Templates:
Prototype document elements

Joseph Wright
The LaTeX Project

joseph@texdev.net

Templates?

- Many documents have similar features: elements

Templates?

- Many documents have similar features: elements
- Copying code every time is error prone

Templates?

- Many documents have similar features: elements
- Copying code every time is error prone
- But there are both smaller and larger variations

Templates?

- Many documents have similar features: elements
- Copying code every time is error prone
- But there are both smaller and larger variations
- *Templates*: mechanism to swap out how elements are implemented

Template steps

1. Declare a template *type*: the ‘thing’

Template steps

1. Declare a template *type*: the ‘thing’
2. Create one or more interfaces to flesh out the type

Template steps

1. Declare a template *type*: the ‘thing’
2. Create one or more interfaces to flesh out the type
3. Define the implementation for the interface

Template steps

1. Declare a template *type*: the ‘thing’
2. Create one or more interfaces to flesh out the type
3. Define the implementation for the interface
4. Use what you’ve made: see later

Template types

- There are only really a small number of template types
- As such, it’s likely most will be provided by the kernel
- Little code here: it’s about defining semantics

Template interfaces

```
\DeclareTemplateInterface{type}{template}{no.~of args}
  {key list}
```

- This step is to help designers/users: no code
- Keys will flexibility of this template
- Different templates for the same type can have different keys …

Template interfaces

```
\DeclareTemplateInterface{type}{template}{no.~of args}
  {key list}
```

- This step is to help designers/users: no code
- Keys will flexibility of this template
- Different templates for the same type can have different keys …
- …but in practice likely to see common ones

Template code

```
\DeclareTemplateImplementation{type}{template}{no. of args}
  {key bindings}{code}
```

- The business end: define the code
- Can be given separate from the interfaces in the sources

Using templates

- We can directly `\UseTemplate`

Using templates

- We can directly `\UseTemplate`
- But that means we always parsing the key list

Using templates

- We can directly `\UseTemplate`
- But that means we always parsing the key list
- Instead, most templates are used by making *instances*

Using templates

- We can directly `\UseTemplate`
- But that means we always parsing the key list
- Instead, most templates are used by making *instances*
- In an instance, the key list is parsed into a set of (fast) assignments

Using templates

- We can directly `\UseTemplate`
- But that means we always parsing the key list
- Instead, most templates are used by making *instances*
- In an instance, the key list is parsed into a set of (fast) assignments
- We can change one template/instance for another and it should still work

User control

- Document-wide changes can be made:
  edit instance defaults
- One-off changes also allowed using established key names
- Can also make copies of instances and edit these ‘child’ templates