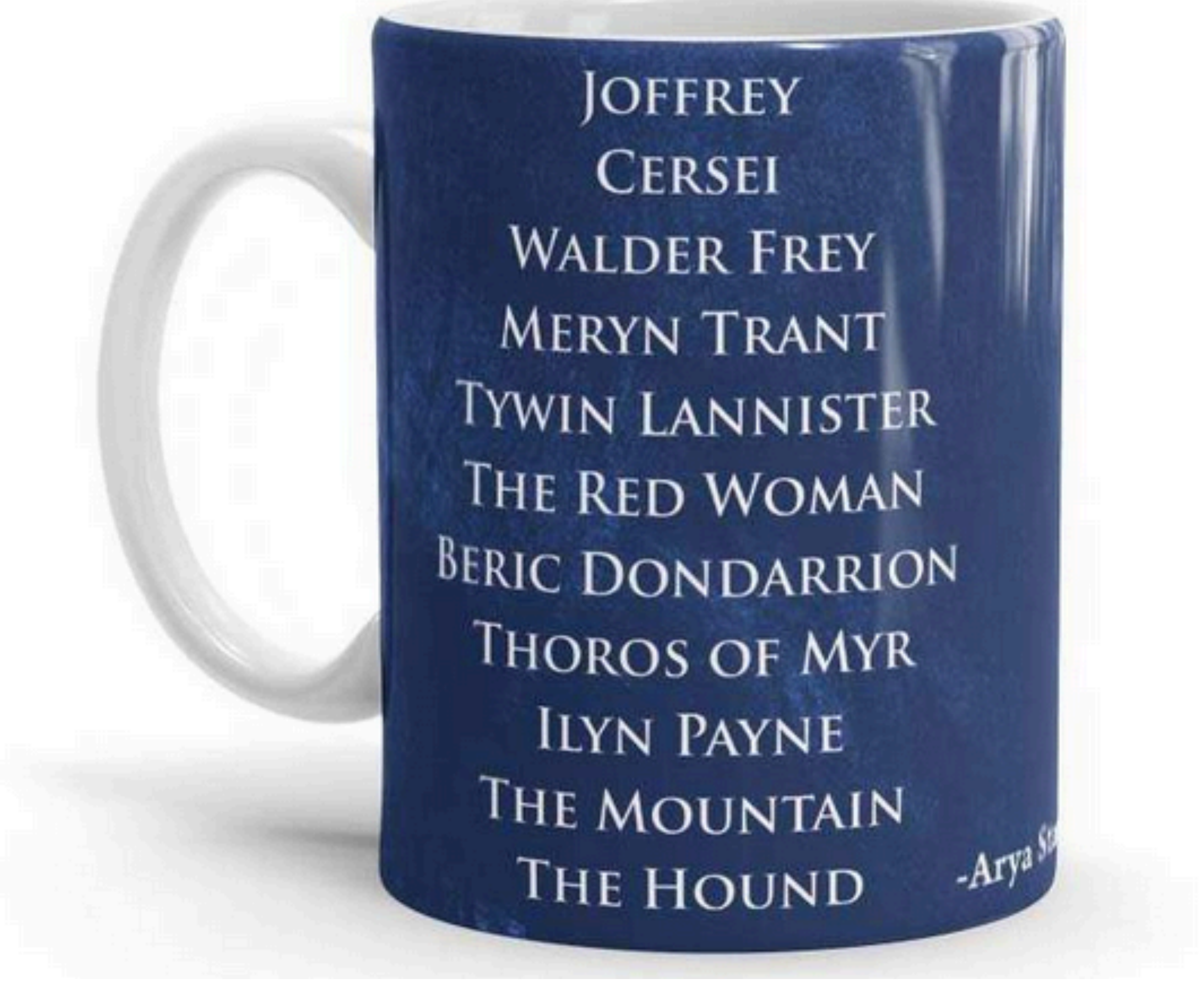


CSCI 141

Scott Wehrwein

List Basics

Goals

- Know how to create, index, slice, and check for membership in lists.
- Understand the behavior of the `+, *, in, not in,` operators on lists.

