

# Object Calisthenics





<https://www.amazon.com/ThoughtWorks-Anthology-Technology-Innovation-Programmers/dp/193435614X>

Utiliser un seul niveau d'indentation par méthode .....	3
N'utilisez pas le mot-clé else .....	4
Enveloppez toutes les primitives et les chaînes.....	5
Utilisez des collections de première classe.....	5
Utilisez un seul point par ligne.....	6
Ne pas abréger .....	7
Gardez toutes les entités petites .....	8
N'utilisez aucune classe avec plus de deux variables d'instance .....	9
N'utilisez pas de getters/setters/propriétés .....	10

# Utiliser un seul niveau d'indentation par méthode

```
function register()
{
    if (!empty($_POST)) {
        $msg = '';
        if ($_POST['user_name']) {
            if ($_POST['user_password_new']) {
                if ($_POST['user_password_new'] == $_POST['user_password_repeat']) {
                    if (strlen($_POST['user_password_new']) > 5) {
                        if (strlen($_POST['user_name']) < 65 && strlen($_POST['user_name']) > 1) {
                            if (preg_match('/^[a-z\d]{2,64}$/i', $_POST['user_name'])) {
                                $user = read_user($_POST['user_name']);
                                if (!isset($user['user_name'])) {
                                    if ($_POST['user_email']) {
                                        if (strlen($_POST['user_email']) < 65) {
                                            if (filter_var($_POST['user_email'], FILTER_VALIDATE_EMAIL)) {
                                                create_user();
                                                $_SESSION['msg'] = 'You are now registered so please login';
                                                header('Location: ' . $_SERVER['PHP_SELF']);
                                                exit();
                                            } else $msg = 'You must provide a valid email address';
                                        } else $msg = 'Email must be less than 64 characters';
                                    } else $msg = 'Email cannot be empty';
                                } else $msg = 'Username already exists';
                            } else $msg = 'Username must be only a-z, A-Z, 0-9';
                        } else $msg = 'Username must be between 2 and 64 characters';
                    } else $msg = 'Password must be at least 6 characters';
                } else $msg = 'Passwords do not match';
            } else $msg = 'Empty Password';
        } else $msg = 'Empty Username';
        $_SESSION['msg'] = $msg;
    }
    return register_form();
}
```



N'utilisez pas le mot-clé else



Enveloppez toutes les primitives et les chaînes

```
public class Person
{
    private ZipCode zipCode;

}

...

public class ZipCode {

    private String code;

    public ZipCode (String code)
    {
        validate(code);
        this.code = code;
    }

    private void validate(code) {...}

}
```

