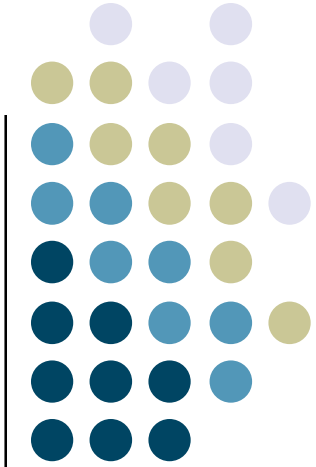
Python Tips & Review

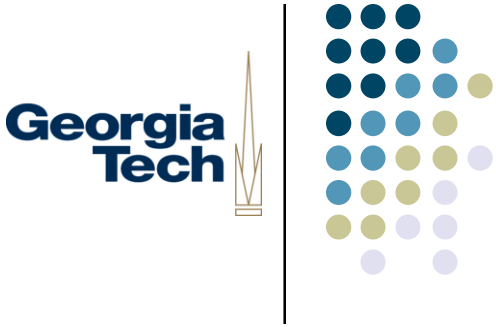


John Stasko

CS 6452

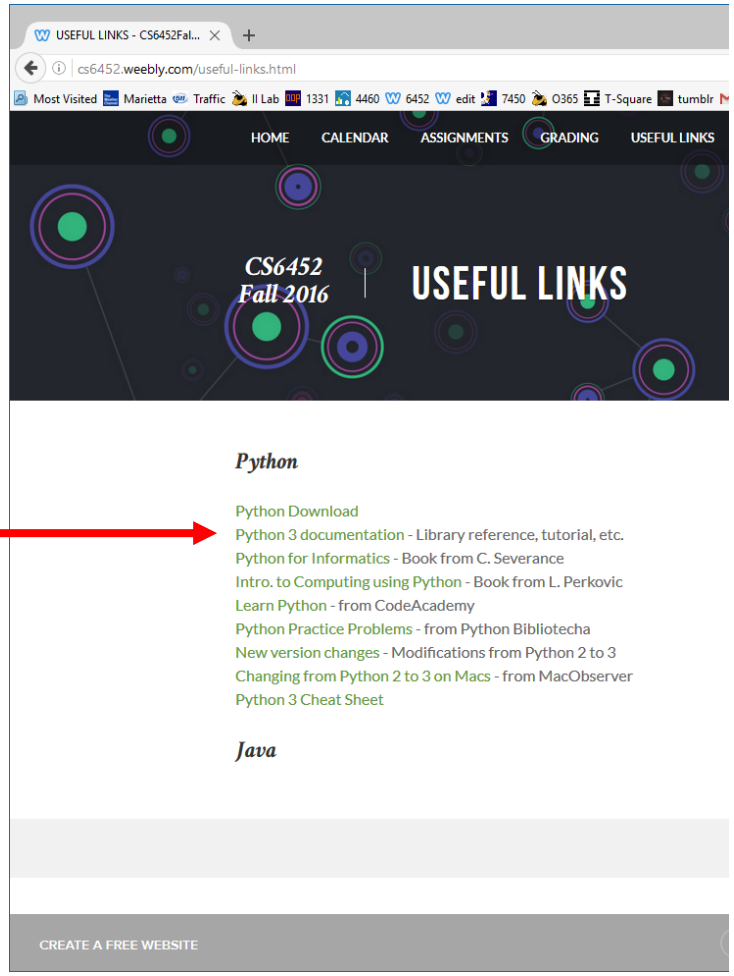
Prototyping Interactive Systems

