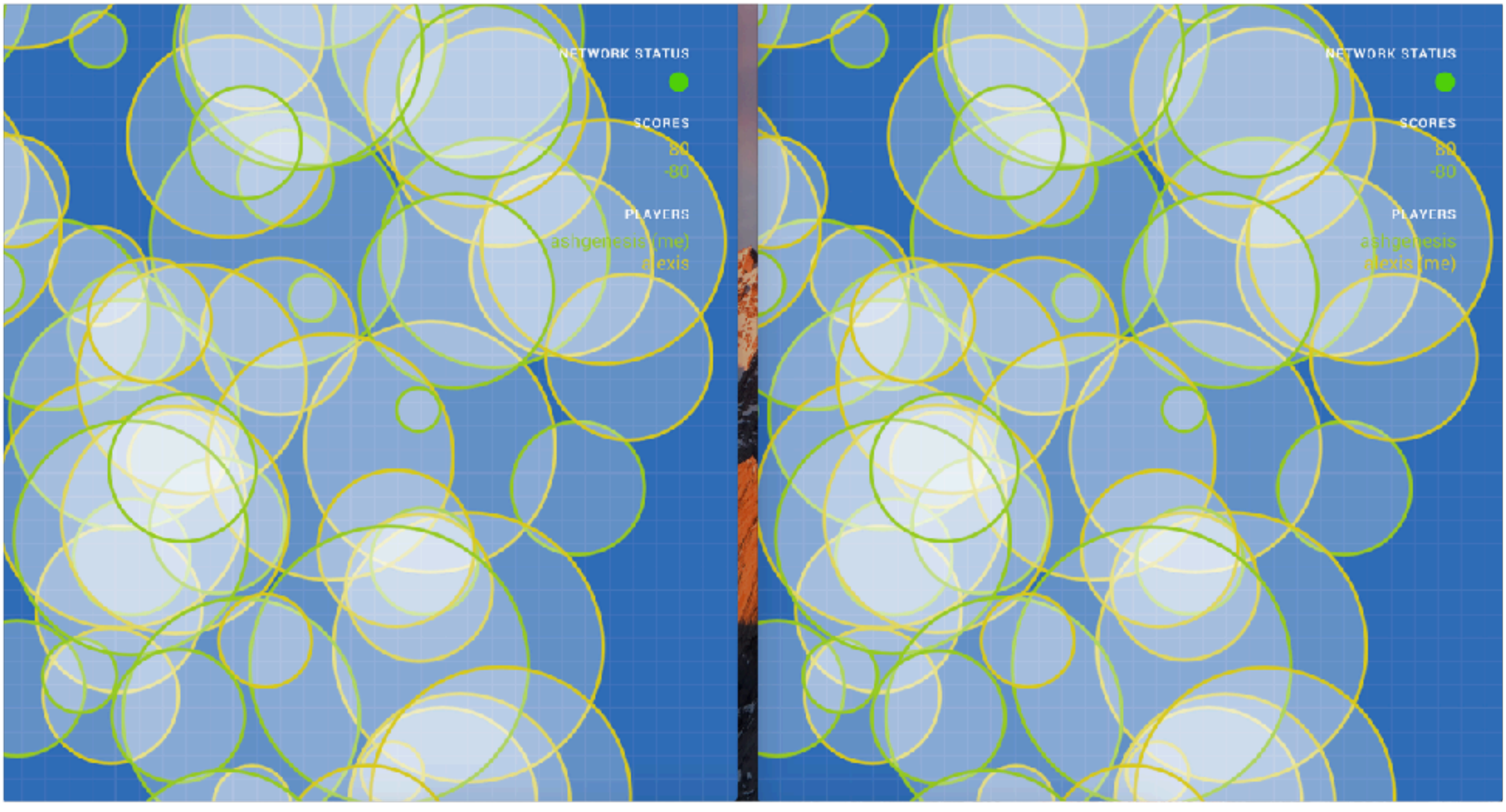
Websocket

```
$server->on(
    'message',
    function (Event\Bucket $bucket) use ($scores) {
        //…

        if (false === $message = @json_decode($data['message'])) {
            $bucket->getSource()->close();

            return;
        }

        switch ($message->type) {
            case 'server/player/new':
                //…
                break;

            case 'server/bubble/new':
                //…
                break;

            case 'server/bubble/delete':
                $bucket->getSource()->broadcast(
                    json_encode([
                        'type' => 'client/bubble/delete',
                        'id'   => $message->id
                    ])
                );
        }
    }
);
```

