What this book will do for you...

- Modify or customize the views and editors quickly
- Learn about the basics of PHP development including class, methods, and getters/setters
- Perform refactoring of elements such as renaming, copying, and moving
- Integrate frameworks using PhpStorm to generate Models, Views, and Controllers
- Use a CVS repository to commit files and check out projects using PhpStorm
- Find out more about the use of PhpStorm in a team
- Learn about safe deployment using Phing and PhpStorm
- Create plugins getter/setter methods, exception breakpoints, and CVS repositories
In this package, you will find:

- The authors biography
- A preview chapter from the book, Chapter 1 'Cooking with PhpStorm'
- A synopsis of the book’s content
- More information on PhpStorm Cookbook

About the Authors

If being knowledgeable and presentable is uniqueness, Mukund Chaudhary is the correct person to be termed unique. Having started his career primarily in programming, he developed an interest in management that led him to pursue an MBA even after obtaining an engineering degree.

Mukund is a team worker and is currently working as a technical product manager at Nityo Infotech Services Pvt. Ltd., New Delhi. His knowledge combined with the presentation skills give him an edge over his contemporaries. He likes to spend his spare time writing or with family. You can follow him on Twitter at https://twitter.com/mukund002.

I take this opportunity to extend my thanks to the strong and warm support provided to me by my family—my parents, my better half, and my grandfather. It was their motivation that led me to complete writing my first book ever.

My friends also deserve a fair share of the thanks. They were always there by my side to provide positive ideas and constructive criticism, as a result of which I was able to complete this tough task with much ease. I would also like to give a special thanks to the editors and all the people who played a part in the publishing of this book.
Ankur Kumar is an apt example of a dark horse. He often outperforms his contemporaries when he is in full flow. A team mentor by role, Ankur works for Nityo Infotech Services Pvt. Ltd. His seat remains vacant most of the time as he remains busy solving the problems of his teammates. Ankur has around 6 years of experience in the IT industry. He has worked in various roles for various companies across the globe. He has been working on LAMP stack for the last 3 years continually. He finds programming a challenge worth taking up and enjoys the one-to-one tussle with problem solving.

Ankur is a strong believer of the fact that a human should always make other humans happy in one way or the other. Even in the middle of tough tasks, he can be found cracking jokes (often on himself) to make the work environment lighter.

If someone is to be acknowledged, it should be the Almighty. It was due to his will that I got the opportunity to convert myself into a part-time author. Somewhere at the back of my mind, I always had this wish to impart the knowledge I gained all these years solving programming problems.

I sincerely thank everyone involved in the book-writing process to have understood the ideas that I wished to convey. The editors do deserve a fair share of thanks for their hard work.

My parents, my sister, and my better half are my real strength, and I will always be thankful to the Almighty for giving me a life worth living.
PhpStorm Cookbook

PhpStorm is a smart PHP IDE that gets your code and understands its structures. It provides code autocompletion, refactoring, syntax highlighting, and much more. PhpStorm makes debugging and testing really easy, especially with debugger configuration validation.

*PhpStorm Cookbook* contains several tutorials that will allow you to run PHP development at a good pace by helping you test and debug, which is often ignored by developers.

The book starts off focusing on how to cook using PhpStorm, gives you an overview of PHP development, and then dives deep into the details of refactoring and building your project using the PhpStorm IDE. After that, it moves to integrating frameworks and testing and debugging them. We then move on to using PhpStorm in a team, using Phing and cooking plugins. We conclude that this book will be of a great help to a PHP programmer who loves to use IDE and looks forward to working faster.

**What This Book Covers**

*Chapter 1, Cooking with PhpStorm,* gives you a quick overview about accessing a project, customizing a view, moving to the previous and next editor locations, and more.

*Chapter 2, PHP Development,* covers PHP classes; methods, including the getter/setter method; code blocks; commenting; and more.

*Chapter 3, Refactoring and Building,* includes renaming, copying, moving, and deleting elements; it includes code searching, comparing, and setting the run configuration.

*Chapter 4, Integrating Framework,* covers the need for a framework and how frameworks such as Symphony and Yii work with PhpStorm.

*Chapter 5, Testing and Debugging,* gives you an overview about installing PHPUnit, testing an application, setting a breakpoint, changing code on the fly, and much more.