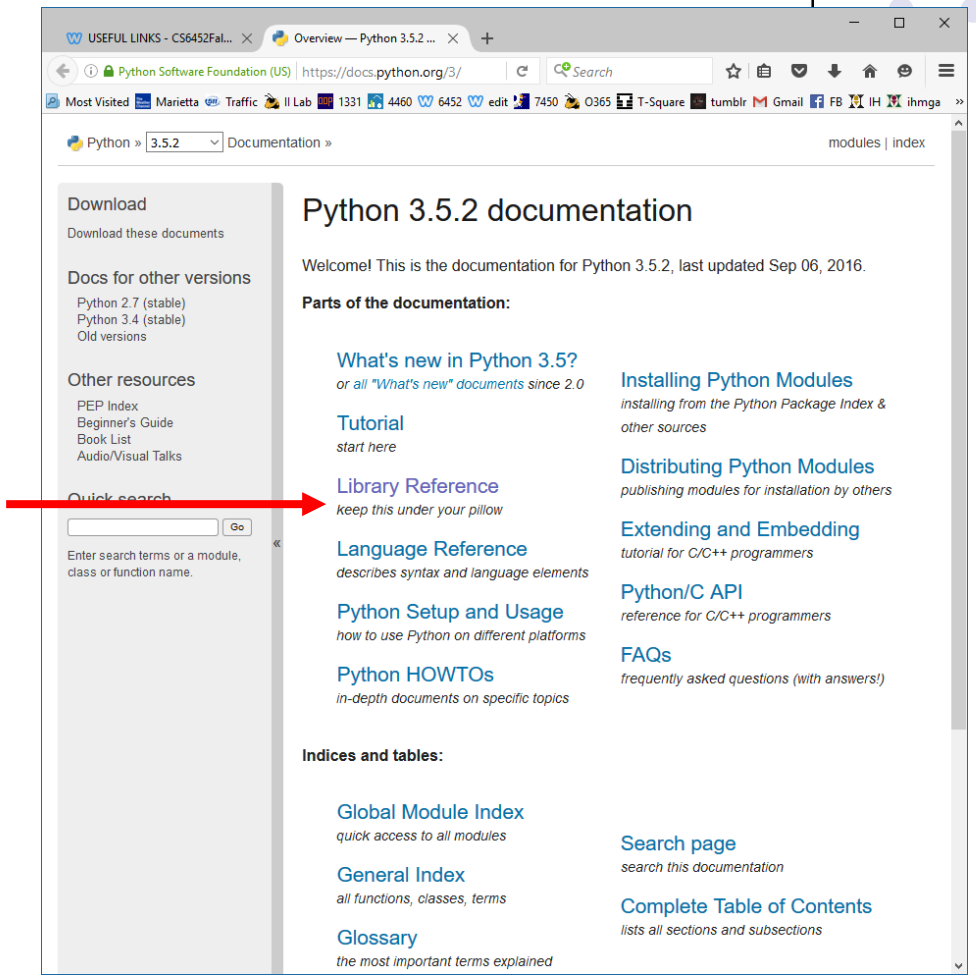
Learning Objectives



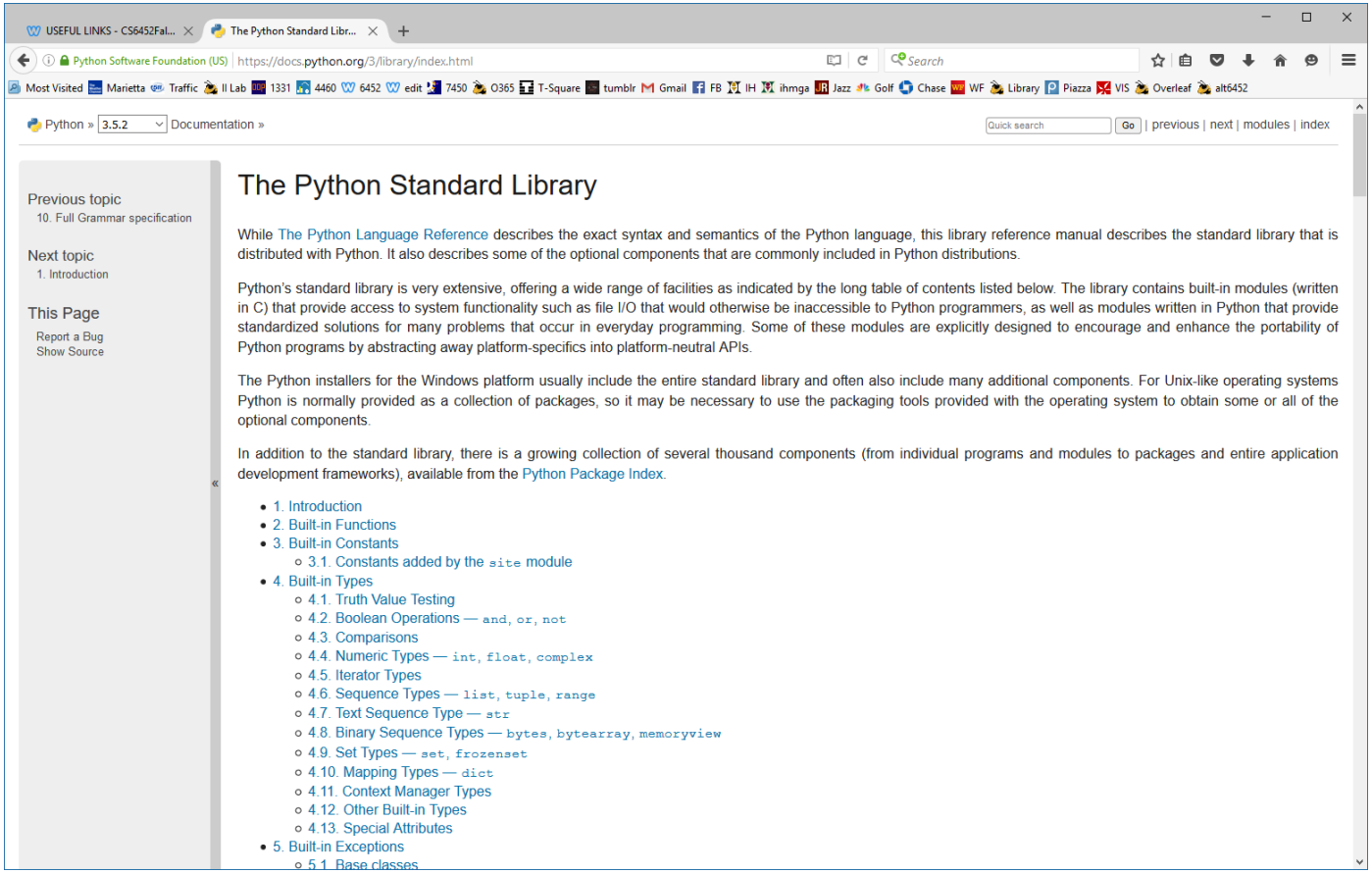
- Review some ideas
- See a new one or two
- Get some help on the HW

