

- PHP Guardian - Quick Reference Card
- PART 1: FIRST TIME SETUP (Do This Once)
 - Step 1: Open Terminal
 - Step 2: Go to Your Projects Folder
 - Step 3: Download the Guardian Files
 - Step 4: Go Into the Project
 - Step 5: Make It Your Own Project
 - Step 6: Install the Safety Tools
 - Step 7: Create Your Code Folder
 - Done!
- PART 2: OPENING YOUR PROJECT
 - Step 1: Open VSCode
 - Step 2: Open Your Project Folder
 - Step 3: Find the Claude Sidebar
- PART 3: TALKING TO CLAUDE
 - At the Start of Each Day
 - When You Want Claude to Build Something
 - After Claude Writes Code
 - When You Get an Error
 - When You Don't Understand
- PART 4: THE SAFETY RULES
 - What Claude Should ALWAYS Do:
 - What Claude Should NEVER Do:
- PART 5: IF SOMETHING GOES WRONG
 - If You Broke Something
 - If You Want to See What Changed
 - If You Want to Run the Safety Check Manually
 - If You're Completely Stuck
- PART 6: SAVING YOUR WORK
 - Step 1: See What Changed
 - Step 2: Add Your Changes
 - Step 3: Save with a Message
 - If the Safety Check Fails
- HELPFUL PHRASES FOR CLAUDE
- REMEMBER
- EMERGENCY CONTACTS

PHP Guardian - Quick Reference Card

Keep this by your computer. Follow these steps in order.

PART 1: FIRST TIME SETUP (Do This Once)

Step 1: Open Terminal

- On Mac: Press **Cmd + Space**, type **Terminal**, press Enter
 - You'll see a black window with a blinking cursor
-

Step 2: Go to Your Projects Folder

Type this and press Enter:

```
cd ~/Documents
```

(Or wherever you keep your projects)

Step 3: Download the Guardian Files

Type this and press Enter:

```
git clone https://github.com/[THE-LINK-I-SENT-YOU] my-project
```

Replace **my-project** with your project name (no spaces, use dashes)

Wait for it to finish downloading.

Step 4: Go Into the Project

Type this and press Enter:

```
cd my-project
```

Step 5: Make It Your Own Project

Type these two commands, one at a time:

```
rm -rf .git
```

```
git init
```

Step 6: Install the Safety Tools

Type these three commands, one at a time. Wait for each to finish:

```
composer install
```

(Wait... this takes a minute)

```
pip install pre-commit
```

(Wait...)

```
pre-commit install
```

Step 7: Create Your Code Folder

```
mkdir src
```

Done!

Your project is ready. Now open it in VSCode.

PART 2: OPENING YOUR PROJECT

Step 1: Open VSCode

- Click the VSCode icon (blue icon with >< symbol)
-

Step 2: Open Your Project Folder

1. Click **File** in the top menu
 2. Click **Open Folder...**
 3. Navigate to your project folder
 4. Click **Open**
-

Step 3: Find the Claude Sidebar

- Look for the Claude icon on the left side (it looks like a sparkle or star)
- Click it to open the chat panel
- This is where you talk to Claude

PART 3: TALKING TO CLAUDE

At the Start of Each Day

Copy and paste this into Claude:

```
Read the CLAUDE.md file in this project.  
Follow those rules for everything you write today.
```

Then tell Claude what you want to work on.

When You Want Claude to Build Something

Say:

```
I want to [describe what you want].
```

```
Before writing code:
```

- ```
1. Tell me what files you'll create
2. Explain your plan simply
3. Wait for me to say OK
```

---

### After Claude Writes Code

---

Always ask:

Check the code you just wrote for:

- Security problems
- Fake or test data
- Debug code like `var_dump`
- Anything that's not finished

---

## When You Get an Error

---

Copy the error message and say:

I got this error:

[paste error here]

Explain what it means in simple words.  
Show me exactly what to fix.

---

## When You Don't Understand

---

Just say:

Explain that more simply.  
Use everyday words, not programmer words.

Claude will try again with simpler language.

