

# Performance

# Code Profiling

- ❖ When profiling our code we will find what is causing it to be slow. There are more than a few profiling tools we can use.
- ❖ The Zend Studio is already bundled with a profiling tool easy to use.



# Code Profiling

The screenshot displays the Zend Studio interface with a PHP profiler active. The 'Execution Flow' tab is selected, showing a tree view of the execution process and a table of execution statistics.

Function	File	Total execution time	Duration time (ms)
(main)	demo.php	98.62%	2277.46
a	demo.php	98.62%	2277.38
b	demo.php	0.0%	0.02
c	demo.php	0.0%	0.01

The code editor at the bottom shows the source code for 'demo.php':

```
5  
7  
8 function b()  
9 {  
10     return c(5,6);
```

# Caching

- ❖ When using caching then instead of regenerating a web page each and every time a request for that page arrives, the PHP script generates it the first time it is asked to and then it stores a copy. The next time a request for the very same page arrives the static page that was already created is sent back.

# Caching

- ❖ There are various ways for implementing caching in our web application. The simplest would be implementing it by ourselves in our code.
- ❖ When implementing caching by ourselves it is highly important to regenerate the web pages when needed.
- ❖ There are cases in which caching is not relevant (search engines, forums etc.).