Don’t do it in production!
When the BUG comes up
DON’T PANIC
Be yourself
Debugging mindset

- Fix the Problem, Not the Blame.
- Don't Assume It – Prove It.
- The best way to start fixing a bug is to make it reproducible.
- Now what happens before and after.
- Don't Live with Broken Windows. (The Boy Scout Rule)
- Explain the bug to someone else. (Rubber duck)
- Stay aware of what you're doing.
- Understand Stack Traces
- Test Your Software, or Your Users Will.
- A bad Unit Test is worse than No Test. Robert C. Martin (Uncle Bob).
- Ask yourself questions.
Debugging checklist

- Is the problem being reported a direct result of the underlying bug, or merely a symptom?

- Is the bug really in the code? Is it in the OS? Or is it in third party packages?

- If you explained this problem in detail to a coworker, what would you say?

- If the suspect code passes its tests, are the tests complete enough? What happens if you run the test with this data?

- Do the conditions that caused this bug exist anywhere else in the system?
Debugging checklist

- Is there an easier way?
- Am I solving the right problem?
- Why is this a problem?
- What makes it hard?
- Do I have to do it this way?
- Does it have to be done at all?

$CFG->debug = (E_ALL | E_STRICT);
Moodle Debug Developer

Dashboard / Site administration / Development / Debugging

Debugging

- **Debug messages**
  - Developer: extra Moodle debug messages for developers
  - Default: NONE: Do not show any errors or warnings
  - If you turn this on, then PHP’s error_reporting will be increased so that more warnings are printed. This is only useful for developers.

- **Display debug messages**
  - Default: Yes
  - Set to on, the error reporting will go to the HTML page. This is practical, but breaks XHTML, JS, cookies and HTTP headers in general. Set to off, it will send the output to your server logs, allowing better debugging. The PHP setting error_log controls which log this goes to.

- **Debug email sending**
  - Default: No
  - Enable verbose debug information during sending of email messages to SMTP server.

- **Performance info**
  - Default: No
  - If you turn this on, performance info will be printed in the footer of the standard theme.

- **Show origin of languages**
  - Default: No
  - If enabled, language string components and identifiers are displayed when ?strings=1 or &strings=1 is appended to the page URL.

- **Show validator links**
  - Default: No
  - Enable if you want to have links to external validator servers in page footer. You may need to create new user with username $validator, and enable guest access. These changes may allow unauthorized access to server, do not enable on production sites!

- **Show page Information**
  - Default: No
  - Enable if you want page information printed in page footer.

[Save changes]
Moodle Debug Developer

config.php

$CFG->debug = (E_ALL | E_STRICT); // === DEBUG_DEVELOPER
$CFG->debugdisplay = 1; // unless watch the logs.
$CFG->debugusers = '2,8'; // List of user ids that that always see debug messages.
$CFG->cachejs = false; // Prevent JS caching
$CFG->debugstringids = 1; // Show origin of languages strings
$CFG->langstringcache = false; // Disable language strings cache
$CFG->themedesignermode = true; // Disable CSS and image cache
$CFG->divertallemailsto = 'root@localhost.local'; // Divert all outgoing emails to this address
$CFG->noemailever = true; // no emails or other messages should ever be send to real users
$CFG->divertallemailsexcept = 'tester@dev.com, fred(\+.*)?@example.com'; // Send emails only for certain email addresses you want to let through for testing.
Moodle Debug Developer

config.php

$CFG->perfdebug = 15;
$CFG->debugpageinfo = true;
Debugging

**PHP**

```php
if ($CFG->debugdeveloper) {
    ...
    debugging('Argument $record must be an instance of stdClass.', DEBUG_DEVELOPER);
    ...
}
```

**JS**

```javascript
define(['core/log'], function(log) {
    log.info(....);
});
```

*Web inspector*
Code Linting

https://docs.moodle.org/dev/Linting

PHP_CodeSniffer & Local_codechecker (by Tim Hunt)
JavaScript (ESLint) https://eslint.org/
CSS/SCSS/LESS (stylelint) https://stylelint.io/
Lint in Editors (Sublime, Atom, Vim, ...)

