

# p\_slides

## dead simple way to create semantic, nice to look at slides

- forget about styling, only think about content
  - *write markdown (uses [showdown.js](#))*
  - *slides are automatically styled nicely (uses [twitter bootstrap](#))*
  - *code samples are syntactically highlighted (uses [jquery syntax](#))*
  - *generates nice slides in the browser (uses [slidy.js](#))*
    - use arrow keys to navigate
    - use the generated 'table of contents' for quick navigation
    - supports printing to pdf (see [presentation.pdf](#))

# usage

- edit presentation.html to create your content
  - use [markdown syntax](#) *with* [table extension](#)
  - *create page breaks using '---'*
- open presentation.html in your favourite browser
  - *tested in current versions of chrome/safari/ff*
- if need be, print the document to pdf
  - *slides will automatically get separated into pages*

# syntax highlighting

- write your code in <pre> tags
- annotate the given language using a css class

## example code

```
<pre class="syntax c">
static int foo;
void bar(void) {
    foo = 0;
    while (foo != 255) ; }
</pre>
```

## becomes

1.	static int foo;
2.	void bar(void) {
3.	foo = 0;
4.	while (foo != 255) ; }

# syntax highlighting

- if you want to syntax highlight *all* your code in the same way then you can uncomment and customize the following line at the bottom of the presentation.html file:

```
$('#pre > code').parent().addClass("syntax cpp");
```