

# Stacks

---



**Simon Robinson**

SOFTWARE DEVELOPER

@techiesimon [www.simonrobinson.com](http://www.simonrobinson.com)



# Overview



## Stack<T>

- Items (or tasks) to be processed
  - New items added to collection
  - Items removed as they are processed
- Undo is good example
- Stack<T> makes solving this easy



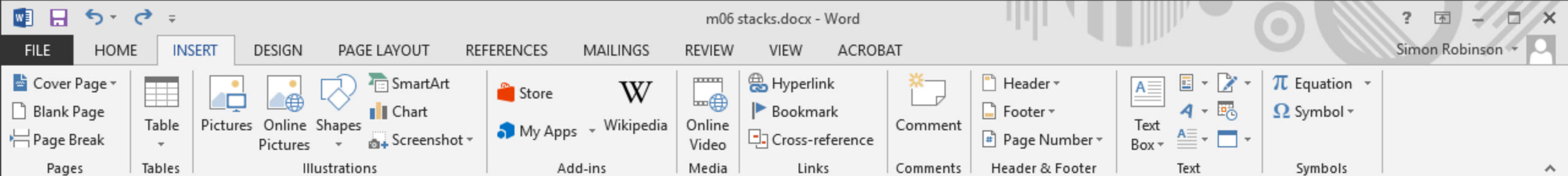
# Demo



## Undo operation

- What are the requirements?





how Stack<T> is perfectly suited to this kind of scenario, and **SLIDE** makes coding a solution much easier and more intuitive than would be the case the more general purpose collections, such as lists, dictionaries. **SLIDE**

I

### Introducing Stack<T>

**SLIDE** I just said that Stack<T> is perfectly **SLIDE** suited to an Undo situation. And I just want to start by thinking **SLIDE** about the requirements for an undo operation.

**WORD** I'm sure you're familiar with Undo from word-processing. When you are using for example Word, each change you make to the document is **MENU** logged so that after lots of edits you have a sequence of change logs. If you make a mistake and need to undo something, you hit the **CLICK** undo button and Word retrieves the most recent change from the log and reverses that change. And as it undoes that change, it'll remove that item from the log. And yes, this screenshot is self-referential and you have just entered an infinite loop from which there is no escape! I'm sorry...

**SLIDE** Anyway, notice that there are two expectations here.

The first is that the only item you ever want to look up and retrieve is the **SLIDE** one that was most recently added. It wouldn't usually make sense for example to **SLIDE** deliberately retrieve –

FILE

Typing ", "  
Typing  
Typing "t support index-based look-up."  
AutoFormat  
Typing "m hesitating because, as you know, linked li...  
AutoFormat  
Typing " want to call this the index, but I"  
Typing "the position in order that the country was at ...  
AutoFormat  
Typing "and "  
Typing " you added them."  
AutoCorrect  
Typing "t want the items to be returned in the revers...  
AutoFormat  
Typing "s useful if you don"  
Typing "i"  
Typing "s very similar to a stack but differs in that "  
Typing "i"  
Typing " and so on"  
Typing "ingle"  
Typing "s"  
Typing "dinglr "  
Typing "hat country"  
Typing "t"  
Typing "ts position"  
Typing "i"  
Typing "<T> just like Stack<T>"  
Typing "xtremely"  
Typing "e"  
Typing "ut first,"  
Typing "B"  
Typing "that "  
Typing "hat item"  
Typing "t"  
Typing "ost recently added"  
Typing "m"  
Typing " obviously"  
Typing  
Typing "is about collections, not "  
Typing  
Typing  
Typing "ade any edits"  
Typing "m"  
Typing "this code"  
Typing "h"  
Typing "o that"

REFERENCES MAILINGS REVIEW VIEW ACROBAT

Store

My Apps

Wikipedia

Online Video

Hyperlink

Bookmark

Cross-reference

Comment

Comments

Header

Footer

Page Number

Header & Footer

Equation

Symbol

Text Box

Text

Symbols

m06 stacks.docx - Word

Simon Robinson

fectly suited to this kind of scenario, and **SLIDE** makes coding a solution much  
itive than would be the case the more general purpose collections, such as lists,  
**SLIDE**