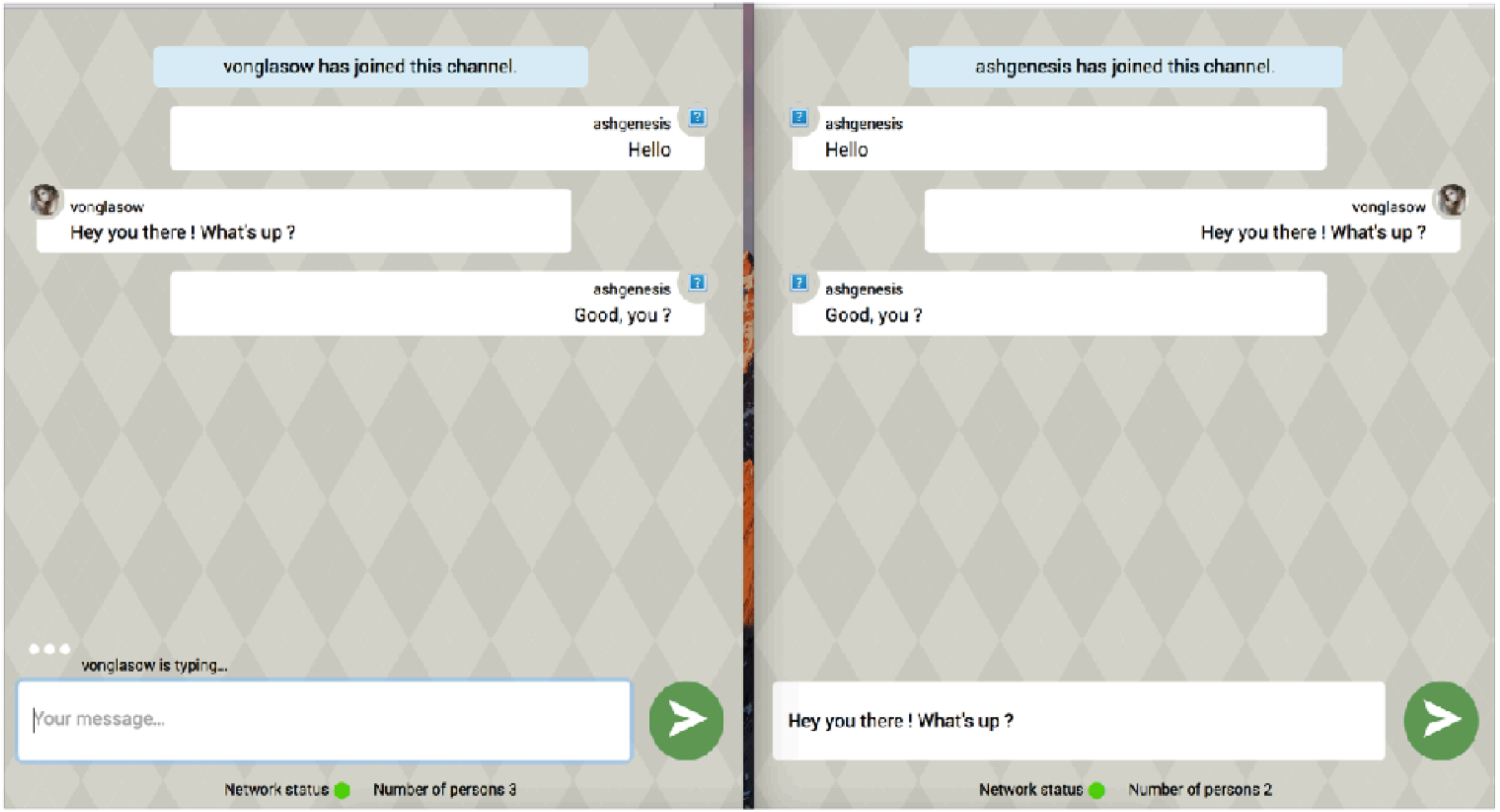
Websocket

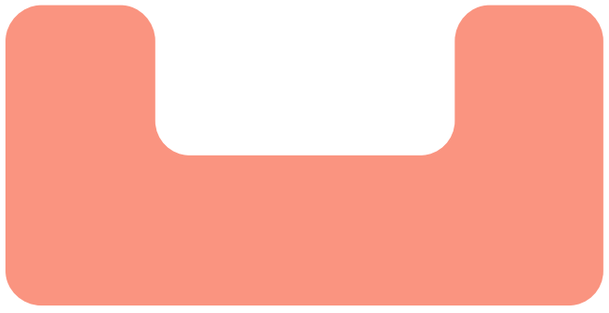
```php
$server->on(
    'message',
    function (Event\Bucket $bucket) use ($scores) {
        //…

        if (false === $message = @json_decode($data['message'])) {
            $bucket->getSource()->close();

            return;
        }

        switch ($message->type) {
            case 'server/player/new':
                //…
                break;

            case 'server/bubble/new':
                //…
                break;

            case 'server/bubble/delete':
                $bucket->getSource()->broadcast(
                    json_encode([
                        'type' => 'client/bubble/delete',
                        'id'   => $message->id
                    ])
                );
        }
    }
);
```

Bubble Game

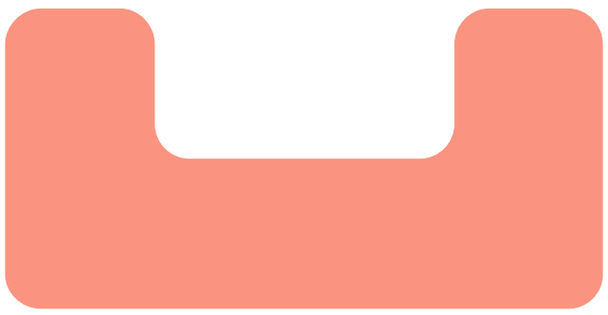

NETWORK STATUS
SCORES
80
-80
PLAYERS
ashgenesis (me)
alexis
NETWORK STATUS
SCORES
80
-80
PLAYERS
ashgenesis
alexis (me)

