

# 2013W1-lecture15

October 20, 2013

## Contents

<b>1</b>	<b>Question of the Day</b>	<b>1</b>
<b>2</b>	<b>Logistics</b>	<b>2</b>
2.1	Programming Assignment #3 . . . . .	2
2.1.1	Demos for PA3 . . . . .	2
2.2	Neat PL research talk upcoming: 24 Oct, 3:30-4:30PM, X836	2
<b>3</b>	<b>Finishing Yesterday!</b>	<b>3</b>
<b>4</b>	<b>What have we learned today?</b>	<b>3</b>

## 1 Question of the Day

Once upon a time, in a spammy language called Python, programmers who wanted to prepend a string to all strings in a list might have to write odd code like this:

```
>>> def prepend(prefix, sequence):
...     return map(lambda s, prefix=prefix: prefix + s, sequence)
...
>>> prepend("holy ", ["hand grenade", "capilano suspension bridge"])
['holy hand grenade', 'holy capilano suspension bridge']
```

Note: `>>` and `...` are just Python prompts.  
Now, they can breathe a sigh of relief and write:

```
>>> def prepend(prefix, sequence):
...     return map(lambda s: prefix + s, sequence)
...
>>> prepend("holy ", ["hand grenade", "capilano suspension bridge"])
['holy hand grenade', 'holy capilano suspension bridge']
```

What changed? Why?

SOLUTION

Of course, Python programmers wouldn't write that at all. They'd use list comprehensions, which really do have little lambdas inside, secretly:

```
:     return [prefix + s for s in sequence]
```

## 2 Logistics

### 2.1 Programming Assignment #3

Final submission due Friday

#### 2.1.1 Demos for PA3

### 2.2 Neat PL research talk upcoming: 24 Oct, 3:30-4:30PM, X836

One bonus point for attending and posting a brief summary of the talk from the CPSC 311 perspective.

Two bonus points for attending and posting a thoughtful, thorough discussion of the talk.

Three bonus points for being the speaker. :)

TypeScript is a programming language whose goal is to support development of large JavaScript programs. TypeScript is a superset of the current JavaScript standard (ECMAScript 5) that adds an optional static type system to JavaScript. TypeScript exists only to support high-level thinking about JavaScript programs; it has no impact on runtime behavior. Because of this, TypeScript is an example of “types for tooling” vs. the more traditional idea of “types for runtime safety.” TypeScript has a novel design for type inference; the goal of the design is to provide maximum convenience (few annotations required) and transparency (chains of inference are clear and local). The TypeScript compiler, incremental static analysis tools,

and specification are open source (see [typescriptlang.org](http://typescriptlang.org)). Several million lines of TypeScript are part of shipping Microsoft products. Since the community preview release in October, 2012, several 100K+ line TypeScript projects have grown up outside of Microsoft and the TypeScript community has created a site, at [github.com](http://github.com), that holds over 100 community-maintained TypeScript descriptions of popular JavaScript frameworks such as jQuery.

Steve Lucco is a Technical Fellow at Microsoft, where he is responsible for Microsoft's web development tools and runtimes. He led the development of Microsoft's Chakra JavaScript engine, which powers Internet Explorer. Currently, Chakra is 30% faster than Chrome V8 on SunSpider, the most widely cited JavaScript benchmark. He started the TypeScript team and contributes to the design and implementation of TypeScript.

### 3 Finishing Yesterday!

### 4 What have we learned today?

- From QotD: The terrible things that happen when language designers do static scoping wrong, and how programmers fix their mistakes.

```
ORGMODECONFIG
```

```
#+DRAWERS: SOLUTION ORGMODECONFIG
```

```
#+COMMENT TODO: change to d:nil (or delete) to not export SOLUTION drawers
```

```
#+OPTIONS: d:t
```