majsdown

Metaprogramming? In my slides?

CppCon 2022

Lightning Talk

Vittorio Romeo

mail@vittorioromeo.com

@supahvee1234

Bloomberg Engineering

TechAtBloomberg.com

Careers

Motivation

- I like writing slides in Markdown + Reveal.js
  - Can be managed in version control
  - More productive compared to WYSIWYG
  - Easy to deal with code and syntax highlighting
- I also get all the power of HTML and CSS
  - Useful, but verbose and not DRY
- But I cannot do anything “fancy”! How about...
  - ...automatically generating Compiler Explorer links for my snippets?
  - ...creating shorthand notation for commonly repeated patterns?
  - ...applying a transformation to a code block?
  - ...embedding a file directly in the Markdown source?
- `majsdown` is here to help!
  - *“The unholy union between Markdown and Javascript.”*

 (C) 2022 Bloomberg Finance L.P. All rights reserved.

A compelling example (#1)

- I want to ensure the validity of the code in all my slides
  - And also give people a chance to experiment with it
- Therefore, I want to add a Compiler Explorer link for each snippet

````
```cpp
#include <iostream>

int main()
{
    std::cout << "I should be using <format>...\n";
    return 0;
}
```

[on Compiler Explorer](https://gcc.godbolt.org/WHATEVER)
````

- Poor solution
  - Need to manually generate the link
  - Need to manually keep the code and the link in sync

 (C) 2022 Bloomberg Finance L.P. All rights reserved.

A compelling example (#2)

- I thought about this problem a bit
- What if I create a script to automatically generate links?
- But not all snippets are created equal...
  - What if I want to hide some includes, or the `main` function?
- I needed a more powerful solution
- So of course I wrote a Markdown preprocessor in C++...
  - ...with an embedded Javascript engine!

````
@@_{godboltify(code)}_
```cpp
#include <iostream>

int main()
{
    std::cout << "I should be using <format>...\n";
    return 0;
}
```
````

(C) 2022 Bloomberg Finance L.P. All rights reserved.

A compelling example (#3)

- The `@@_{godboltify(code)}_` syntax is a “code block decorator”
  - It extracts the contents of a code block
- They can then be passed to any arbitrary JS function, such as `godboltify`
- The JavaScript can be embedded directly inside the Markdown

````
@@${
function godboltLink(code)
{
    return "[on godbolt](" + String.raw`https://godbolt.org/clientstate/
        ${Base64.encode(godboltJson(code))}` + ")";
}
}$

@@_{godboltify(code)}_
```cpp
int main()
{
    // ...
}
```
````

vittorioromeo.com | mail@vittorioromeo.com | vromeo5@bloomberg.net | @supahvee1234 | (C) 2022 Bloomberg Finance L.P. All rights reserved.

A compelling example (#4)

- Of course, the rest of the JavaScript can be embedded as well
  - Or it can also be included from a `.js` file

```
@@${
const Base64 = {/* ... */};

function godboltEscape(src)
{
    return src.replaceAll('"', '\\"').replaceAll("\\n", "\\\\n").replaceAll("\n", "\\n");
}

function godboltJson(src)
{
    return `{"sessions": [{
        "language": "c++",
        "source": "${godboltEscape(src)}",
        "executors": [
        {
            "compiler": {
            "id": "clang_trunk",
            "libs": [],
            "options": "-std=c++20"
        }}]}]}`;
}
}$
```

 (C) 2022 Bloomberg Finance L.P. All rights reserved.

A compelling example (#5)

````
@@_{godboltify(code)}_
```cpp
#include <iostream>

int main()
{
    std::cout << "Automatic Godbolt!\n";
}
```
````

```cpp
#include <iostream>

int main()
{
    std::cout << "Automatic Godbolt!\n";
}
```

on godbolt

Live demo! (#1)

Hello conference!

Today’s date is 2022-09-16.

- My contacts:
  - https://vittorioromeo.com
  - mail@vittorioromeo.com
  - @supahvee1234

Live demo! (#2)

- Decorating a code block:

```
bsl::string greeting = "hello world";
bsl::cout << greeting << bsl::endl;
```

- Embedding a file into the slides:

```
#include <iostream>

int main()
{
    std::cout << "Hello CppCon 2022! I come from a file!\n";
}
```

 (C) 2022 Bloomberg Finance L.P. All rights reserved.

Thanks!

- **https://github.com/vittorioromeo/majsdown**

- Pragmatic Simplicity
  - *Actionable Guidelines To Tame Complexity*
- Thursday 15 @ 15:15 MDT
  - Aurora A / Online A

- *"Embracing Modern C++ Safely"* book signing
  - by J. Lakos, R. Khlebnikov, A. Meredith, & other contributors
- Tuesday 13 @ 12:00 MDT
  - Aurora A / Online A

- Let's keep in touch!
  - https://vittorioromeo.com
  - mail@vittorioromeo.com
  - @supahvee1234