Programmation Web

— PHP et Programmation Objet—

IMT Lille Douai – UV CDAW

Luc Fabresse

luc.fabresse@imt-lille-douai.fr

Plan

Ce cours suppose que vous connaissez :

- Éléments de base PHP : boucles, fonctions, tableaux, ...
- HTTP (header, ...)
- PHP CGI (GET, POST, COOKIES, SESSION)
- traitement des formulaires Web (forms)

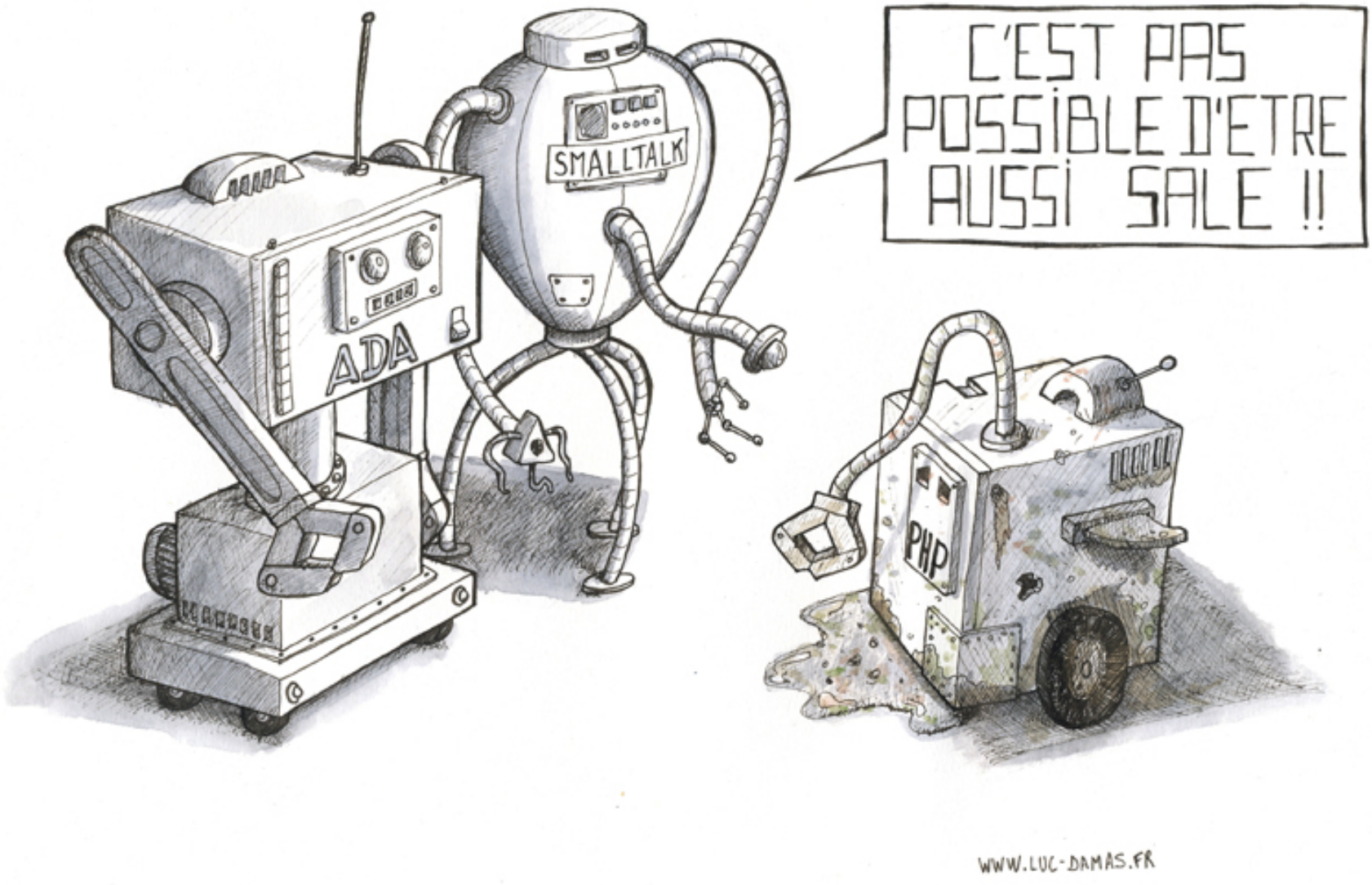
Auto-formation

cf. cours PHP de l'UV IDAW
cf. doc PHP `https://www.php.net/manual/en/`

Plan

https://www.php.net/manual/en/book.classobj


M'SIEUR Y'A QUOI DANS CE CARTON ?
DES SACS À VOMI...
AUJOURD'HUI, C'EST COURS DE PHP OBJET !
*
WWW.LUC-DAMAS.FR


C'EST PAS POSSIBLE D'ETRE AUSSI SALE !!
SMALLTALK
ADA
PHP
WWW.LUC-DAMAS.FR

Syntaxe

```php
class MaClasse
{
  private $private; // Attribut privé
  protected $protected; // Attribut protégé
  public $public; // Attribut public

  function __construct() { } // Contructeur
  function __destruct() { } // Destructeur
  private function maFonction() { } // Méthode privé
}
```

private, **protected** et **public** définissent la visibilité d'une méthode ou d'un attribut. (la valeur par défaut est **public**).

Membres statiques

En php il est possible de déclarer des membres ou des méthodes comme statiques. La déclaration **static** doit être faite après la déclaration de visibilitée.

Exemple

```php
class Foo
{
  public static $static = "foo";

  public static function staticFunction() {
    // bloc
  }
}
```

Membres statiques : accès (::)

Définition

L'opérateur de résolution de portée aussi appelé le symbole "double deux points" (::), fournit un moyen d'accéder aux membres **statiques** ou **constants** ainsi qu'aux **éléments redéfinis par la classe** *(source : fr.php.net)* .

```php
class MaClass {
  const CONSTANTE = "const_value";
  public static $bar = "bar";

  public static function showConstant() {
      echo self::CONSTANTE . "<br>";
  }
}

echo MaClass::$bar . "<br>"; // "bar"

echo MaClass::CONSTANTE . "<br>"; // "const_value"

MaClass::showConstant(); // "const_value"

$a = new MaClass();
$a->showConstant(); // "const_value"
```

Les mots-cles self et parent

Définitions

- **parent** désigne la super-classe
- **self** désigne la classe courante

Example de redéfinition

```php
abstract class ParentClass {
    public function doSomething() {
        echo "done.\n";
    }
}