*Chapter 6, Using PhpStorm in a Team,* includes creating a CVS repository, connecting PhpStorm to a CVS repository, committing a file, checking out projects, and much more.

*Chapter 7, PhpStorm and Phing,* gives you an overview of connecting Phing to PhpStorm, building an application, setting the Phing editor option, and more.

*Chapter 8, Cooking Library Plugins,* is about validating a plugin, code hinting for the plugin, and more.

*Chapter 9, Code Quality Analysis,* covers the mess detector and the code sniffer; it also covers locating code duplicates and code formatting and rearrangement.
Cooking with PhpStorm

In this chapter, we will cover the following recipes:

- Showing/hiding views
- Customizing a view
- Accessing any project file
- Maximizing views and editors
- Going back to the previous editor
- Going back to the previous edit location
- Linking views to editors
- Creating a key binding
- Creating a bookmark
- Restoring deleted resources
- Customizing with PhpStorm
- Adding an include path
- Excluding unwanted directories from the index
- Using code completion

Introduction

Finally, your hard work will produce some results—the amount of sweat that you had to perspire to code in PHP will now decrease. The reason behind your being unable to love PHP as a programming language was never trivial. Handling PHP without a strong IDE is no less tough than driving at night with the headlights off. It is always said that PHP is a loosely-typed language (God bless those who say this!), yet PHP errors are nightmarish—with all of those mysterious 500 errors, 403 errors, and the like.
But let us not forget that the error code definitely has a meaning. The Apache server is unable to interpret PHP as such, so it gives you an indication that probably the homework is incomplete and that you have to look into your PHP code and find the exact error yourself. You are left adrift in a leaky boat in a stormy ocean... sad!

To be able to come out of this stormy ocean with a leaky boat, you definitely need some aid: a patch, maybe some better material in the form of a good editor with some really smart intelligence; such intelligence will not only inform you about the error, but also help you a great deal in solving the problem. In short, you need a smart IDE, such as PhpStorm, to code in PHP just as the other creatures in the world of programming do.

PhpStorm understands that programmers are not robots, and thus it leaves no stone unturned in making coding easier for them—be it the beautiful syntax highlighting, code formatting, language injection, error inspection, bookmarking lines of code, and all possible permutations and combinations of the thoughts that are currently coming to your mind. Yes, PhpStorm has a solution to all your coding woes, and it ensures that you can love PHP.

PhpStorm is created and maintained by JetBrains, an organization with many similar products in other languages as well. The people at JetBrains respect every community of programmers, and hence they provide IDEs to many programming languages, including Java. PhpStorm is built using the Java programming language and has been carefully designed to be able to support almost all the common and uncommon tools that directly or indirectly provide assistance in programming.

PhpStorm is an intelligent code editor that provides smart code completion, syntax highlighting, extended code formatting configuration, on-the-fly error checking, code folding, support for language mixtures, and so on.

**Showing/hiding views**

A *view*, as the name suggests, is a visual area in PhpStorm that enables you, the PHP programmer, to see the various aspects and/or perspectives of the code.

This includes the following:

- The structure of the code you toiled for
- Your favorites (or quick view shortcuts)
- The future plans in the form of a TODO list
- A bird’s-eye view of your entire PHP project
- How the code changes in the local system and/or the subversioning system
How to do it...

Perform the following steps:

1. Views in PhpStorm can be shown/hidden via the following keyboard shortcuts and some navigation menu items as well as via some mouse clicks:

<table>
<thead>
<tr>
<th>Views</th>
<th>Shortcuts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terminal view</td>
<td>Alt + F12</td>
</tr>
<tr>
<td>TODO view</td>
<td>Alt + 6</td>
</tr>
<tr>
<td>Project view</td>
<td>Alt + 1</td>
</tr>
<tr>
<td>Favorites view</td>
<td>Alt + 2</td>
</tr>
<tr>
<td>Structure view</td>
<td>Alt + 7</td>
</tr>
</tbody>
</table>

2. The best shortcut to access these features is located in the bottom-left corner in PhpStorm and is depicted in the following screenshot:

How it works...

Let us serve the views one at a time so that they can be served hot:

- **The Terminal view**: PhpStorm ensures that you don't leave it for any reason whatsoever. In case you wish to do some command-line activity, such as running some shell commands, setting some cron jobs or whatever, there is this command terminal available inside PhpStorm. Please remember the keyboard shortcut Alt + F12.

