



(the password is "spooky")



# Phantom Types

A spooky way to get more guarantees at compile time



# The humble datatype

```
datatype 'a tree = Empty  
                | Node of 'a tree * 'a * 'a tree
```

# The ~~humble~~ datatype

Parametrized/Polymorphic Type

`datatype 'a tree = Empty`

Generative

`| Node of 'a tree`

Sum Type

`* 'a *`

Product Type

`'a tree`

Recursive Type

# Parametric Polymorphism

Types have *type parameters*.

```
datatype 'a option = NONE | SOME of 'a
```

parameter  use of parameter

Values have *implicit type parameters*.

```
fun singleton (x : 'a) : 'a list = [x]
```

use of parameter  use of parameter

```
val l : 'a list = []
```

use of parameter

# Type Theory?



*"You're my type"*

# SML?

EXPLICIT  
(not valid SML)

```
fun 'a singleton (x : 'a) : 'a list = [x]  
  
(int singleton) (10)
```

IMPLICIT  
(valid SML)

```
fun singleton (x : 'a) : 'a list = [x]  
  
singleton (10)
```

# Multiple parameters

```
datatype ('a, 'b) either = Left of 'a  
                        | Right of 'b
```





# Type vs Datatype

`datatype` creates new types and requires constructors

```
datatype ('a, 'b) either = Left of 'a  
                        | Right of 'b
```

`type` renames an already existing type

```
type 'a or_error = ('a, string) either
```

# Phantom Types

Useless type parameters!



# Phantom Types

Datatype or type declarations where the type parameter *is never used*

```
datatype `a index = Idx of int  
                | OutOfBounds of int
```

```
type `a integer = int
```

Why would you  
ever  
want that?

For lots of things!



# To statically...

- Keep track of where data came from or where it's going
- Enforce privacy or data sanitization policies
- Keep track of data structure properties
- Control access to operations on data structures

# Enforce Sanitization

```
{ firstName: "<script src=\"sketchysite.com\"></script>Mickey",  
  lastName: "Mouse" }
```



```
{ firstName: "Mickey",  
  lastName: "Mouse" }
```

User Input



Sanitize



Display on Website

First name:

Last name:

Your input was received as:

# Access Control

Mutable and immutable versions of data structure