**Stack<T>**

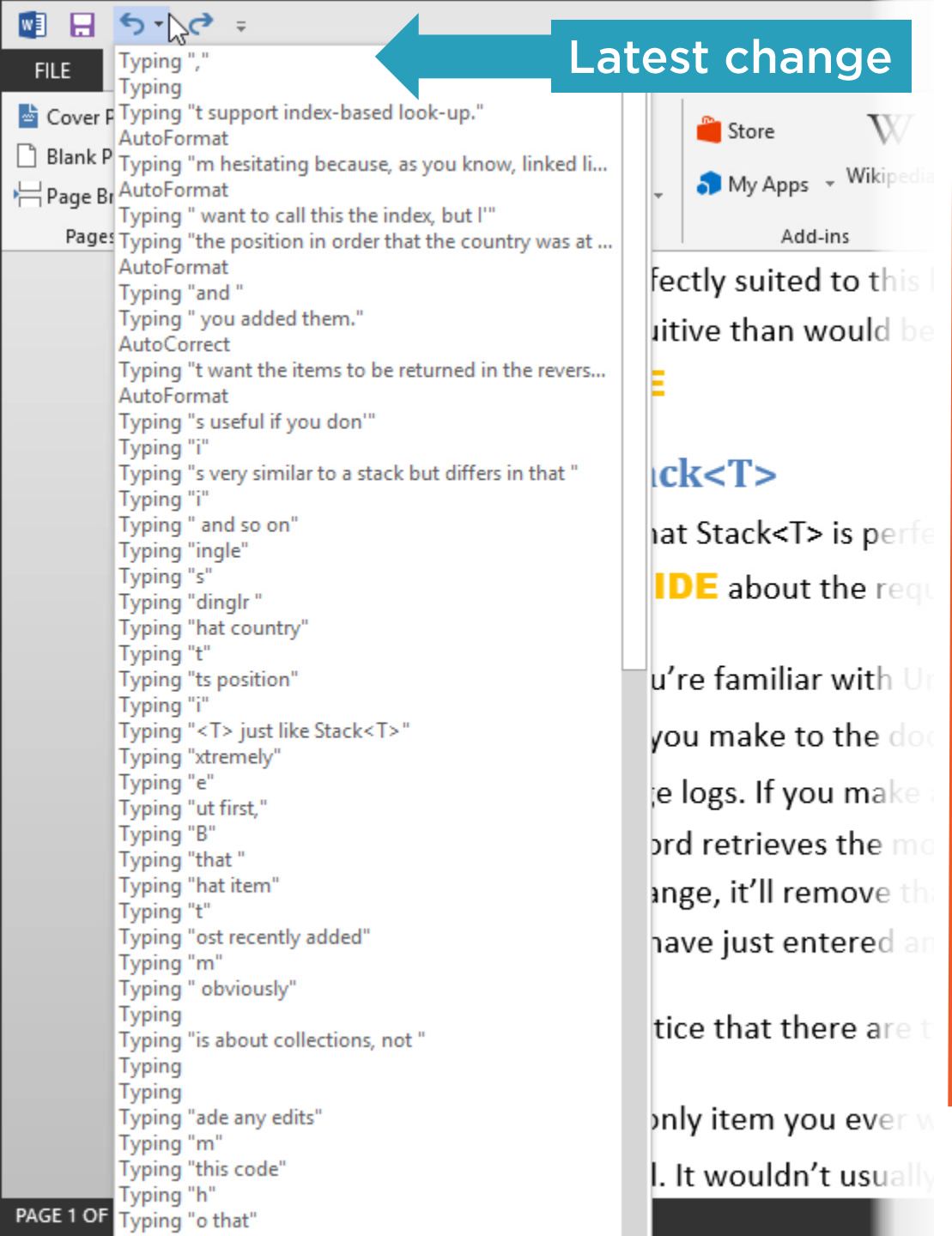
at Stack<T> is perfectly **SLIDE** suited to an Undo situation. And I just want to  
**SLIDE** about the requirements for an undo operation.

u're familiar with Undo from word-processing. When you are using for example  
you make to the document is **MENU** logged so that after lots of edits you have  
e logs. If you make a mistake and need to undo something, you hit the **CLICK**  
ord retrieves the most recent change from the log and reverses that change. And  
ange, it'll remove that item from the log. And yes, this screenshot is self-  
have just entered an infinite loop from which there is no escape! I'm sorry...

tice that there are two expectations here.

only item you ever want to look up and retrieve is the **SLIDE** one that was  
l. It wouldn't usually make sense for example to **SLIDE** deliberately retrieve –

PAGE 1 OF 1140%



# Undo Expectations

## You only retrieve the latest change

- Retrieving other changes makes no sense
- Processing a change removes it from the list of changes

## This is a stack





## Stack terminology

- ~~Add~~
- **Push** items onto a stack
- ~~Remove~~
- **Pop** items from a stack
- Last-in first-out collection