class ChildClass extends ParentClass {
    public function doSomething() {
        echo "attempting something.\n";
        parent::doSomething();
    }
}

$obj = new ChildClass();
$obj->doSomething(); // "attempting something.\n" puis "done.\n"
```

Les mots-cles self et parent

Définition

- **parent** désigne la super-classe
- **self** désigne la classe courante

```php
class MaClass {
  const CONSTANTE = "const_value";
  public static $bar = "bar";

  public static function showConstant() {
      echo self::CONSTANTE . "<br>";
  }
}

class MaClass2 extends MaClass
{
  public static $staticiv = "statique";

  public static function test() {
    echo parent::$bar."\n"; //Equivaut à MaClass::$bar
    echo self::$staticiv."\n"; //Equivaut à MaClass2::$static
  }
}
```

Mot-clé static

http://php.net/manual/en/language.oop5.late-static-bindings.php

```php
class A {
    public static function message() {
       return "done";
    }
    public static function log() {
        echo self::message();
    }
}

class B extends A {
    public static function message() {
        return "currently doing";
    }
}

B::log();
```

Vive PHP !

Le mot-clé `static` permet d'avoir de la *liaison tardive* lors d'un envoi de message qui utilise un sélecteur de méthode de classe

Remarques

$this est une pseudo-variable (commence donc pas $) permettant de référencer l'objet courant dans le code d'une méthode

```php
class Foo
{
  public $var = "value";

  public __construct() {
    $this->var = "initialized";
  }
}
echo Foo::$var; //Error : Undeclared static property
$c=new Foo();
echo $c->var; //Affiche : initialized
```

La variable $this

```php
class A
{
  function toto()
  {
    if (isset($this)) {
      echo "$this existe";
    } else {
      echo "$this n'existe pas";
    }
    echo "\n";
  }
}
```

```php
$a = new A();
$a->toto();
A::toto();
```

Sortie

$this est définie (A)
$this n'est pas définie.

En d'autres termes ...

La variable **$this** n'existe pas dans les fonctions statiques.

Surcharge magique (abracadabra...)

http://php.net/manual/fr/language.oop5.overloading.php

Exemple

```php
class Test
{
    protected $cols;

    function __set($name, $value) {
        $this->cols[$name] = $value;
    }

    function __get($name) {
        return $this->cols[$name];
    }
}

$obj = new Test;

$obj->a = 1;
echo $obj->a . "\n";
```


PHP
iss