PHPXREF
http://phpxref.sourceforge.net/
Profiling

Dashboard / Site administration / Development / Profiling
- https://docs.moodle.org/dev/Profiling_PHP#Tideways_for_php7
- https://tracker.moodle.org/browse/MDL-62280 (Tideways XHProf)
Run tasks

```
php admin/tool/task/cli/schedule_task.php
   --execute="\tool_cohortroles\task\cohort_role_sync"
   --showdebugging
```

In Moodle 3.6:
```
php admin/tool/task/cli/adhoc_task.php --execute
```

```
$ php admin/tool/task/cli/schedule_task.php --execute="\tool_cohortroles\task\cohort_role_sync" --showdebugging
Execute scheduled task: Sync cohort role assignments (tool_cohortroles\task\cohort_role_sync)
Sync cohort roles...
Added 0
Removed 0
... used 2 dbqueries
... used 0.45213103294373 seconds
Scheduled task complete: Sync cohort role assignments (tool_cohortroles\task\cohort_role_sync)

$ php adhoc_task.php --execute --showdebugging

Execute adhoc task: core\task\build_installed_themes_task
... started 22:24:39. Current memory use 11.1MB.
```
Web Service

**CURL**

curl 'http://localhost/m/stable_master/webservice/rest/server.php?moodlewsrestformat=json' --data 'wsfunction=tool_mobile_get_config&wstoken=08788646962394c92aab6c26a3be7abe&section=mobileapp' | python -m "json.tool"
Web Service

**BROWSER**

http://localhost/m/stable_master/webservice/rest/server.php?wsfunction=tool_mobile_get_config&wstoken=08788646962394c92aab6c26a3be7abe

http://localhost/m/stable_master/webservice/rest/simpleserver.php?wsusername=wstest&wspassword=wstest&wsfunction=core_user_create_users&users[0][username]=mootesdev&users[0][firstname]=Test&users[0][lastname]=MootWS&users[0][email]=mootesdev@moodle.com&users[0][password]=12345&users[0][customfields][0][type]=cp&users[0][customfields][0][value]=08208
Web Service Resources

WS API
https://docs.moodle.org/dev/Web_services_API

Dev Docs
https://docs.moodle.org/dev/Creating_a_web_service_client

MoodleHQ WS Clients:
https://github.com/moodlehq/sample-ws-clients

Juan Leyva Master of WS
http://es.slideshare.net/juanleyva/mastering-moodle-web-services-development
Moodle Mobile

Debugging WS Request
https://docs.moodle.org/dev/Moodle_Mobile_debugging_WS_requests

Use an online web version of the app that can be easily debugged using the Chrome browser.
https://prototype.moodle.net/mobile/webapp/

Development using Chrome or Chromium
https://docs.moodle.org/dev/Moodle_Mobile_development_using_Chrome_or_Chromium

`ionic serve --browser chromium`
Docker to the rescue

**LDAP**

docker run -p 389:389 --name my-openldap-container --hostname ldap.example.org --env LDAP_REPLICATION=true --detach osixia/openldap:1.2.0

docker run -p 6443:443 --name phpldapadmin-service --link my-openldap-container:ldap-host --env PHPLDAPADMIN_LDP_HOSTS=ldap-host --detach osixia/phpldapadmin:0.7.1

You can go to [https://localhost:6443/](https://localhost:6443/) in your browser
login: cn=admin,dc=example,dc=org
password: admin

[https://tracker.moodle.org/browse/MDL-61296](https://tracker.moodle.org/browse/MDL-61296)
Moodle Docker

https://github.com/moodlehq/moodle-docker

**moodle-docker:** Docker Containers for Moodle Developers

- This repository contains Docker configuration aimed at Moodle developers and testers to easily deploy a testing environment for Moodle.

