# Templating Basics



Michael Van Sickle

@vansimke



#### Introduction



**Purpose of templates** 

**Basic functions** 

Loading

**Subtemplates** 

**Template composition** 



Template

Data



Document



```
Dear {{.FirstName}},
                                type Data struct {
                                   FirstName string
We would like to personally
invite you to {{.Event}}!
                                              string
                                    Event
                                    AuthorName string
Sincerely,
{{.AuthorName}}
```



```
Dear {{.FirstName}},
                                type Data struct {
                                   FirstName string
We would like to personally
invite you to {{.Event}}!
                                              string
                                    Event
                                    AuthorName string
Sincerely,
{{.AuthorName}}
```



```
Dear Amanda,
                               type Data struct {
                                   FirstName string
We would like to personally
invite you to a party!
                                    Event string
                                   AuthorName string
Sincerely,
Franklin
```



**Template** 

**Template** 

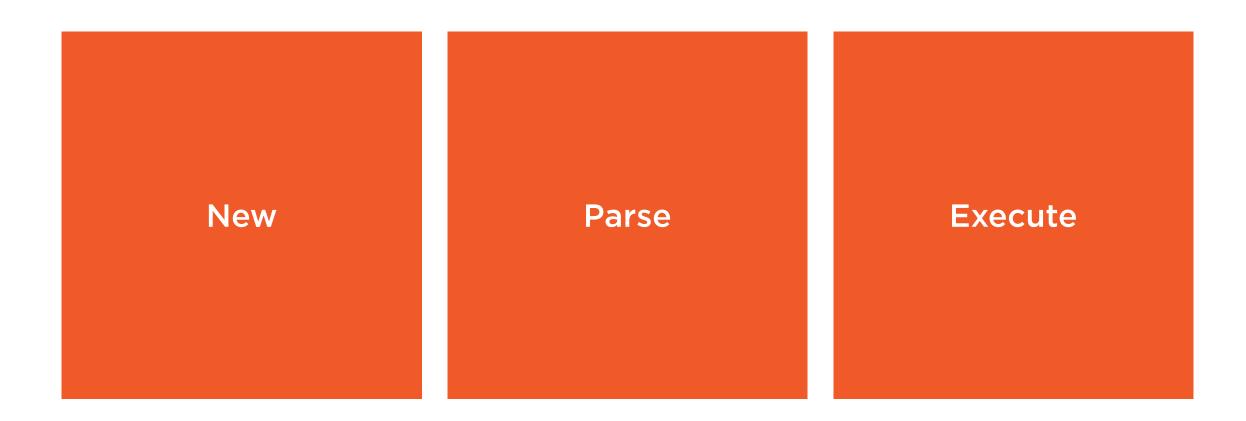
**Template** 

Data

Document



#### Basic Functions





# Loading Templates

**ParseGlob ParseFiles** Lookup Must



# Subtemplates

```
Template
<!DOCTYPE html>
<html>
<head>
      <title>Lemonade Stand Supply</title>
      <link rel="stylesheet" href="..."/>
</head>
<body>
      <div id="content"></div>
      <script>...</script>
      <script>...</script>
</body>
</html>
```

### Subtemplates

```
Template
<!DOCTYPE html>
<html>
<head>
      <title>Lemonade Stand Supplv</title>
      <link rel="stylesheet" hre {{define "head"}}}</pre>
                                           <head>...</head>
</head>
<body>
                                    {{end}}
      <div id="content"></div>
      <script>...</script>
      <script>...</scrip {{define "scripts"}}</pre>
</body>
                                  <script>...</script>
</html>
                                  <script>...</script>
                           \{\{end\}\}
```

# Subtemplates

```
Template
  <!DOCTYPE html>
  <html>
  {{template "head"}}
  <body>
         <div id="content"></div>
        {{template "scripts"}}
  </body>
  </html>
Warning: Reusing a template name overwrites original!
```



```
Subtemplate
                                           Flow of page
                                             preserved
<!DOCTYPE html>
<html>
{{template "head"}}
<body>
                                          Duplicated code
      <div id="content"></div>
      {{template "scripts"}}
</body>
</html>
```

```
Layout template
<!DOCTYPE html>
<html>
{{define "head"}}{{end}}}
{{template "head" .}}
<body>
      <div id="content">
            {{template "content" .}}
      </div>
      {{define "scripts"}}{{end}}
      {{template "scripts" .}}
</body>
</html>
```



```
Layout template
<!DOCTYPE html>
<html>
{{define "head"}}{{end}}}
{{template "head" .}}
<body>
      <div id="content">
            {{template "content" .}}
      </div>
      {{define "scripts"}}{{end}}
      {{template "scripts" .}}
</body>
</html>
```



```
Layout template
<!DOCTYPE html>
<html>
{{block "head" .}}{{end}}
<body>
      <div id="content">
            {{template "content" .}}
      </div>
      {{block "scripts" .}}{{end}}
</body>
</html>
```

```
Required child
{{define "content"}}
....
{{end}}
```

```
Optional child
{{define "head"}}
....
{{end}}
```



Template

Document

Data



```
t := template.Must(template.New("").Parse("..."))
t.Execute(w, nil)
```



```
type Context struct {
   Title string
   ImageURL string
context := Context{Title: "The Title", ImageURL: "http://..."}
t := template.Must(template.New("").Parse("..."))
t.Execute(w, context)
```



```
type Context struct {
   Title string
   ImageURL string
context := Context{Title: "The Title", ImageURL: "http://..."}
t := template.Must(template.New("").Parse("..."))
t.Execute(w, context)
```



```
<!DOCTYPE html>
<html>
<head>
    <title>{{.Title}}</title>
</head>
<body>
    <img src="{{.ImageURL}}"/>
</body>
</html>
```

```
type Context struct {
    Title string
    ImageURL string
}
```



### Summary



Purpose of templates

**Basic functions** 

Loading

**Subtemplates** 

**Template composition** 