- **The TODO view**: To live the life of a programmer, you need to have multiple core processors to be time-bound; infinite cloud storage volume so that whatever is said to you is recorded somewhere eternally; and 32 GB of RAM so that no context switch pushes the active process out of the memory onto the disk. In short, you need to be Jarvis (Iron Man series). Wake up! You are still a human being with a natural tendency to start dreaming and hence stray from the main topic and forget the main course.
A TODO view in the PhpStorm IDE is a feature with which you can record your future plans to make changes in the code. This not only helps you (in most cases) to remember your plans but also to search for them later when required. A TODO feature is just a sticky note that has literally occupied all the space on your workbench. Alt + 6 in your sticky note, please.

The Project view: A project is analogous to a mechanic's workshop, where great things are made in a not-so-great-looking way. Your project contains all sorts of commented-out code, sample logics, test logics, and failed algorithms, which definitely make you smile at the end of the day when you turn off your computer and go home. In PhpStorm, the Project view is a hierarchical view that shows you the exact hierarchy your code has and what files are located at which node in the tree. In order to show the Project view, use Alt + 1.

The Favorites view: A favorite is an item or a list that you need quite frequently, so PhpStorm has provided a feature with which you can very easily look up those items. By default, PhpStorm provides three favorites: the bookmarks in your code, the debugger breakpoints in your code that you might have set, and a list containing other favorite items with the same name as you create your project in PhpStorm. The keyboard shortcut is Alt + 2.

The Structure view: What is more relieving to the mind and the soul than to see the entire code in a consolidated format? The PhpStorm IDE provides a Structure view for code in such a way that all the methods are shown in the form of a list. This enables you to have greater control over the members of any class that you created and prevents namespace collision and method duplication. In order to open the Structure view, all you need to remember is the keyboard shortcut Alt + 7.

The Database view: Although programming a database is one of the most daunting tasks in the world of programming, PhpStorm eases off this pressure by adding a Database view to the system. It lets you view the database tables, the procedures stored in it, the data stored in it, and a number of general tasks related to databases, which would otherwise have required a dedicated application of its own. The database views can be shown or hidden by simple mouse clicks, as shown in the following recipes. You have the freedom to choose the database server of your choice—it could be a server running on your local system or something running over in the cloud over an IP address and protected by a username and password.

Customizing a view

There are times in the life of a software engineer when nothing seems to be going right, and hence the only thought that comes to mind is to change everything.

PhpStorm takes care of that time period as well and provides customization to the views—you can customize the way views appear.
How to do it...

There are two roads that can take you to the world of customization:

- The **Settings** button available in the top-right corner of the individual view, as shown in the following screenshot:

![Settings Button](image)

- The other very easy option and the most familiar friend of yours is the right-click context menu on the title bar of the toolbar.

How it works...

In general, there are some context menu options available to all the views, such as the orientation modes: floating, docked, pinned, and split. The views can be in:

- **Floating Mode**: The views are free to be located anywhere on the screen
- **Docked Mode**: The views do not disappear automatically on losing focus
- **Pinned Mode**: The view stays at one location until changed
- **Split Mode**: You can have multiple views shown at a time

The **Show View as Tabs** option lets you filter or group your tasks based on various criteria, such as project-based, currently-being-edited-based, and scope-based. The **Move** menu allows you to move the view to the top, right, left, or bottom depending upon the current orientation of the view. There are options to stretch or squeeze the view of a window to increase / decrease the size of the view window. The keyboard shortcut is Ctrl + Shift + Up / Ctrl + Shift + Down to increase / decrease the size.
There are individual options available for the views. Stay awake to explore customizations of individual view windows, as shown in the following list:

- **Terminal View**: More terminal windows can be opened depending on your requirements by clicking on the + symbol provided with the Terminal view. If you feel that you need to close the active terminal window, you can very easily do so by clicking on the - symbol on the left-hand side. The commands available for the underlying operating system work fine in this terminal; for example, Linux users can type \texttt{Ctrl + D} to exit the terminal, and Windows users can type in their favorite command in this terminal.