Chat

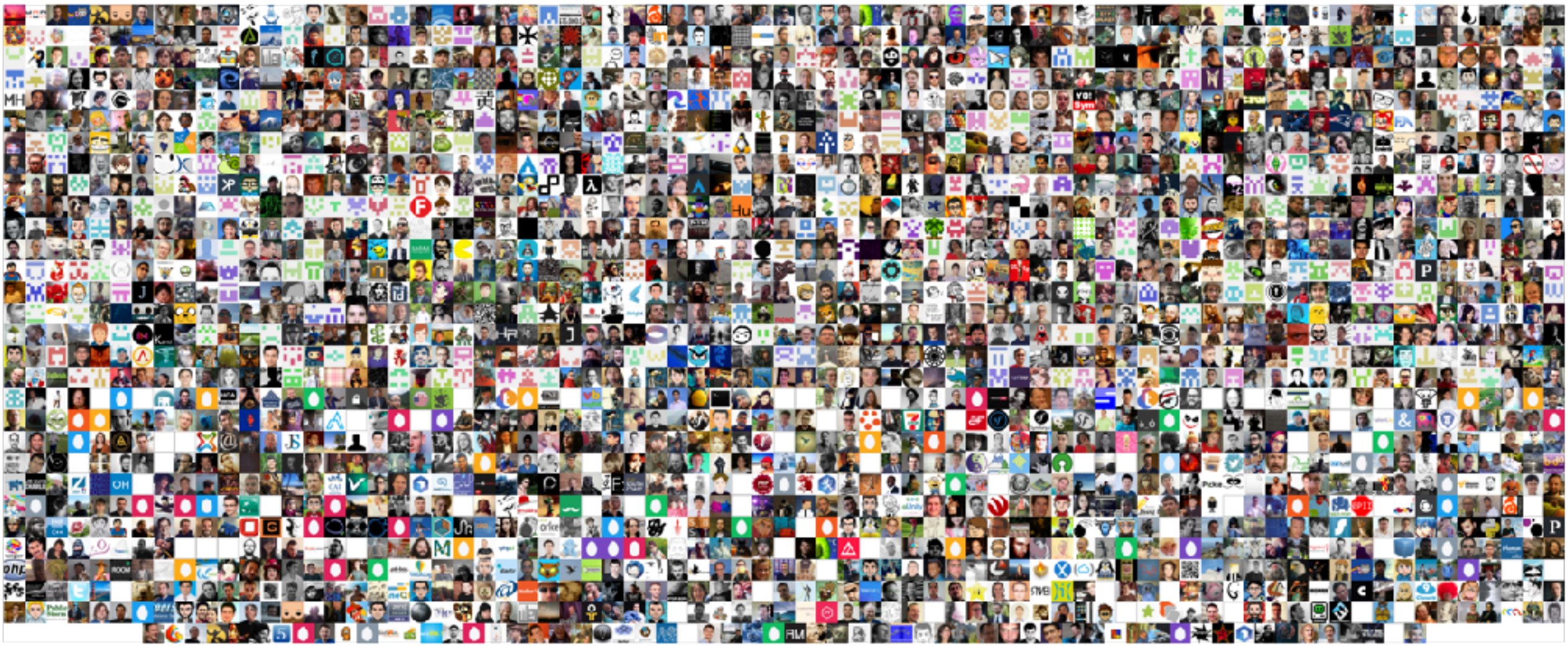

vonglasow has joined this channel.
ashgenesis
Hello
vonglasow
Hey you there ! What's up ?
ashgenesis
Good, you ?
vonglasow is typing...
Your message...
Network status
Number of persons 3
ashgenesis has joined this channel.
ashgenesis
Hello
vonglasow
Hey you there ! What's up ?
ashgenesis
Good, you ?
Hey you there ! What's up ?
Network status
Number of persons 2





