61A Lecture 17
Announcements
Linked Lists
Linked List Structure

A linked list is either empty or a first value and the rest of the linked list
Linked List Structure

A linked list is either empty or a first value and the rest of the linked list

\[ 3, 4, 5 \]
A linked list is either empty or a first value and the rest of the linked list

\[3, 4, 5\]
Linked List Structure

A linked list is either empty or a first value and the rest of the linked list.

\[3, 4, 5\]

**Link instance**

<table>
<thead>
<tr>
<th>first:</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>rest:</td>
<td></td>
</tr>
</tbody>
</table>

**Link instance**

<table>
<thead>
<tr>
<th>first:</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>rest:</td>
<td></td>
</tr>
</tbody>
</table>
A linked list is either empty or a first value and the rest of the linked list.
Linked List Structure

A linked list is either empty or a first value and the rest of the linked list.

3, 4, 5
A linked list is either empty or a first value and the rest of the linked list.
Linked List Structure

A linked list is either empty or a first value and the rest of the linked list.

A linked list is a pair.

The first (zeroth) element is an attribute value.

3, 4, 5
**Linked List Structure**

A linked list is either empty or a first value and the rest of the linked list.

A linked list is a pair

The first (zeroth) element is an attribute value

The rest of the elements are stored in a linked list

3, 4, 5
A linked list is either empty or a first value and the rest of the linked list.

A class attribute represents an empty linked list.

The first (zeroth) element is an attribute value.

The rest of the elements are stored in a linked list.
A linked list is either empty or a first value and the rest of the linked list.

A class attribute represents an empty linked list.

The first (zeroth) element is an attribute value.

The rest of the elements are stored in a linked list.

Link(3, Link(4, Link(5, Link.empty)))
Linked List Structure

A linked list is either empty or a first value and the rest of the linked list.

```
3, 4, 5
```

```
Link instance
<table>
<thead>
<tr>
<th>first:</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>rest:</td>
<td></td>
</tr>
</tbody>
</table>

Link instance
<table>
<thead>
<tr>
<th>first:</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>rest:</td>
<td></td>
</tr>
</tbody>
</table>

Link instance
<table>
<thead>
<tr>
<th>first:</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>rest:</td>
<td></td>
</tr>
</tbody>
</table>

Link.empty
```
A linked list is either empty or a first value and the rest of the linked list.

\[ \text{Link(3, Link(4, Link(5, Link.empty)))} \]
Linked List Structure

A linked list is either empty or a first value and the rest of the linked list.

\[
\text{Link instance} \quad \begin{array}{|c|}
\hline
\text{first:} & 3 \\
\hline
\text{rest:} & \text{Link instance} \\
\hline
\end{array} \quad \begin{array}{|c|}
\hline
\text{first:} & 4 \\
\hline
\text{rest:} & \text{Link instance} \\
\hline
\end{array} \quad \begin{array}{|c|}
\hline
\text{first:} & 5 \\
\hline
\text{rest:} & \text{Link instance} \\
\hline
\end{array} \quad \text{Link.empty}
\]

\text{Link(3, Link(4, Link(5, Link.empty)))}
Linked List Structure

A linked list is either empty or a first value and the rest of the linked list.

```
Link instance
first: 3
rest:  

Link instance
first: 4
rest:  

Link instance
first: 5
rest:  

Link.empty
```

\( \text{Link}(3, \text{Link}(4, \text{Link}(5, \text{Link.empty}))) \)
A linked list is either empty or a first value and the rest of the linked list.

\[ \text{Link} \text{ instance} \begin{array}{|c|} \hline \text{first:} & 3 \\ \text{rest:} & \text{Link} \text{ instance} \\ \hline \end{array} \begin{array}{|c|} \hline \text{first:} & 4 \\ \text{rest:} & \text{Link} \text{ instance} \\ \hline \end{array} \begin{array}{|c|} \hline \text{first:} & 5 \\ \text{rest:} & \text{Link} \text{ instance} \\ \hline \end{array} \]

\[ \text{Link}(3, \text{Link}(4, \text{Link}(5, \text{Link}.\text{empty}))) \]
Linked List Structure

A linked list is either empty or a first value and the rest of the linked list.

\[ 3, 4, 5 \]

```
Link instance
first:   3
rest:    

Link instance
first:   4
rest:    

Link instance
first:   5
rest:    

Link(3, Link(4, Link(5)))
```
Linked List Class

\[ \text{Link}(3, \text{Link}(4, \text{Link}(5))) \]
Linked List Class

Linked list class: attributes are passed to \texttt{__init__}

\begin{verbatim}
Link(3, Link(4, Link(5)))
\end{verbatim}
Linked List Class

Linked list class: attributes are passed to `__init__`

```python
class Link:

    Link(3, Link(4, Link(5)))
```


Linked List Class

Linked list class: attributes are passed to __init__

```python
class Link:

    def __init__(self, first, rest=empty):
        Link(3, Link(4, Link(5)))
```

Linked List Class

Linked list class: attributes are passed to \_\_init\_\_

```python
class Link:
    def \_\_init\_(self, first, rest=empty):
        assert rest is Link.empty or isinstance(rest, Link)

Link(3, Link(4, Link(5)))
```
Linked List Class