**Features:**
- All supported database servers (PostgreSQL, MySQL, Microsoft SQL Server, Oracle XE)
- Behat/Selenium configuration for Firefox and Chrome
- Catch-all smtp server and web interface to messages using MailHog
- All PHP Extensions enabled configured for external services (e.g. solr, ldap)
- All supported PHP versions
- Zero-configuration approach
- Backed by automated tests

**Oracle Example**

export MOODLE_DOCKER_WWWROOT=/Home/dev/htdocs/moodleoracle
export MOODLE_DOCKER_DB=oracle

cp config.docker-template.php $MOODLE_DOCKER_WWWROOT/config.php

bin/moodle-docker-compose up -d

bin/moodle-docker-wait-for-db
Expose local servers to internet

**NGROK** ([https://ngrok.com/](https://ngrok.com/))

**SERVEO** ([https://serveo.net/](https://serveo.net/))
MDK - Moodle Development Kit

- [https://github.com/FMCorz/mdk](https://github.com/FMCorz/mdk) developed by Frédéric Massar
- [https://docs.moodle.org/dev/Moodle_Development_kit](https://docs.moodle.org/dev/Moodle_Development_kit)

#update mdk
sudo pip install --upgrade moodle-sdk

#your mdk config file.
./moodle-sdk/config.json

#place where save script that you want to run like mdk run SCRIPT
./moodle-sdk/scripts

#upgrade your Moodle instances
mdk upgrade --all --update

#config your master branch
mdk config set masterBranch 36

#install a new instance of Moodle
mdk create -v master -n stable_master -i --run users mindev

#install a new integration instance of Moodle
mdk create -v 34 -n int34 --integration -i --run users mindev
MDK - Moodle Development Kit

#create a new branch MDL-56141-master-test
mdk pull MDL-56141 -t
#run phpunit tests and init the environment
mdk phpunit -r -u mod/assign/tests/locallib_test.php
#run behat tests and init the environment
mdk behat -r --tags=@message_popup
#install community plugins
mdk plugin install tool_policy
#create a new clean branch MDL-61575-master
mdk fix 61575
#force push the commits.
mdk push -t -f
#do the magic backport
mdk backport -v 35
GDPR Privacy Utils for testing

https://docs.moodle.org/dev/Privacy_API/Utilities

Scripts to:
- Test of privacy API compliance
- Test of exporting user data
- Test of deleting user data

Save the scripts into .moodle-sdk/scripts
Run scripts with $mdk run testcompliance.php

```bash
$Mac-de-Carlos:stable_master carlosagile$ mdk run testcompliance.php
$Mac-de-Carlos:stable_master carlosagile$ /usr/bin/git remote -v
$Mac-de-Carlos:stable_master carlosagile$ /usr/bin/git config --get remote.origin.url
Running 'testcompliance.php' on 'stable_master'
The following plugins are not compliant:
  => local_moodlecheck
Testing the compliant plugins:
  == antivirus_clamav (antivirus_clamav\privacy\provider) ==
  Claims not to store any data with reason:
    'The ClamAV antivirus plugin does not store any personal data.'
  == availability_completion (availability_completion\privacy\provider) ==
  Claims not to store any data with reason:
    'The Restriction by activity completion plugin does not store any personal data.'
  == availability_date (availability_date\privacy\provider) ==
  Claims not to store any data with reason:
    'The Restriction by date plugin does not store any personal data.'
  == availability_group (availability_group\privacy\provider) ==
  Claims not to store any data with reason:
    'The Restriction by group plugin does not store any personal data.'
  == availability_grade (availability_grade\privacy\provider) ==
  Claims not to store any data with reason:
    'The Restriction by grades plugin does not store any personal data.'
```
More Tools

Test your code online [https://fiddles.io/#](https://fiddles.io/#)
JSFiddle
SQLFiddle

Chrome Dev Tools
[https://developers.google.com/web/tools/chrome-devtools/](https://developers.google.com/web/tools/chrome-devtools/)

Test APIs and WS with Postman
[https://www.getpostman.com/](https://www.getpostman.com/)

Catch mails with MailCatcher
[https://mailcatcher.me/](https://mailcatcher.me/)
Thank You
Extra Bonus

Rubber Duck Debugging
https://rubberduckdebugging.com

1. Beg, borrow, steal, buy, fabricate or otherwise obtain a rubber duck (bathtub variety)

2. Place rubber duck on desk and inform it you are just going to go over some code with it, if that’s all right.

3. Explain to the duck what your code is supposed to do, and then go into detail and explain your code line by line.

4. At some point you will tell the duck what you are doing next and then realise that that is not in fact what you are actually doing. The duck will sit there serenely, happy in the knowledge that it has helped you on your way.