New Beamer Presentation

Drew Conway

January 22, 2010

Introduction

I usually have the title and introduction slides in the `plain` format, and then the rest in `fragile`

- I also use a lot of itemized lists

Sectioning and columns

I also like to use sections and subsections to call out where in the talk I am.

Columns are also great for side-by-side comparisons

So I use those a lot too

Template for code example

I like to use code blocks to call out examples, and teh `alert` and `uncover` functions In Python the dict type is a data structure that represents a key→value mapping

Template for code example

I like to use code blocks to call out examples, and teh `alert` and `uncover` functions In Python the dict type is a data structure that represents a key→value mapping

Working with the dict type

```
# Keys and values can be of any data type
>>> fruit_dict={"apple":1,"orange":[0.23,0.11],"banana":True }
```

Template for code example

I like to use code blocks to call out examples, and teh `alert` and `uncover` functions In Python the dict type is a data structure that represents a key→value mapping

Working with the dict type

```
# Keys and values can be of any data type
>>> fruit_dict={"apple":1,"orange":[0.23,0.11],"banana":True }

# Can retrieve the keys and values as Python lists (vector)
>>> fruit_dict.keys()
["orange","apple","banana"]
```

Template for code example

I like to use code blocks to call out examples, and teh `alert` and `uncover` functions In Python the dict type is a data structure that represents a key→value mapping

Working with the dict type

```
# Keys and values can be of any data type
>>> fruit_dict={"apple":1,"orange":[0.23,0.11],"banana":True }

# Can retrieve the keys and values as Python lists (vector)
>>> fruit_dict.keys()
["orange","apple","banana"]

# Or create a (key,value) tuple
>>> fruit_dict.items()
[("orange",[0.23,0.11]),("apple",1),("Banana",True)]
```

Template for code example

I like to use code blocks to call out examples, and teh `alert` and `uncover` functions In Python the dict type is a data structure that represents a key→value mapping

Working with the dict type

```
# Keys and values can be of any data type
>>> fruit_dict={"apple":1,"orange":[0.23,0.11],"banana":True }

# Can retrieve the keys and values as Python lists (vector)
>>> fruit_dict.keys()
["orange","apple","banana"]

# Or create a (key,value) tuple
>>> fruit_dict.items()
[("orange",[0.23,0.11]),("apple",1),("Banana",True)]
# This becomes especially useful when you master Python ‘‘list comprehension’’
```

Template for code example

I like to use code blocks to call out examples, and teh `alert` and `uncover` functions In Python the dict type is a data structure that represents a key→value mapping

Working with the dict type

```
# Keys and values can be of any data type
>>> fruit_dict={"apple":1,"orange":[0.23,0.11],"banana":True }

# Can retrieve the keys and values as Python lists (vector)
>>> fruit_dict.keys()
["orange","apple","banana"]

# Or create a (key,value) tuple
>>> fruit_dict.items()
[("orange",[0.23,0.11]),("apple",1),("Banana",True)]
# This becomes especially useful when you master Python ‘‘list comprehension’’
```

The Python dictionary is an extremely flexible and useful data structure, making it one of the primary advantages of Python over other languages