- **TODO View**: The special menu options available for this view are listed here:
  - **Select Next Tab**: This shows the next available view inside the TODO view.
  - **Select Previous Tab**: This is used to switch back to the previous view.
  - **Show List of Tabs**: This shows all views. This provides a dropdown to select from the list of available views inside the TODO view, as shown in the following screenshot:

- **Project View**: If you are in the Project View window, you again deserve more than the general options available. The options are described here. Stay tuned and read on.
  - **Select Next View**: Use this option to go to the next view tab.
  - **Select Previous View**: Use this option to go to the previous view tab.
  - **Show List of Views**: Use this option to select from the list of available views for this window.
- **Show Members**: If you put a tick mark here, you will be able to see the member functions and variables in the file or code.

- **Autoscroll to Source**: This option enables a very useful feature. If you select a file here, the same file will be automatically opened in the editor.

- **Autoscroll from Source**: This is a feature similar to the immediately preceding option. If you select a file in the editor, the Project view will automatically show the same file selected in the Project view.

- **Sort by Type**: This feature, though seemingly trivial, might prove to be very useful in organizing the files and folders in the project currently being worked upon.

- **Folders Always on Top**: You can use this feature to let the folders bubble upwards, and the files are automatically shown under the list of folders.

The following screenshot shows how the Project view looks:

![Project View](image)

- **Favorites View**: This view section provides three special features:
  - **Show Members**: This feature shows the list of members in the selected favorite
  - **Autoscroll from Source**: This is similar to the Project view
  - **Autoscroll to Source**: This is similar to the Project view
The **Favorites** view is as shown in the following screenshot:

![Structure View](image)

- **Structure View**: All you enthusiastic programmers out there should make a habit of using this feature intensively because this is more of an engineering practice than a PHP principle. You should always keep this feature handy, as it definitely is your savior at those times when you feel lost in your hard-labored project. So check your seatbelt, and get ready for this informative ride:
  - **Sort by Visibility**: Use this feature to sort your files and folders according to the visibility of the member functions and variables.
  - **Sort Alphabetically**: Friends, remember that this feature will be useful and helpful only in cases where naming conventions have been followed by the rule book. This feature enables the sorting of member functions and variables alphabetically. A descriptive name for the feature, isn't it?
  - **Show Includes**: This feature shows the files included or required programmatically through PHP code as well.
  - **Show Fields**: This feature show the fields in the class/file.
  - **Show Constants**: This feature shows the constant values in the class/file.
  - **Show Inherited**: This feature shows those files/classes that have an inheritance relationship as well.
  - **Show Toolbar**: This feature shows the discussed features in the form of quick access buttons, as shown in the following screenshot:
Database View: Most PHP programmers might ask—why do I need to have the Database view in PhpStorm? There should have been something like an SQLStorm for SQL databases.

The reason is related to usability. SQL inside PhpStorm is added just to make everything accessible under one roof and stop the programmer from switching back and forth between PhpStorm and SQLStorm. The right-click context menu options available for this feature are:

- **Flatten Schemas**: This feature shows the database schema in a rather flattened way rather than in a tree structure.

- **Group Tables by Type**: This feature groups the tables by their type. This helps in searching for a table in the database and enhances the viewing of the tables.

- **Sort Columns**: This feature sorts columns by name for the database selected.
The **Database** view options are as shown in the following screenshot:

![Database View Options](image)

---

**Accessing any project file**

Software engineering, at times, becomes more about adjustments than creation. There are practical cases where the developer has no option but to switch to some other project created earlier and now wants to use some feature (without infringing the copyrights and without running the gauntlet of legal action).

**Getting ready**

If you need to access any project file in PhpStorm, a few decisions can be made. Accessing the project is more important when you handle multiple projects at a time.

**How to do it...**

If you vaguely remember some of the content that was in the file, you can do a project-wide search to find the target file. Does it make you think of the old story of searching for a needle in a haystack?

1. You can open a folder where your project is supposedly located, as shown in the following screenshot. You can initiate the process by navigating to **File | Open**.
**How it works...**

Don't worry even if the folder is not a PhpStorm project. PhpStorm is smart enough to manage that. Once opened, the project can be searched for the desired file/resource. PhpStorm will remember the folder as a project, and once indexed, the files of that folder can be accessed just as your current project files.

