

A photograph of a woman with long dark hair tied back, wearing a light-colored top, sitting on a wooden deck. She is looking directly at the camera with a slight smile. The background shows a residential area with houses, trees, and a clear sky. The lighting suggests it's either sunrise or sunset, creating a warm glow.

Create beautiful presentations:

*Supendeck*

If you want to go fast, go alone. If you want to go far,  
go together.

African Proverb



This is an example of an awesome widget



Awesome Widget

# Two Column Header

This is your main header, providing a context or introducing the core concept covered in this slide.

## Left Heading

- Point A
- Point B
- Point C

## Right Heading

- Point X
- Point Y
- Point Z

A close-up photograph of a branch with small, oval-shaped green leaves. The branch is positioned at the top left of the frame. The background is a soft, out-of-focus green with a subtle bokeh effect, creating a natural and fresh feel.

# Key Features

- Innovative design
- User-friendly
- Energy-efficient

**Bold Text**

*Italic Text*

~~Strikethrough~~

Inline Code

[Link](#)



If you want to go fast, go alone. If you want to go far,  
go together.

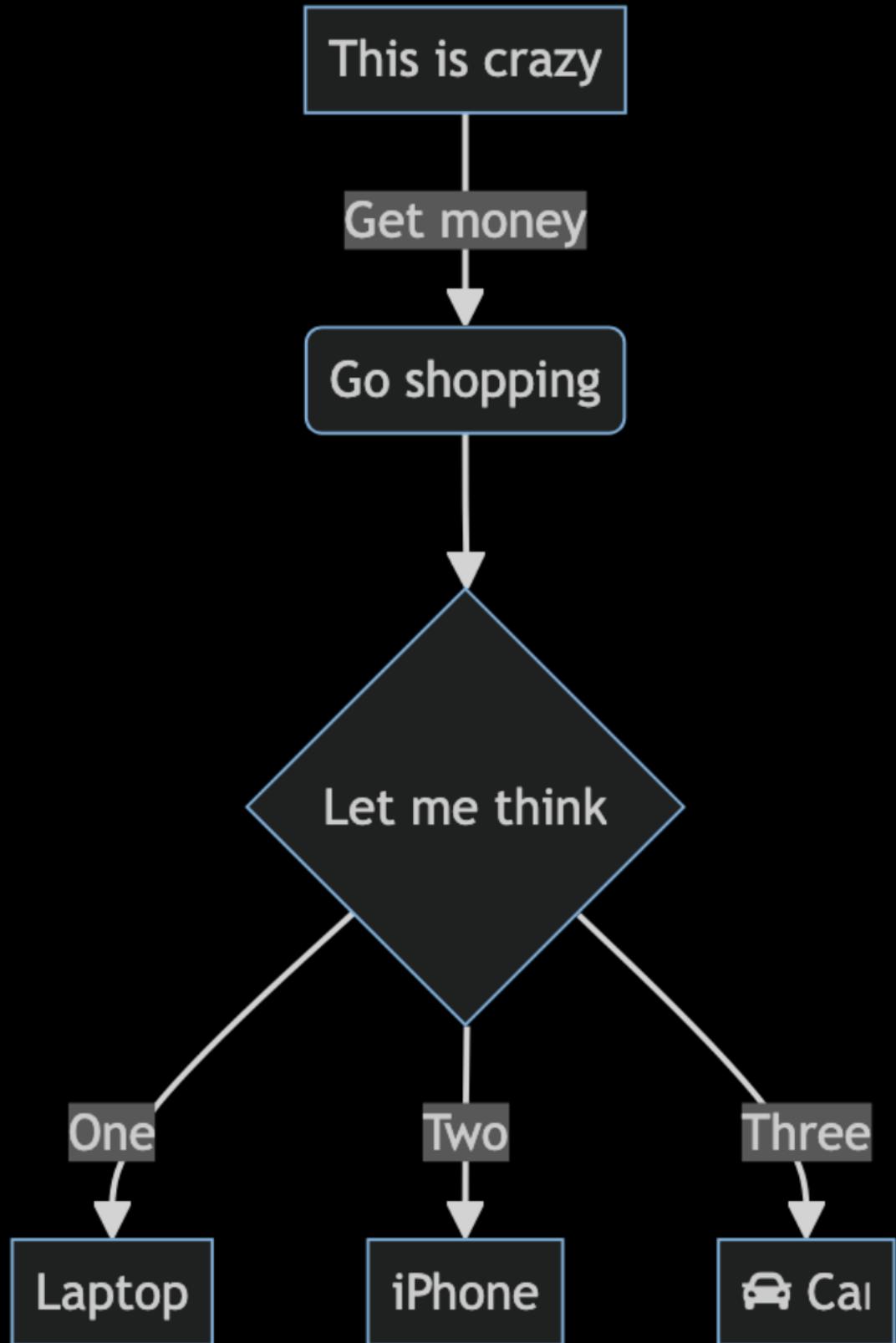
African Proverb



```
int factorial(int n) {  
    if (n == 0) {  
        return 1;  
    } else {  
        return n * factorial(n - 1);  
    }  
}
```

1. Ordered list item 1
2. Ordered list item 2
  - Unordered list item 1
  - Unordered list item 2

<b>Header 1</b>	<b>Header 2</b>
Cell 1A	Cell 1B
Cell 2A	Cell 2B



# Mermaid Example

## Subtask

Task List Item 1

Task List Item 2

foo

bar

baz

bim

---

## Dividers

---

```
class SyntaxTags {  
    const SyntaxTags._();  
    static final left = '::left::';  
    static final right = '::right::';  
    static final content = '::content::';  
}  
  
Map<String, List<String>> parseContentWithTags(  
    String input, List<String> tags) {  
    final Map<String, List<String>> parsedContent = {};  
    int lastTagEndIndex = 0;  
    String currentTag = SyntaxTags.content;  
  
    for (int i = 0; i < input.length; i++) {  
        for (String tag in tags) {  
            if (input.substring(i).startsWith(tag)) {  
                // Add the content before this tag to the list  
                final content = input.substring(lastTagEndIndex, i).trim();  
                if (content.isNotEmpty) {  
                    parsedContent.putIfAbsent(currentTag, () => []).add(content);  
                }  
  
                // Update the current tag and last tag end index  
                currentTag = tag;  
                lastTagEndIndex = i + tag.length;  
            }  
        }  
    }  
    return parsedContent;  
}
```