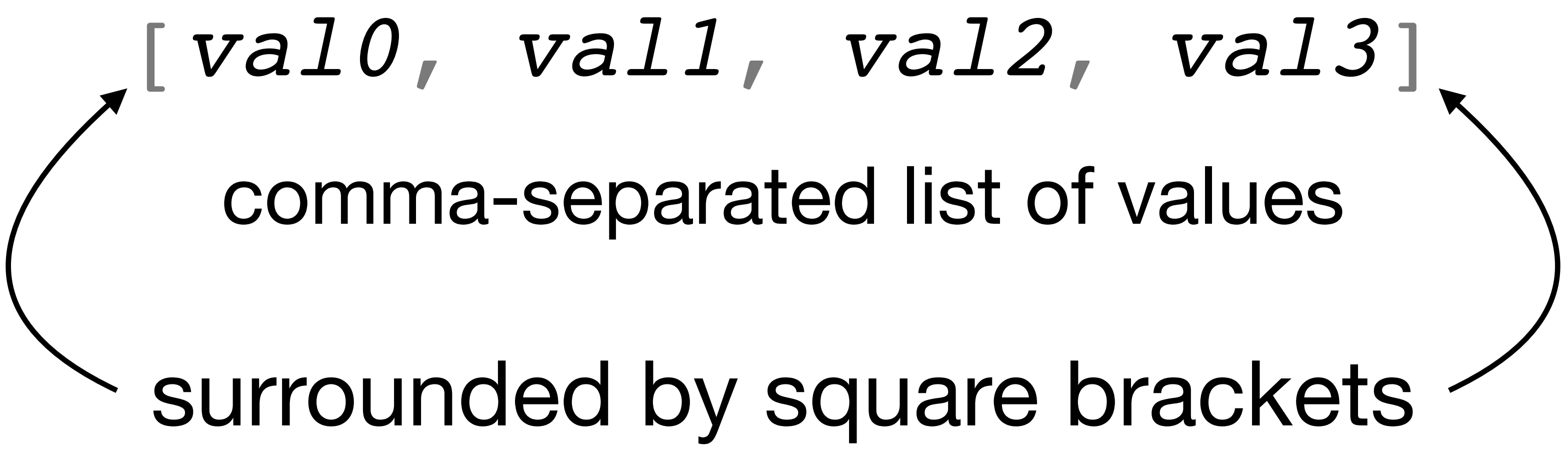
Lists: Yet Another Sequence Type

A list is an object that contains a sequence of values.

We've seen them before.

```
for value in [1, 16, 4]:
    print(value)
```

Syntax:



```
[val0, val1, val2, val3]
```

comma-separated list of values

surrounded by square brackets

The values (elements) can be of any type(s)!

What can we do with Lists?

A lot of this should look familiar.

```
a_list = ["Scott", 34, 27.7]
```

These things work analogously to strings:

- Indexing
- Slicing
- The len function
- in and not in operators
- + and * operators

What can go in lists?

- Like tuples, *any* value can go in a list.
    - int, str, tuple, list, Turtle, ... *any value*

Demo

A lot of this should look familiar.

Demo

A lot of this should look familiar.

`a_list = ["Scott", 34, 27.7]`	make 'em
`a_list[0]`	index 'em
`a_list[-1]`	index 'em
`a_list[1:]`	slice 'em

Demo

```python
a_list = ["Scott", 34, 27.7]
len(a_list)
len(["abc"])
len([])
34 in a_list
"34" not in a_list
a_list + ["Wehrwein", "WWU"]
["na"] * 16 + ["Batman"]
a_list[0:2]
```

Demo

Lists can contain any type: lists, tuples, turtles, ...

```
a_list = ["Scott", [34, 27.7, (39, 70)]]

a_list[0]

a_list[1]

a_list[1][2]

a_list[1][2][0]
```