Pykidz!

A 'Kideveloper' initiative

I do, I learn!

Composite Types

- Lists - We covered this last time
- Tuples - Now we cover this first
- Maps/Dict
- Sets

Tuples

- List with () ?

Tuples vs. Lists

- Lists are ordered and so are Tuples
- Lists can contain arbitrary objects (including nested items) and so can Tuples
- List elements can be accessed by index and so can Tuples
- Lists are dynamic & mutable but Tuples are not

Tuples operations

- Indexing and Slicing
- Membership ('in' and 'not in')
- Truthy/Falsy

Tuples are immutable

- Let us id them
- Use when you don't want change

Tuples

- Defining gotchas

Tuples

- Assignment, packing & unpacking

Composite Types

- Lists - We covered this last time
- Tuples - Ok, that is done
- Maps/Dicts - Next this
- Sets

Dicts

- Collection of key & value

Dicts

- Define them

Dicts

- Access by key
- What key?
  - Anything immutable

Dicts

- Change a dict

Dicts

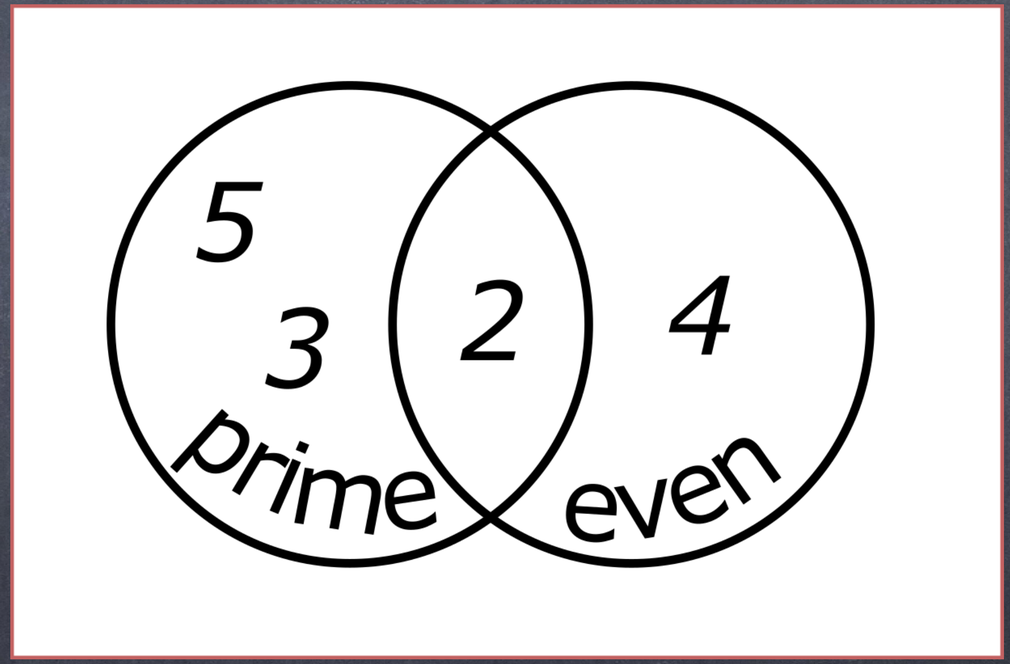
- Membership and methods

Composite Types

- Lists - We covered this last time
- Tuples - Ok, that is done
- Maps/Dicts - Done this
- Sets - The last one

Sets - Definition

- A set can be thought of simply as a well-defined collection of distinct objects, typically called items
- Distinct and unordered



Sets - Creation and change

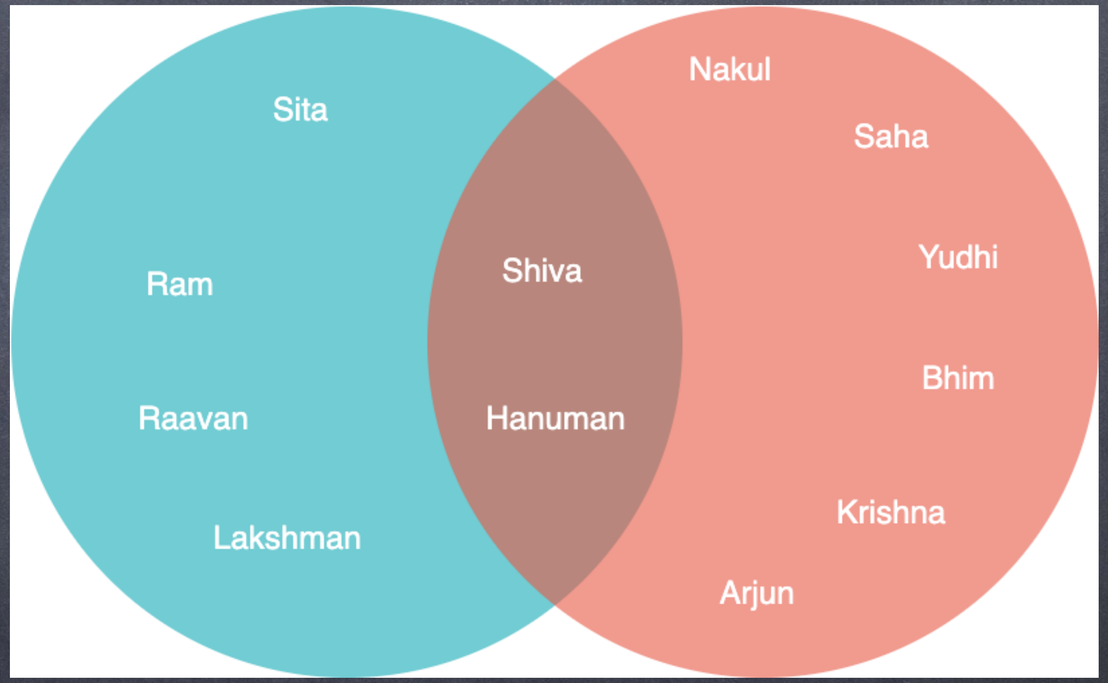
- What dan be in it?
- Change a set?

Sets

- Some properties

Sets: The Super Ops

- | or union
- & or intersection
- <= or issubset
- >= or issuperset
- |= or update



Sets - element methods

- add
- remove
- discard
- clear

Composite Types

- Lists - Done Earlier!
- Tuples - Ok, Done!!
- Maps/Dicts - Ok, Done!!!
- Sets - Ok, Done!!!!

We are done!
I do, I learn!