Utilisez des collections de première classe

# Utilisez un seul point par ligne

Loi de Déméter

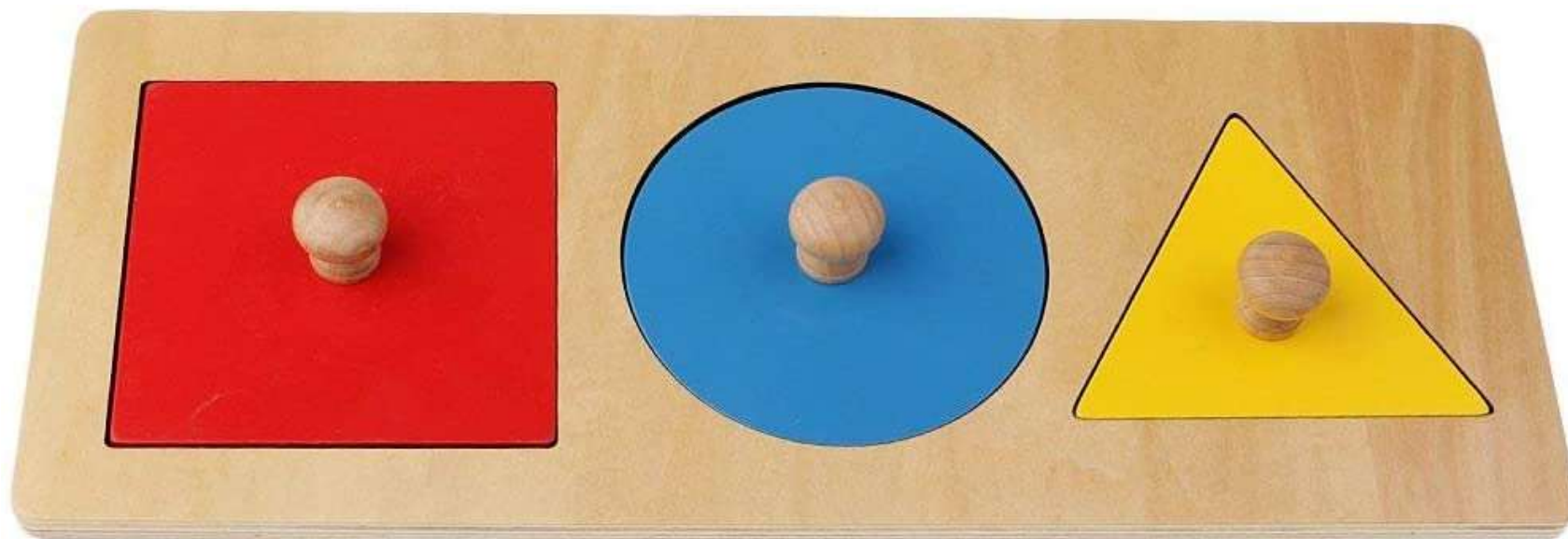
« Ne parlez qu'à vos amis immédiats »

```
$this->gitBackupDir = PluginManager::instance()  
->getPluginByName('git')  
->getPluginInfo()  
->getPropVal('git_backup_dir');
```

Ne pas abrégé

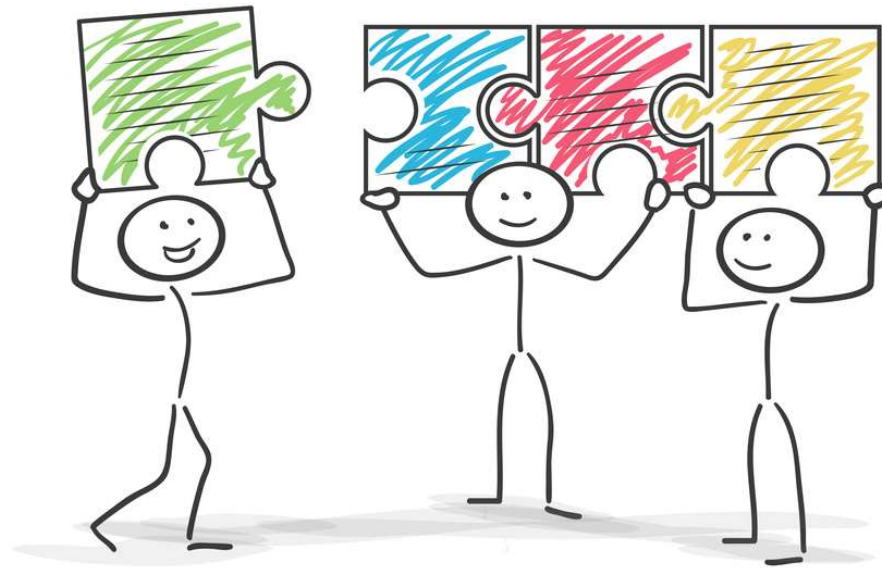


Gardez toutes les entités petites





N'utilisez aucune classe avec plus de deux variables d'instance



# N'utilisez pas de getters/setters/propriétés

« Tell don't ask »

```
public class Player
{
    private int score;

    public Player(int score)
    {
        this.score = score;
    }

    public void setScore(int score)
    {
        this.score = score;
    }
}
```



```
public class Player
{
    private int score;

    public Player(int score)
    {
        this.score = score;
    }

    public void increaseScoreBy(int points)
    {
        score += points;
    }

    public void decreaseScoreBy(int points)
    {
        score -= points;
    }
}
```