

COMP
110

Recursive Structures

Recursion

- When a process or structure is defined in terms of itself
- Examples:
 - File System
 - Tree Branches

Recursion

function data type

- When a `process` or `structure` is defined in terms of itself
- Examples:
 - File System
 - Tree Branches

NOTE:

Like loops, recursion
needs to terminate!

Recursion

function data type

- When a **process** or **structure** is defined in terms of itself
- Examples:
 - File System
 - Tree Branches
- Made of:
 - Base case
 - Recursive Step

Structures

Recursive Data Type:

- Contains itself as an attribute/part of an attribute

```
class Node:  
    data: int  
    next: Node
```

Linked Lists vs Lists

```
class Node:  
    data: int  
    next: Node
```

Recursive Structures vs. Functions

```
class Node:  
    data: int  
    next: Node
```

```
13 ∨ def f(x: int) -> int:  
14 ∨ |     if x == 0:  
15   | |         return x  
16 ∨ |     else:  
17   | |         return 1 + f(x-1)
```

VS Example...

Ending Recursion

- Terminates on a **base case**
- Recursive attribute replaced with **None** type



In Memory

```
6  ✓ class Node:
7      |     """My Node class for linked lists."""
8      |
9      |     data: int
10     |     next: Node | None
11     |
12  ✓ def __init__(self, data: int, next: Node | None):
13     |     """Construct Node."""
14     |     self.data = data
15     |     self.next = next
```

```
1  node_c = Node(2, None)
2  node_b = Node(1, node_c)
3  node_a = Node(0, node_b) #head of list
```