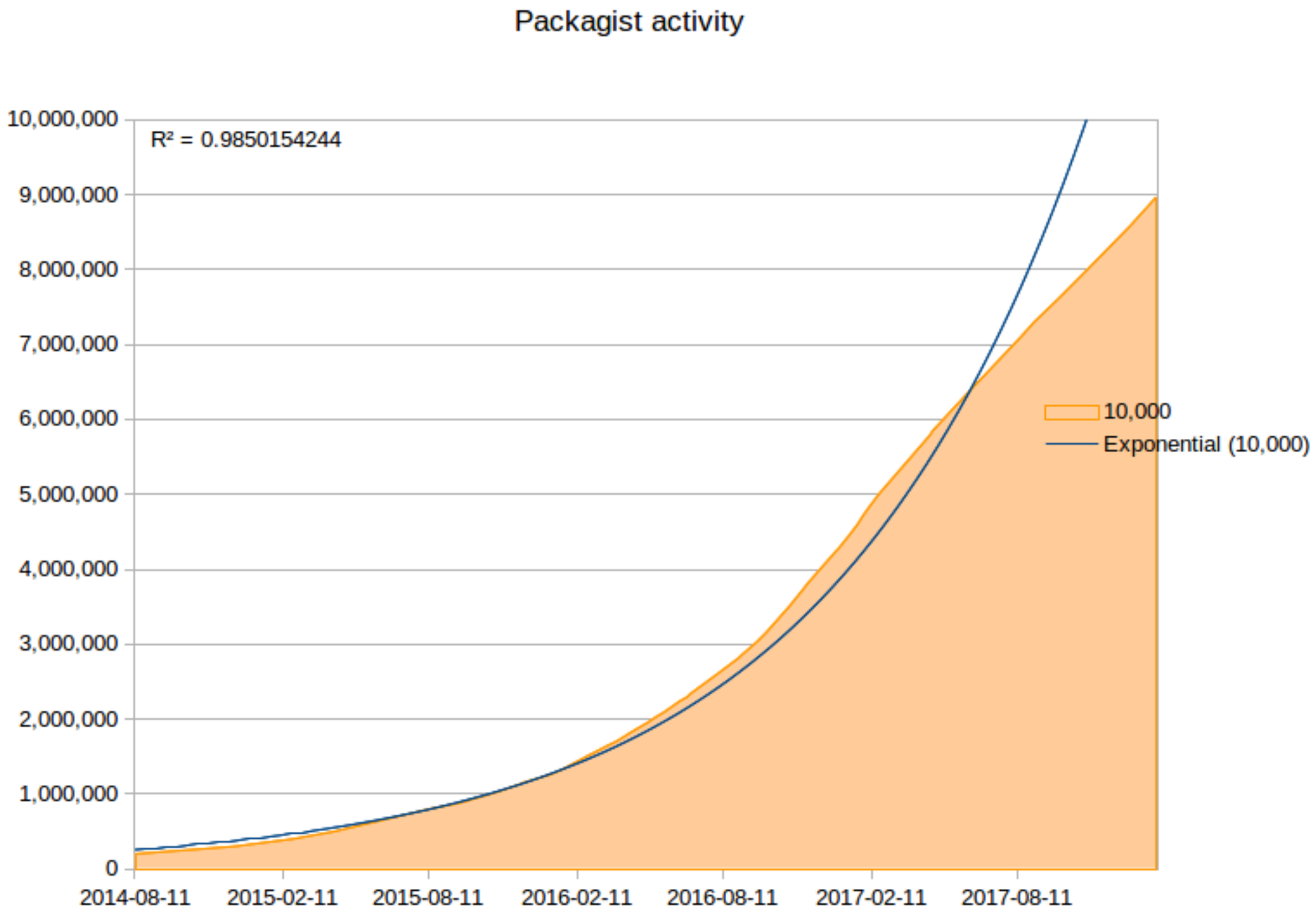



Communauté

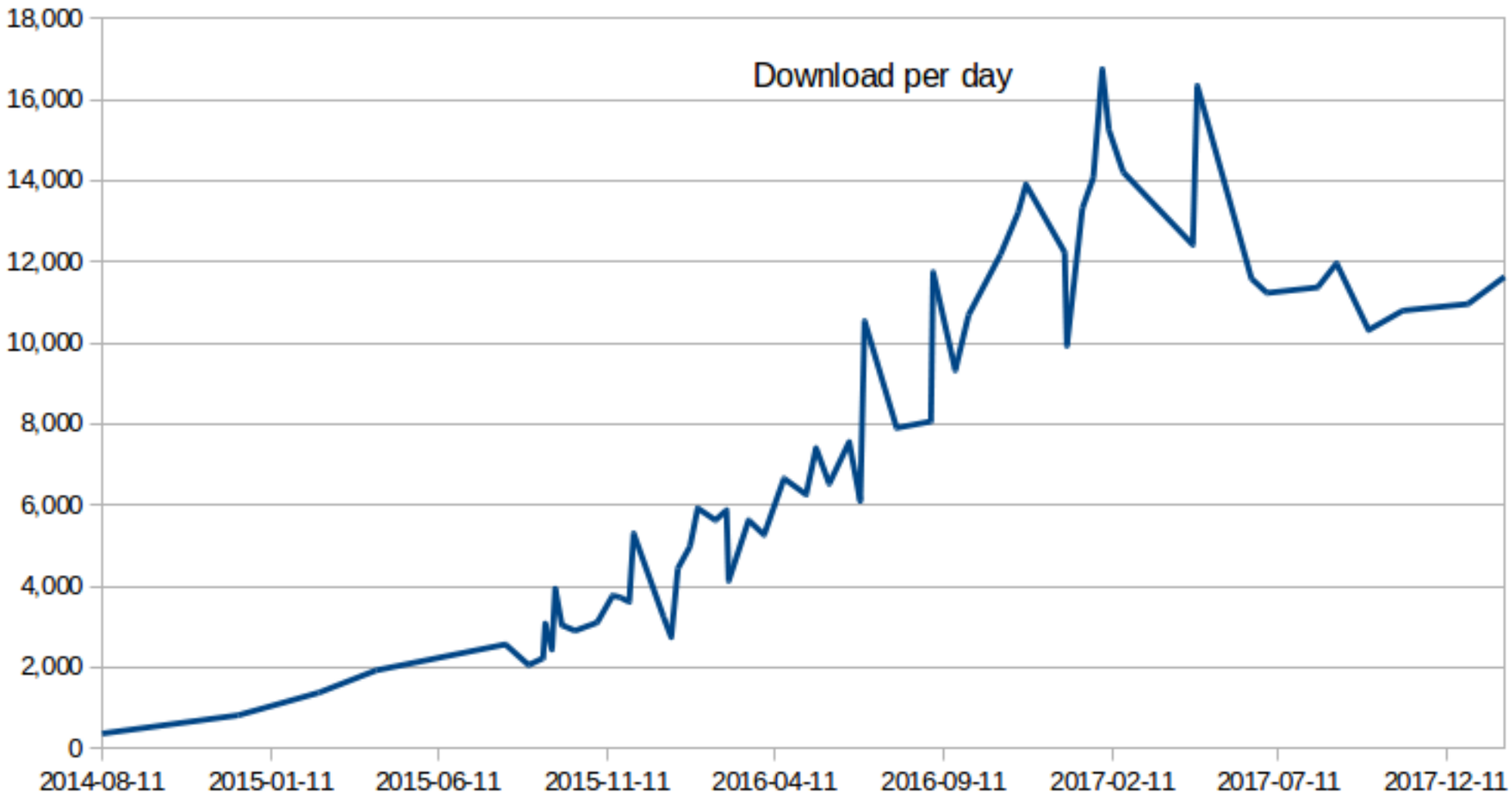


~2200 personnes

Packagist


Packagist activity
R² = 0.9850154244
10,000,000
9,000,000
8,000,000
7,000,000
6,000,000
5,000,000
4,000,000
3,000,000
2,000,000
1,000,000
0
2014-08-11
2015-02-11
2015-08-11
2016-02-11
2016-08-11
2017-02-11
2017-08-11
10,000
Exponential (10,000)

Téléchargement



Ils nous font confiance

Ils nous font confiance

Ils nous font confiance

Ils nous font confiance

fruux

ownCloud

atpik

Ils nous font confiance

fruux

owncloud

atpik

Wallonie SPW Service public de Wallonie

Ils nous font confiance

femto-st
SCIENCES & TECHNOLOGIES
DISC

fruux

atpik

Wallonie
SPW
Service public de Wallonie

ownCloud