---

## PART 4: THE SAFETY RULES

---

---

## What Claude Should ALWAYS Do:

---

| For This...          | Claude Should Write...                                                    |
|----------------------|---------------------------------------------------------------------------|
| Database queries     | Use <code>\$pdo-&gt;prepare()</code> with <code>:name</code> placeholders |
| Showing data on page | Wrap in <code>htmlspecialchars()</code>                                   |
| Storing passwords    | Use <code>password_hash()</code>                                          |
| API keys & secrets   | Get from <code>\$_ENV['NAME']</code> not in code                          |
| Errors               | Use <code>throw new Exception()</code>                                    |
| Test data            | Use real values or ask you                                                |

## What Claude Should NEVER Do:

| Never This                                           | Why It's Bad             |
|------------------------------------------------------|--------------------------|
| <code>"SELECT * WHERE id = \$id"</code>              | Hackers can break in     |
| <code>echo \$name</code> without escaping            | Hackers can inject code  |
| <code>md5(\$password)</code>                         | Too easy to crack        |
| <code>\$api_key = "sk-abc123..."</code>              | Anyone can steal it      |
| <code>die("error")</code> or <code>var_dump()</code> | Debug code, not for real |
| <code>test@example.com</code>                        | Fake data left behind    |

## PART 5: IF SOMETHING GOES WRONG

### If You Broke Something

Open Terminal and type:

```
git checkout .
```

This undoes all changes since your last save.

---

# If You Want to See What Changed

---

```
git diff
```

Shows what's different. Press **q** to exit.

---

# If You Want to Run the Safety Check Manually

---

```
php scripts/guardian.php
```

This checks all your code for problems.

---

# If You're Completely Stuck

---

Say to Claude:

```
I'm stuck. I was trying to [what you wanted].
I did [what you did].
Now [what's happening].
Help me fix this step by step.
```

---

## PART 6: SAVING YOUR WORK

---

### Step 1: See What Changed

---



In Terminal:

```
git status
```

---

## Step 2: Add Your Changes

---

```
git add .
```

---

## Step 3: Save with a Message

---

```
git commit -m "Describe what you did"
```

Example: `git commit -m "Added contact form"`

---

## If the Safety Check Fails

---

The commit will stop and show you problems.

Ask Claude:

```
The safety check found this problem:
```

```
[paste the error]
```

```
How do I fix it?
```

---

## HELPFUL PHRASES FOR CLAUDE

---

## When You Want To... Say...

---

|                 |                               |
|-----------------|-------------------------------|
| Make a new page | "Create a page that shows..." |
|-----------------|-------------------------------|

---

|             |                                |
|-------------|--------------------------------|
| Make a form | "Make a form that collects..." |
|-------------|--------------------------------|

---

|           |                             |
|-----------|-----------------------------|
| Save data | "Save this to the database" |
|-----------|-----------------------------|

---

|           |                                           |
|-----------|-------------------------------------------|
| Show data | "Show all the [things] from the database" |
|-----------|-------------------------------------------|

---

|           |                  |
|-----------|------------------|
| Add login | "Add user login" |
|-----------|------------------|

---

|               |                                 |
|---------------|---------------------------------|
| Fix something | "This doesn't work: [describe]" |
|---------------|---------------------------------|

---

|                 |                            |
|-----------------|----------------------------|
| Understand code | "Explain this code simply" |
|-----------------|----------------------------|

---

|            |                                  |
|------------|----------------------------------|
| Start over | "Let's try a different approach" |
|------------|----------------------------------|

---

# REMEMBER

---

1. **Claude is patient** - Ask as many questions as you need
  2. **Copy-paste is your friend** - Use the prompts in [.claude/prompts/](#)
  3. **Save often** - Use `git commit` after each working feature
  4. **Ask for explanations** - "Explain that simply" always works
  5. **The safety checks protect you** - If they fail, Claude can help fix it
- 

# EMERGENCY CONTACTS

---

If you're stuck and Claude can't help:

- The person who gave you this guide: \_\_\_\_\_
  - Their phone/email: \_\_\_\_\_
- 

*Print this document and keep it next to your computer.*