Documentation

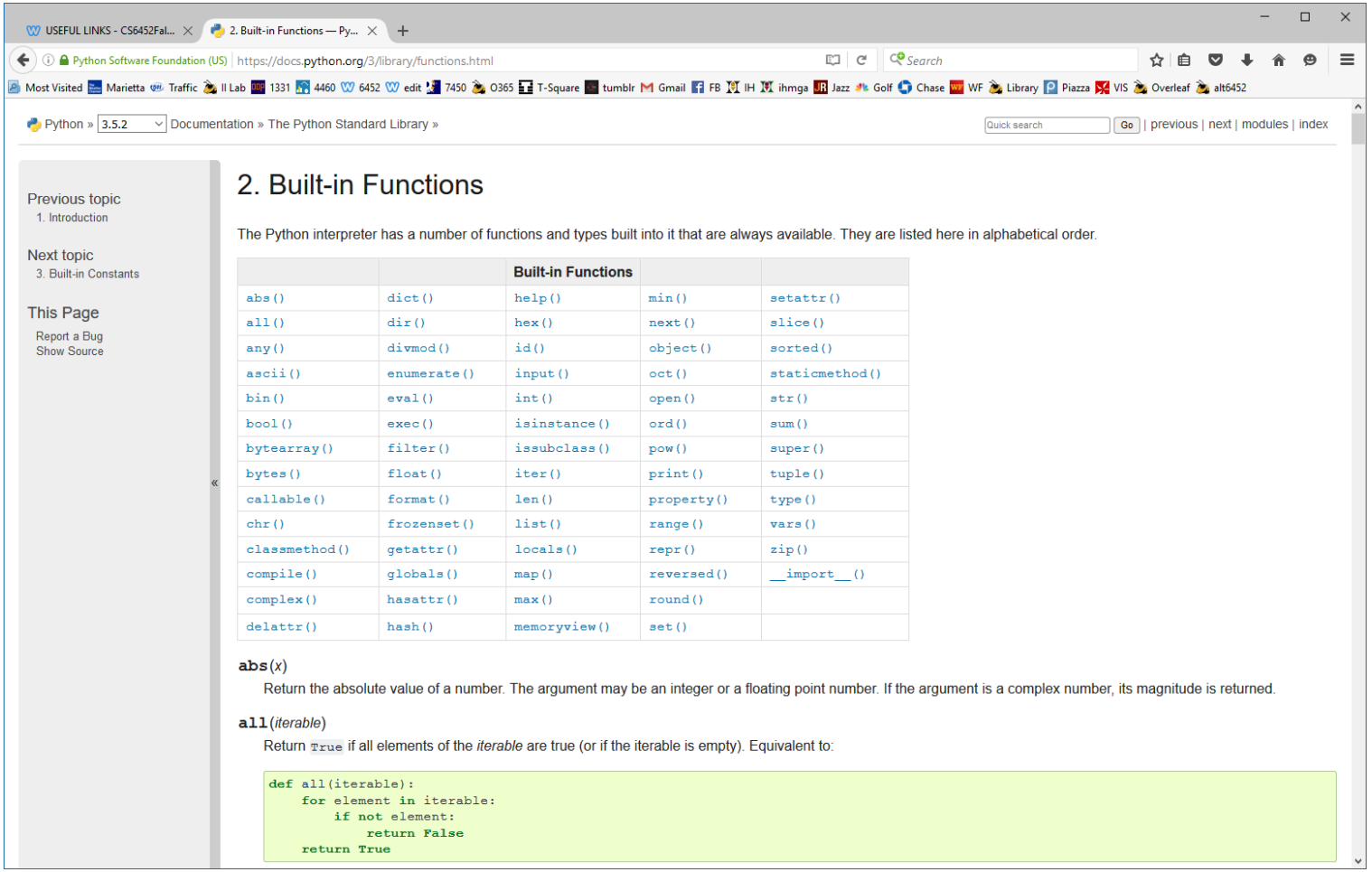




Standard Library



Built-in Functions



Python » 3.5.2 Documentation » The Python Standard Library »

Quick search | Go | previous | next | modules | index

Previous topic
1. Introduction

Next topic
3. Built-in Constants

This Page
Report a Bug
Show Source

2. Built-in Functions

The Python interpreter has a number of functions and types built into it that are always available. They are listed here in alphabetical order.

		Built-in Functions		
abs()	dict()	help()	min()	setattr()
all()	dir()	hex()	next()	slice()
any()	divmod()	id()	object()	sorted()
ascii()	enumerate()	input()	oct()	staticmethod()
bin()	eval()	int()	open()	str()
bool()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	
delattr()	hash()	memoryview()	set()	

abs(*x*)

