

Heading 1:

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Below is an underline:

Blockquote Level 1: Info

Blockquote Level 2: Danger

Blockquote Level 3: Hint

Wassupppp testing inline stuff: **here's highlight**, here's bold, *here's italic*.

Below is an image:



Below is a mutli-line codeblock with syntax highlighting:

```
use std::iter::Peekable;

use super::ast::Node;

pub fn parse_inline_code(line: &str) → Option {
    if line.trim().starts_with("`") && line.trim().ends_with("`") {
        return Some(Node::Code {
            language: "inline".to_string(),
            content: vec![Node::Text(line[2..line.len() - 1].to_string())];
        });
    }

    None
}

pub fn parse_multiline_code(lines: &mut Peekable) → Option
where
    I: Iterator>,
{
    // Grab the first line (which should be the ```)
    let first_line = lines.next()?.ok()?;
    if !first_line.trim().starts_with("```") {
        return None;
    }
}
```

```

let lang = first_line.trim().trim_start_matches("```").to_string();
let mut content = vec![];

// keep consuming lines until we hit closing ```
while let Some(Ok(peek_line)) = lines.peek() {
    if peek_line.trim().starts_with("```") {
        lines.next(); // consume the closing ```
        break;
    }

    let line = lines.next().unwrap().unwrap();
    content.push(Node::Text(line));
}

Some(Node::Code {
    language: if lang.is_empty() {
        "unknown".to_string()
    } else {
        lang
    },
    content,
})
}

```

Below is a custom take on generating tables:

// Made With the following syntax:

```

:::table
heading1, heading2
value1, value2
value3, value4
:::

```

heading1	heading2
value1	value2
value3	value4