You have a hotkey at your disposal for the purpose: Try Ctrl + Shift + N in times of need, and type in some letters to get a hint list of filenames matching your typed string.

**Maximizing views and editors**

Although the IT industry is flourishing, and candidates from all across the globe are making a living by IT jobs as programmers, companies continue to be misers in the area of providing infrastructure and maintaining proper ergonomics for the programmers. A small monitor is one such ergonomic infrastructure problem. PhpStorm comes to the rescue.
There is a feature to maximize a view or editor at will. The views can be maximized (when in the docked mode) by the resize option available in the right-click context menu or in the settings menu. For easier handling, the mouse drag can also be handy. Just click and drag the required window to resize.

Good practice says that an editor should be the only object that should be kept visible at all times. The other views should be opened on demand and closed as soon as the task is done. The undocked mode is meant for this purpose. The view appears on demand and disappears as soon as another view or editor is clicked on.

There is an excellent feature available with PhpStorm that lets you concentrate solely on your coding by providing a fullscreen mode, which shows you only the PhpStorm window. Yes, you got it right. Only the PhpStorm window is visible, and no frame border or operating system items are visible.

How to do it...

To turn on this feature, you need to perform the following steps:

1. Go to View | Enter Full Screen.
2. To turn off, go to View | Exit Full Screen. Wasn't that easy? Indeed.

An even better feature provided by PhpStorm is the presentation mode. In this, you will only view the editor window with the fonts enlarged. All other features will be available, but only on demand. This is the best viewing option that can be provided by an IDE to work with small or single monitors. Although the motive behind this was to help the programmer demonstrate some coding to an audience, the usage stated in this section is also useful. To turn on this feature, go to View | Enter Presentation Mode, and to turn off, go to View | Exit Presentation Mode.

Going back to the previous editor

It is strongly advised that humans only perform one task at a time. This is said because (most) humans cannot multitask, and hence if they attempt to do so, they make mistakes. The same is the case with you, dear.

Getting ready

In the event that you have to work on a project that demands you modify a number of files at once, there is a big chance that you might forget which editor you came from and where to go if the current file being scanned is not the correct copy.
How to do it...

In PhpStorm, you can go back to the previous editor / next editor / choose from the list of open editors by remembering (and obviously using) keymaps. To do so, perform the following steps:

1. Use Alt + Left for previous editor.
2. If you want to move to the next editor, use Alt + Right.

There's more...

A facility that allows you to move to any part of the code or any class in the open project is available in PhpStorm. This facility enables you to access at random any part of any class or any part of any file that is open in the currently open project, as shown in the following screenshot:

![Search Criteria](image)

And, as usual, there are some customizations involved in the behavior of PhpStorm. There is a filter symbol. Clicking on the filter symbol, quite implicitly, allows you to filter the search criteria. You can tell PhpStorm not to look in a certain file or resource type by simply unchecking the unwanted file or resource type. Refer to the preceding screenshot for clarity.

To enable random access, every item needs to be added to the index. You can now give credit to the PhpStorm development team for developing this powerful indexing system.

Going back to the previous edit location

You can not only switch between the currently opened editors, but also navigate to the last file you made changes to.
**Getting ready**

This is particularly helpful in all those cases where you seem to be lost in the load of work that your manager has assigned to you and who wants it to be done in a very short span of time. All programmers in the world have many of the same woes and worries.

**How to do it...**

In order to access the previous edit location(s), use \textit{Ctrl} + \textit{Shift} + \textit{E}, and select the desired file from the list shown in the pop up. It is pretty easy to understand how it works. Let’s see a screenshot explaining how to go to the previous editor:

![Screenshot of recently edited files]

**Linking views to editors**

Sometimes, the name is self-explanatory. Sometimes, the sentences have a hint. If you got the hint, you must have understood that this feature provided by PhpStorm is intended to show you exactly where in the project hierarchy the code is. If you did not get the hint, don’t worry—you’ve now got it since you’ve read the text. Please stay tuned for more tasty recipes.

PhpStorm provides two ways of linking between view and editor. PhpStorm can highlight the location of the file you are currently editing in the project tree hierarchy shown in the Project view.

**How to do it...**

PhpStorm can also open the file in the editor as soon as you select it in the Project view. Perform the following steps:

1. To use this feature, you are advised to read the explanation of \textit{Autoscroll to Source} given in the \textit{Customizing a view} recipe of this chapter.