Return the absolute value of a number. The argument may be an integer or a floating point number. If the argument is a complex number, its magnitude is returned.

all(*iterable*)

Return `True` if all elements of the *iterable* are true (or if the iterable is empty). Equivalent to:

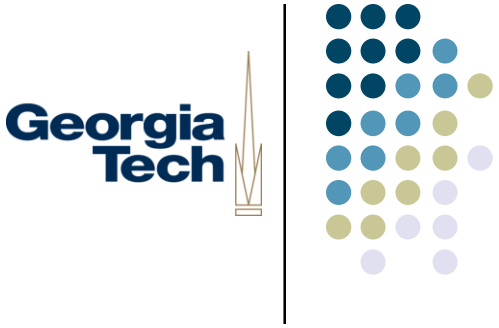
```
def all(iterable):
    for element in iterable:
        if not element:
            return False
    return True
```

Design Advice

- Break program into set of "things" that need to be done
- Make each of those things be a function
- HW?

- Challenge?

Importing Data



- How would you do it?
- Recall

readCSVdictionary.py

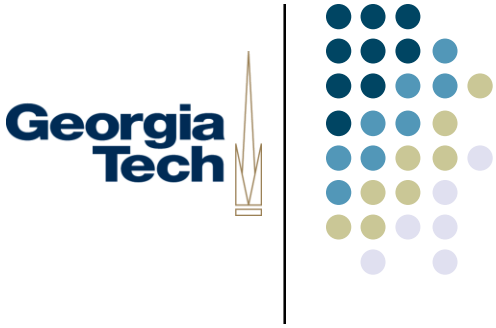
```
import csv

reader = csv.DictReader(open("students.csv"))
# puts headers in reader.fieldnames

for row in reader:
    # do work
```

What is in "reader" now?

Back to that Challenge

- What do we have to do with "reader"?