Linked list class: attributes are passed to \texttt{\_\_init\_}

```python
class Link:

def \_\_init\_(self, first, rest=empty):
    assert rest is Link.empty or isinstance(rest, Link)
    self.first = first
    self.rest = rest
```

```
Link(3, Link(4, Link(5)))
```
Linked List Class

Linked list class: attributes are passed to __init__

class Link:

def __init__(self, first, rest=empty):
    assert rest is Link.empty or isinstance(rest, Link)
    self.first = first
    self.rest = rest

    Link(3, Link(4, Link(5)))
Linked List Class

Linked list class: attributes are passed to __init__

class Link:

def __init__(self, first, rest=empty):
    assert rest is Link.empty or isinstance(rest, Link)
    self.first = first
    self.rest = rest

help(isinstance): Return whether an object is an instance of a class or of a subclass thereof.

    Link(3, Link(4, Link(5)))
Linked List Class

Linked list class: attributes are passed to __init__

class Link:

    empty = ()

def __init__(self, first, rest=empty):
    assert rest is Link.empty or isinstance(rest, Link)
    self.first = first
    self.rest = rest

help(isinstance): Return whether an object is an instance of a class or of a subclass thereof.

    Link(3, Link(4, Link(5)))
Linked List Class

Linked list class: attributes are passed to __init__

class Link:

    empty = ()

    def __init__(self, first, rest=empty):
        assert rest is Link.empty or isinstance(rest, Link)
        self.first = first
        self.rest = rest

    (3, (4, (5, )))

help(isinstance): Return whether an object is an instance of a class or of a subclass thereof.
Linked List Class

Linked list class: attributes are passed to __init__

class Link:
    empty = ()

def __init__(self, first, rest=empty):
    assert rest is Link.empty or isinstance(rest, Link)
    self.first = first
    self.rest = rest

Demo:

    Link(3, Link(4, Link(5)))

help(isinstance): Return whether an object is an instance of a class or of a subclass thereof.
Sequence Operations
Linked List Class

More special method names:

__getitem__  Element selection []
__len__      Built-in len function
Linked List Class

Linked lists are sequences

More special method names:

__getitem__ Element selection []
__len__ Built-in len function
Linked List Class

Linked lists are sequences

class Link:
    empty = ()

    def __init__(self, first, rest=empty):
        assert ...
        self.first = first
        self.rest = rest

More special method names:
__getitem__  Element selection []
__len__      Built-in len function
Linked List Class

Linked lists are sequences

class Link:
    empty = ()

    def __init__(self, first, rest=empty):
        assert ...
        self.first = first
        self.rest = rest

    def __getitem__(self, i):
        if i == 0:
            return self.first
        else:
            return self.rest[i-1]

More special method names:
    __getitem__   Element selection []
    __len__      Built-in len function
Linked List Class

Linked lists are sequences

class Link:
    empty = ()

    def __init__(self, first, rest=empty):
        assert ...
        self.first = first
        self.rest = rest

    def __getitem__(self, i):
        if i == 0:
            return self.first
        else:
            return self.rest[i-1]

More special method names:

    __getitem__  Element selection []
    __len__      Built-in len function

This element selection syntax
Linked List Class

Linked lists are sequences

class Link:
    empty = ()

    def __init__(self, first, rest=empty):
        assert ...
        self.first = first
        self.rest = rest

    def __getitem__(self, i):
        if i == 0:
            return self.first
        else:
            return self.rest[i-1]

More special method names:
    __getitem__  Element selection []
    __len__     Built-in len function

Calls this method

This element selection syntax
Linked List Class

Linked lists are sequences

class Link:
    empty = ()

    def __init__(self, first, rest=empty):
        assert ...
        self.first = first
        self.rest = rest

    def __getitem__(self, i):
        if i == 0:
            return self.first
        else:
            return self.rest[i-1]

    def __len__(self):
        return 1 + len(self.rest)

More special method names:

__getitem__ Element selection []
__len__ Built-in len function

Calls this method

This element selection syntax
Linked List Class

Linked lists are sequences

```python
class Link:
    empty = ()

    def __init__(self, first, rest=empty):
        assert ...
        self.first = first
        self.rest = rest

    def __getitem__(self, i):
        if i == 0:
            return self.first
        else:
            return self.rest[i-1]

    def __len__(self):
        return 1 + len(self.rest)
```

More special method names:

- `__getitem__`  Element selection `[]`
- `__len__`  Built-in `len` function
Linked List Class

Linked lists are sequences

class Link:
    empty = ()

    def __init__(self, first, rest=empty):
        assert ...
        self.first = first
        self.rest = rest

    def __getitem__(self, i):
        if i == 0:
            return self.first
        else:
            return self.rest[i-1]

    def __len__(self):
        return 1 + len(self.rest)

More special method names:
    __getitem__  Element selection []
    __len__     Built-in len function

Calls this method

This element selection syntax

Methods can be recursive too!
(Demo)
Linked List Processing

[<map exp> for <name> in <iter exp> if <filter exp>]

(Demo)