2. This linking system also applies to the class members (the functions and variables) as well. PhpStorm can autoscroll (or link) between the editor and the view and hence ease some of the pressure of development from your shoulders. So brotherly!
Creating a key binding

For all those people who find key bindings difficult to visualize, key bindings mean keyboard shortcuts, and they are the gear changers in a programmer's life. When you are concentrating hard on the project that you have to complete within a stipulated amount of time, you can bet anything in this world for a way or a means by which your speed of working gets affected positively. Bet on keyboard shortcuts. You will be on the winning side.

PhpStorm, again, specifies some default keymap shortcuts. Don't worry friends! You can very easily customize the keyboard shortcuts to suit your taste.

How to do it...

1. Use the keyboard shortcut Ctrl + Alt + S or choose from the File menu.
2. Go to the Settings dialog box and type keymap in the search box provided at the top and PhpStorm will highlight the target for you.
3. Choose from a list of the default keymap settings, as shown in the following screenshot, but better late than never. Before jumping to change the shortcuts, understand how shortcuts work in PhpStorm. The keymap shortcuts provided by PhpStorm by default are not directly editable, so in order to customize them, you need to make a copy of the default settings. Sounds confusing? The PhpStorm wizard creates the copy for you. Your mind prompts a question—where are my settings saved then? Here comes the answer—under *nix based systems.
How it works...

PhpStorm uses the directory `~/.WebIde70/config/keymaps` to store all the keymap schemes that you create. Still not feeling confident? Here are the tricks of the trade.

PhpStorm will copy it for you. It will ask you for a name for this keymap scheme. The name you specify will be respected by PhpStorm. A careful observation would easily tell you that PhpStorm will always make you remember that your current selection is based on the setting provided by PhpStorm. After you have made the selection to copy and modify, all you have to do is to press the Apply button. Bingo! You just created a copy of one default setting, and now you are all set to customize PhpStorm behavior your own way.

To verify that, you can go to `~/.WebIde70/config/keymaps` and see that PhpStorm just created a new XML file for you that contains the settings you created via the PhpStorm GUI. Behind every successful frontend, there is an equally strong backend!

If you think that you can speak XML, you can create some settings via the GUI and then view the contents of the XML file to observe how the settings are saved. You can later add items directly to the XML file. You naughty hacker! Control your smile!

Creating a bookmark

Bookmarks are great. A bookmark lets you put a flag somewhere in your code so that you have a shortcut to access that code.

A bookmark in PhpStorm has the same meaning as that in a book where a reader puts some indication in the form of some piece of paper, by turning the page corner, or by any means through which the reader can access the page directly.

How to do it...

PhpStorm bookmarks are created with ease. The add bookmark option can be had by going to Navigate | Bookmarks.

The three options available are explained here:

- **Toggle Bookmark**: This feature enables you to create a bookmark at some specified line of code if it does not exist and to delete a bookmark from some line of code if it does.
Chapter 1

- **Toggle Bookmark with Mnemonic:** This feature enables you to toggle a bookmark with the added ability to be able to access that bookmark using keyboard shortcuts. Suppose you choose the alphanumeric 0, as shown in the following screenshot, you will be able to access that bookmark by using Ctrl + 0

```
public function asseenAction()
{
    $this->loadLayout();
    $block = $this->getLayout()->getBlock($this->getLayout())->getBlock();
    $this->renderLayout();
}
```

**Show Bookmarks:** This is rather a simple feature. It simply lets you view all the bookmarks you created for the project you are working on.

Quick access to bookmark
In order to have quick access to the bookmark feature with the mnemonic feature, all the hard work that you have to do is remember the keyboard shortcut Ctrl + F11. The rest will be taken care of by PhpStorm automatically.

**Restoring deleted resources**

To err is human. To recover from the error cleanly is PhpStorm. It is perfectly okay for you to commit mistakes. You could be thinking of your ex or your reporting manager's atrocities on you to make you work overtime (while he himself keeps tinkering with social networking sites), and you happen to accidentally delete some lines from your code. You have no idea what you did, and with a heavy heart, you went home at the close of day. The very next morning, when you are about to start work, you suddenly realize that you deleted some lines of code and you don't remember which! Is this something like salt on burnt skin? Need ointment? Breathe and thank PhpStorm's team of developers who knew that you are a human prone to making drastic mistakes like these.
**Getting ready**

You can recover the lines deleted from your code. Yes, you got that right. There is the history feature in PhpStorm that is one of the best in its class in terms of the visual assistance it provides in finding the change that has occurred to your code—be it local, that is, you were the culprit behind the notorious code change, be it from the team, that is, someone else’s code ruined yours. PhpStorm helps you point out the mistake or error and helps you in fixing things quickly.