Data Structure



```
[
    { "Name":"Glenmorangie", "Rating":"94", "Age":"15", …}
    { "Name":"Balvenie", "Rating":"87", "Age":"6", …}
    { "Name":"Woodford", "Rating":"90", "Age":"12", …}
    { "Name":"Glenfiddich", "Rating":"98", "Age":"35", …}
    …
]
```

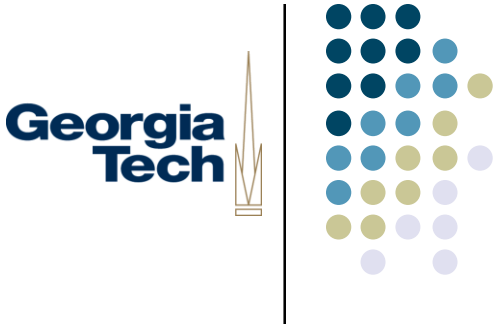
Match by Name

- How?
  - Walk list
  - Check name of each one
  - If match, you got it

How to check name of each one?

```
if target_name == row["Name"]:
```

Matching Prices

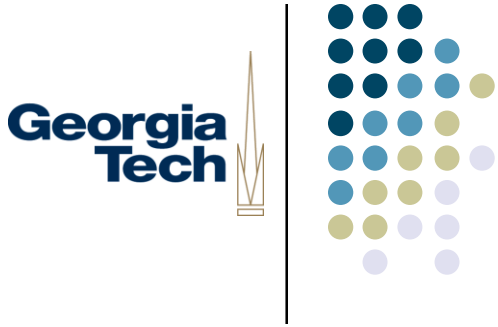


- How?
- Walk list, check each price in range
- Then what? (sorted by rating)
- It's OK to create other useful data structures

Formula

- Need to calculate formula for each whiskey
  - What if a bad rating or price?
- Next steps
  - Sort all "valid" whiskeys by factor
  - Produce the top 10

Useful Function



sorted(*iterable[, key][, reverse]*)

Return a new sorted list from the items in *iterable*.

Has two optional arguments which must be specified as keyword arguments.

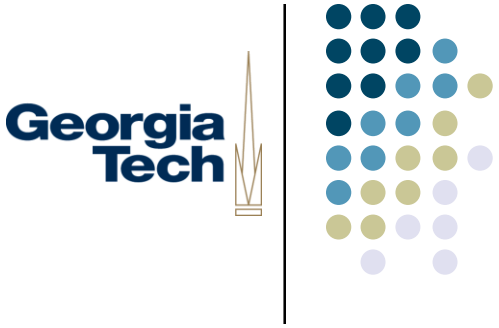
key specifies a function of one argument that is used to extract a comparison key from each list element: `key=str.lower`. The default value is `None` (compare the elements directly).

reverse is a boolean value. If set to `True`, then the list elements are sorted as if each comparison were reversed.

Use `functools.cmp_to_key()` to convert an old-style *cmp* function to a *key* function.

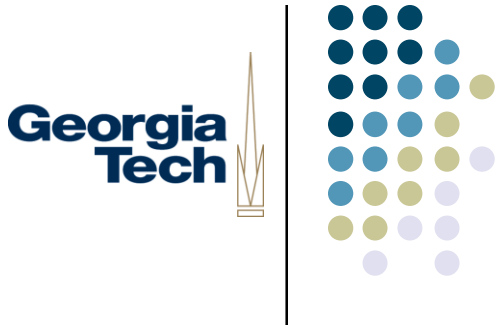
The built-in `sorted()` function is guaranteed to be stable. A sort is stable if it guarantees not to change the relative order of elements that compare equal — this is helpful for sorting in multiple passes (for example, sort by department, then by salary grade).

For sorting examples and a brief sorting tutorial, see Sorting HOW TO.



```
>> a = [5, 8, 3]
>> b = sorted(a)
>> print(b)
[3, 5, 8]
```

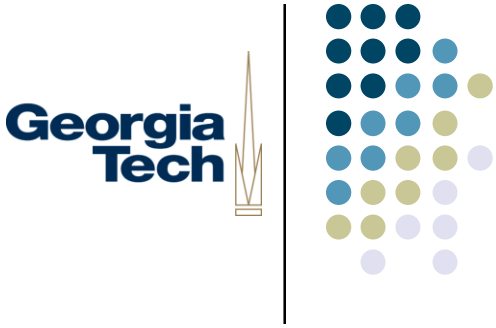
What's our challenge?



```
def getFactor(obj):
    return obj["FactorValue"]
```

```
sorted_list = sorted(factor_list, key=getFactor, reverse=True)
```

Learning Objectives



- Review some ideas
- See a new one or two
- Get some help on the HW

HW 2

- Due Thursday at class time
- Ask questions in Piazza
- Sakshi and I have office hours tomorrow

Next Time

- Accessing data files from the web
- Using web APIs