

# GoWASM ハンズオン

2024/11/17

DevFest Shikoku 2024 GoWASM ハンズオン

Tam@GDG四国

# 自己紹介

Tam@GDG四国

うどんの国からやってきました。

# 早速作ってみましょう。

```
package main

import "fmt"

func main() {
    fmt.Println("Hello, World!")
}
```

## コンパイル

```
GOOS=js GOARCH=wasm go build -o main.wasm main.go
```

Go でコンパイルされた WASM コードを実行するための JS ロードーをコピーします。

```
cp "$(go env GOROOT)/misc/wasm/wasm_exec.js" .
```

※ロードーは必ずコンパイルした環境のものを使ってください。

# ブラウザで動かす

```
<html>
<head>
  <meta charset="utf-8">
  <title>Go WebAssembly</title>

  <!-- wasmファイルを扱うために必要なjsファイルを読み込む -->
  <script src="./wasm_exec.js"></script>
  <script>
    const go = new Go();
    WebAssembly.instantiateStreaming(fetch("markdown-to-html.wasm"), go.importObject).then((result) => {
      const module = result.module;
      const instance = result.instance;
      go.run(instance);
    });
  </script>
</head>
<body><h1 id="tag">Go WebAssembly</h1></body>
</html>
```

# WASM が終了してしまわないようにしましょう！

```
package main

import "fmt"

func main() {
    fmt.Println("Hello, World!")
    println("Go!")
    c := make(chan struct{})
    <-c
}
```

## WASM から JS を操作してみよう！

```
func hello() {  
    // document.body.innerHTML = "<h2>Hello!</h2>";  
    js.Global().Get("document").Get("body").  
        Set("innerHTML", "<h2>Hello!</h2>")  
}
```

## JS から WASM を呼び出してみよう！

```
func start() {  
    // document.getElementById('tag').addEventListener("click", hello);  
    js.Global().Get("document").  
    Call("getElementById", "tag").  
    Call("addEventListener", "click",  
        js.FuncOf(func(this js.Value, args []js.Value) any {  
            hello()  
            return nil  
        })))  
}
```



**以上です。**

基本はこれだけ。

あとはお好みで組み合わせていきましょう！