**How to do it...**

PhpStorm keeps a snapshot of every code that you write. It updates it every time you save. So, if you need to undelete a piece of code, you need to dive into the history using the times of editing the files as anchors. As soon as you change to another window it automatically saves the document...wasn't that cool. Perform the following steps:

1. The **Local History** option is available on the mouse right-click context menu.
2. As soon as you click on or select a particular time, PhpStorm shows you the difference between that file from history and your current file, as shown in the following screenshot:
3. You can revert to (this) particular item by selecting the **Revert** option from the right-click context menu. Done!

4. You can also merge the code line by line by double-clicking on the greater-than sign (>).

**How it works...**

If you deleted a file from PhpStorm, knowingly or unknowingly, you can use the same feature to recover your deleted files. You need to select the target folder or directory from the Project view, access the right-click context menu, and select the **Local History | Show History** option. PhpStorm will not only show you all the files that you deleted very easily, but also in a hierarchical way, as shown in the following screenshot:
All you need to do is access the right-click context menu on the file that you want to recover from the list and choose the **Revert Selection** option. No more cigarettes and coffee combinations, please. Your headache is solved.

---

**Customizing with PhpStorm**

Sometimes thinking enough is not enough. Sometimes doing enough is not enough. The point is, despite learning so much, you still have a feeling that some more customization could have been possible. PhpStorm is at your disposal.

**Getting ready**

The best thing you can do is use PhpStorm for whatever PHP projects you take up. But therein the problem looms large. Executing PHP via a web server is OK, but what do you do about the use case when the script has to run in the background? What if your reporting manager wants you to do some socket programming? You need a command-line interpreter for PHP. Stunned silence? It's time to break that silence. Thank PhpStorm and integrate the command-line PHP interpreter with PhpStorm.

**How to do it...**

Just a small sequence of shortcuts is all that you need to invoke command-line PHP. There are actually two roads, that is, there is a bifurcation ahead. From the bifurcation, you will see two roads—on the left-hand side is the *PhpStormish* way, and on the right-hand side is the *operating systemish* way. The point worth noting is that both ways take you to the same destination. At the destination, you will attain your goal, which is to have the capacity to use PhpStorm for every PHP task.

You will be taken through the PhpStormish way first. This is easier, so you should try it first. This is termed as PhpStormish because PhpStorm lets you think that you are using PhpStorm itself to run your hard toiled PHP code. Perform the following steps:
1. In order to run command-line PHP this way, you need to go to **Settings | PHP**, as shown in the next screenshot:

2. There is this interpreter dropdown. You need to add an interpreter. You can do that by selecting the PHP interpreter from the **bin** folder of the PHP installation.

3. Once selected, you need to customize it by providing various parameters, such as a name and configuration options. On applying the changes, the PHP environment in PhpStorm will be powered by the PHP environment of your underlying operating system.

4. This will enable you to test and run your PHP scripts using the default keyboard shortcut **Ctrl + Shift + F10**.
Cooking with PhpStorm

See also

You can add more interpreters if you have multiple versions of PHP installed and select from those for execution.

That was easy to cook, right? Feeling happy? In this happiness, please don't forget that there was one more road, which was on the right-hand side, and which again took us to the same destination. PHP scripts or codes can be run on the terminal. Terminal—did this word strike bells in your mind? PhpStorm provides you a terminal (emulator) such that you do not have to switch back and forth between your beloved IDE (presumably) PhpStorm and the command-line terminal. You can again use this feature to execute command-line PHP scripts. All you have to do is type `php your-php-file.php` if you have the PHP classpath set, or else you need to add some extra letters to make it `/path/to/installed-php/php your-php-file.php`.

