

POO

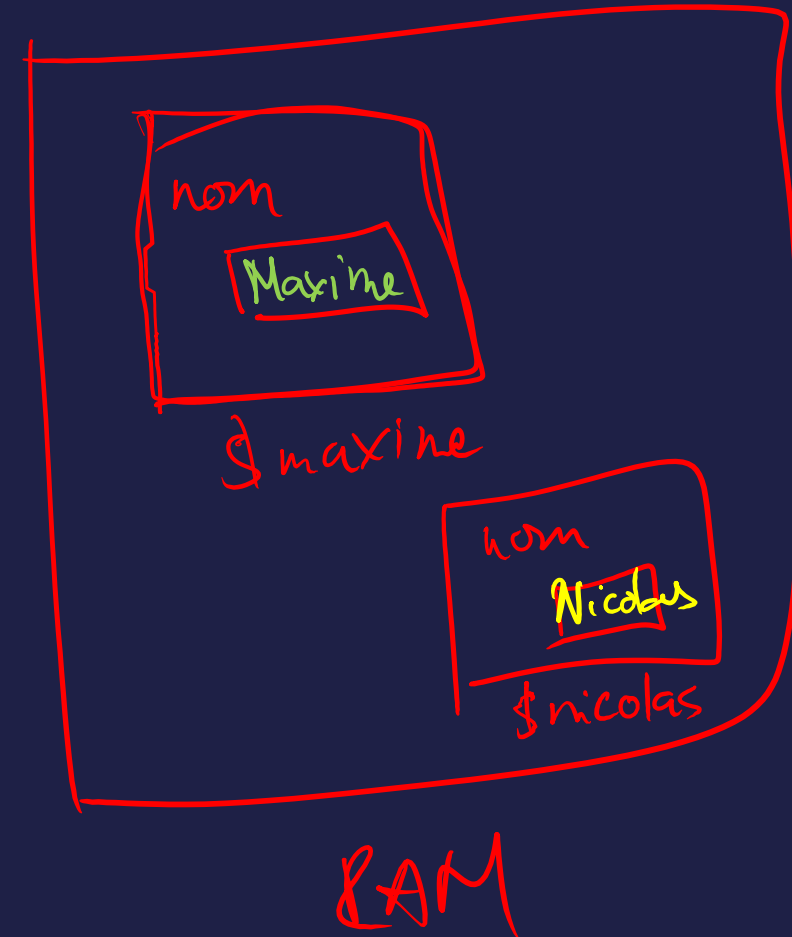
Michael
X  NATIS

1. L'opérateur `$this`
2. Les `constructeurs`
3. Les `exceptions`
4. Les éléments `static`
5. Les `namespaces`

L'opérateur `$this`

```
$maxime = new User();  
$nicolas = new User();
```

```
<?php  
  
class User {  
    private $nom;  
  
    public function getNom(): string {  
        return $this->nom;  
    }  
  
    public function setNom(string $value) {  
        $this->nom = $value;  
    }  
}  
  
$maxime = new User();  
$maxime->setNom('Maxime');  
echo ($maxime->getNom());  
// Output : Maxime  
  
?>
```



L'opérateur `$this` permet
de désigner l'objet courant

Les **setters** et les **getters**
permettent un **accès**
contrôlé aux propriétés

Les constructeurs

```
<?php  
  
class User {  
    fun toto() {  
        function __construct(string $nom, string $email) {  
        }  
    }  
}  
  
$maxime = new User('Maxime', 'Lemaire');  
$maxime -> toto();  
?>
```

\$bertrand = new User('Bertrand', 'toto');

Les **constructeurs**
permettent de **paramétrer**
la construction d'un objet.
C'est la 1^{ère} méthode qui est
appelée

Les exceptions

Lancement d'une erreur (exception)

```
<?php  
  
throw new Exception("L'utilisateur n'a pas d'adresse email");  
  
?>
```

Réaction à une exception

```
<?php

try {

    // Code avec une exception potentielle

} catch (Exception $e) {

    echo 'Une erreur est survenue: ' . $e->getMessage();

}

?>
```

Les exceptions permettent
une gestion d'erreur facile
mais sont coûteuses
techniquement

Les éléments **static**

```
<?php

class User {

    public static $prenom = 'Sandra';

    public static function sePresenter() {
        echo self::$prenom ;
    }

    public function changerDePrenom() {
        self::$prenom = 'Jean';
    }
}

$jean = new User();
$jean->changerDePrenom();
$jean->sePresenter();
// Output : Jean

$sandra = new User();
$sandra->sePresenter();
// Output : Jean
?>
```

Les éléments `static` (propre à la classe et non à l'objet)

Les éléments **static** (propre à la classe et non à l'objet)



Les namespaces

```
<?php

namespace App\Models {

    class User {
        public $prenom;
    }
}

?>
```

```
<?php

include 'User.php';

use App\Models\User;

$thomas = new User();

?>
```


Les namespaces
permettent d'organiser les
classes en groupe