Ils nous font confiance

femto-st
SCIENCES & TECHNOLOGIES
DISC

Inria

fruux

atpik

SPW
Service public de Wallonie
Wallonie

ownCloud

Ils nous font confiance

femto-st
SCIENCES & TECHNOLOGIES
DISC

Inria

fruux

ouest france

atpik

Wallonie
SPW
Service public de Wallonie

ownCloud

Ils nous font confiance

femto-st
SCIENCES & TECHNOLOGIES
DISC

Inria

fruux

ouest france

Wallonie
SPW
Service public de Wallonie

atpik

ownCloud

wallabag

Ils nous font confiance

femto-st SCIENCES & TECHNOLOGIES

DISC

Inria

PHPMETRICS

fruux

ouest france

Wallonie SPW Service public de Wallonie

ntopk

ownCloud

wallabag

Ils nous font confiance

femto-st
SCIENCES & TECHNOLOGIES
DISC

Inria

PHPMETRICS

fruux

ouest france

Wallonie
SPW
Service public de Wallonie

ownCloud

allabag

mozilla

Ils nous font confiance

femto-st
SCIENCES & TECHNOLOGIES
DISC

Inria

PHPMETRICS

fruux

ouest france

Wallonie
SPW
Service public de Wallonie

ownCloud

wallabag

mozilla

Merci

- https://hoa-project.net
- @hoaproject
- irc: freenode #hoaproject
- https://github.com/doctrine/annotations/pull/75
- http://psysh.org
- https://github.com/hoaproject/Contributions-Atoum-PraspelExtension
- https://github.com/vonglasow/game-of-life
- https://github.com/jubianchi/tml
- https://github.com/Hywan/ForumPHP2016
- https://github.com/K-Phoen/rulerz
- https://hoa-project.net/Fr/Awecode/Console-readline.html
- https://hoa-project.net/Fr/Awecode/Websocket.html
- https://github.com/hoaproject/Kitab