The following screenshot shows you how to run PHP through command line in PhpStorm:

![Screenshot showing how to run PHP through command line in PhpStorm](image-url)
Chapter 1

Adding an include path

One of the software engineering principles says that components should be reusable, that is, the structure of the software should be modular. This is a practice accepted worldwide, but this adds an overhead to the translation (interpreter or compiler) to link the modules together by first finding them. PHP is no different.

Getting ready

The PHP interpreter has to search for the required files/folders/resources every time the script is run. PhpStorm can, however, decrease the time required to search for the files/folders by allowing you to set the include path. The include path is the base directory under your operating system that contains all the files and folders to the dependencies.

How to do it...

You have the freedom to set this in PhpStorm. Go to Settings | PHP | Include Path. Select the new path. Click on Apply. Done! Take a look at the following screenshot, which illustrates this point:

![Screenshot of PhpStorm settings with include path section open]
You should always make sure that you plan how to organize your files and folders beforehand so that a minimum number of include path settings are required.

If a folder is provided in the include path, it will be helpful in autocompletion for `include()`, `require()`, and so on.

**Excluding unwanted directories from the index**

Did you try imagining why on earth PhpStorm has information about all the files you added in your project directory structure? How does PhpStorm detect the changes that you make to the disk by adding images, text files, and so on, and manages to show them in the Project view? Yes, the answer is indexing.

**Getting ready**

PhpStorm indexes everything that comes its way. Thus, it remembers the files and folders properly. But there is an overhead involved in this system. The more the number of files in the project, the longer the index created is. If PhpStorm decreases time in searching for files and folders by maintaining an index, how will it manage searching in the index itself when the index list has increased to a very large size?

**How to do it...**

You can help PhpStorm by performing the following steps:

1. Go to **Settings | Directories**.
2. Select a directory, and all you have to remember from here are three keyboard shortcuts: Alt + T, Alt + E, and Alt + R. Consider the following screenshots:
Alt + T will mark a resource as a test resource, E will exclude it from indexing, and R will mark it as the content root. For you, the increase in speed will come from the magic key combination Alt + E. Be ruthless enough to apply this magic to all those directories that you decide are not important enough to be indexed.

A test resource is a type of resource, most probably some PHP script, some directory, or some other file type that is used in the testing process. So, you can mark a resource as a test resource once you know that it will be useful for testing purposes.

PhpStorm, being a carefully written piece of software, maintains a list of all the files that you save in your project, using indexes. So, a large number of files means a longer index and hence the potential danger of slowing PhpStorm down. An excluded resource is a resource—some file(s) or folder(s)—that you have asked PhpStorm not to index.

A content root is a folder or directory under which the contents of the resources you are working on is contained. You would easily be tempted to ask "what purpose does it serve?" The answer is rather simple: it maintains a clean workspace for you (everything in life need not be complicated)!
Using code completion

The best example of the need to use an IDE instead of a standard editor is that an IDE understands your code and thus provides hints about the code that you might be writing or need next.

Getting ready

A standard editor would, at the maximum, provide basic text coloring based on the various keywords. Anyway, without hurting the feelings of editors, PhpStorm provides you with additional options besides providing you the standard code completion system.

How to do it...

There is the smart type completion that not only completes your code, but also finds and suggests the most eligible match for your code. By eligible, it means that the code hint matches the current context (the invoking object). This is particularly useful in cases in which you have honestly stuck to the Software Engineering principle of Inheritance.

Code completion settings in PhpStorm are available at File | Settings | Editor | Code Completion, as shown in the following screenshot:
How it works...

If you have a lot of items for PhpStorm to look up, for example, members of your created class, you can order PhpStorm to provide you with hints in lexicographical order.

To turn this feature on, you need to select the checkbox adjacent to **Sort lookup items lexicographically**. The following screenshot shows the code completion feature:

```
<?php
/** Created by PhpStorm. ...*/
namespace Controllers;

class BookWriting {
    function __construct(){
    }

    public function
    *    clone() mixed
    *    construct() void
    *    __destruct () void
    *    __get (name : string) mixed
    *    __invoke () mixed
    *    __isset (name : string) bool
    *    __set (name : string, value) void
    *    __set_state (an_array : array) mixed
    *    __sleep () array[NULL]
    *    __toString () string
    }
```

Code completion at your convenience! I hope this make sense to you.
Where to buy this book

You can buy PhpStorm Cookbook from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.