# Demo



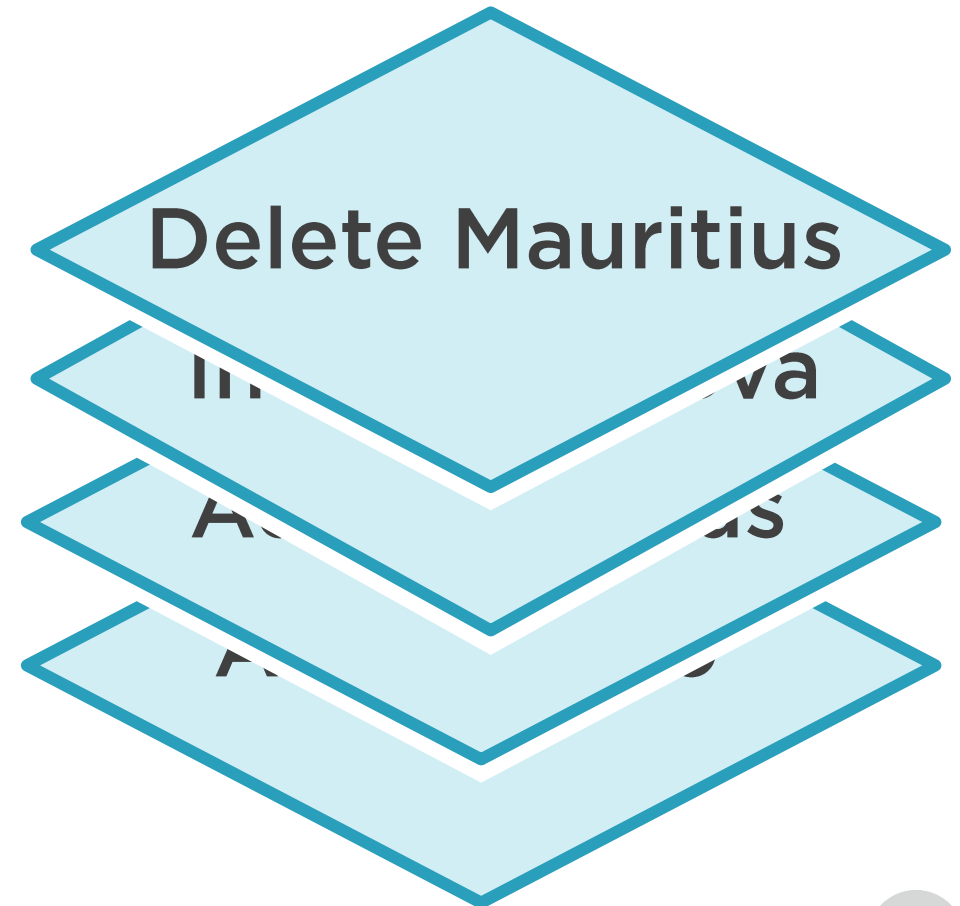
## TourBooker app:

- Undo changes when creating an itinerary





# Change Log



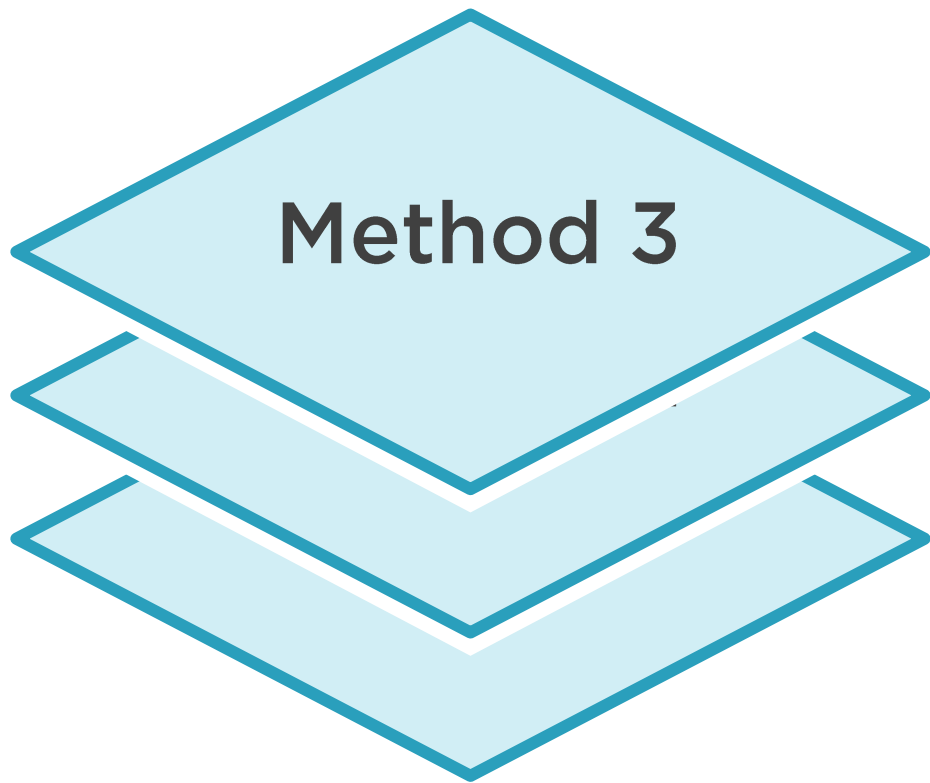


## Stack<T>

- Items stored in order (like List<T>)
- Retrieving an item removes it
- No direct element look-up
- The stack decides which item you get next
  - It's always the most recently added one



# Call Stack



Stack of successive method calls



# Summary



## Stacks

- For collections containing items to be processed
- Ordered
- No direct look-up
- Popping an item gives most recently added, and removes it
- Ideal for undo, or for call stacks

**Next up: Queues**

