

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

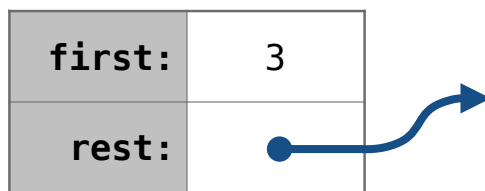
3 , 4 , 5

Linked List Structure

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

3 , 4 , 5

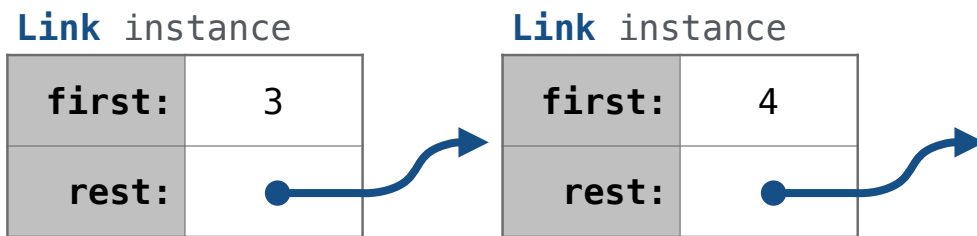
Link instance



Linked List Structure

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

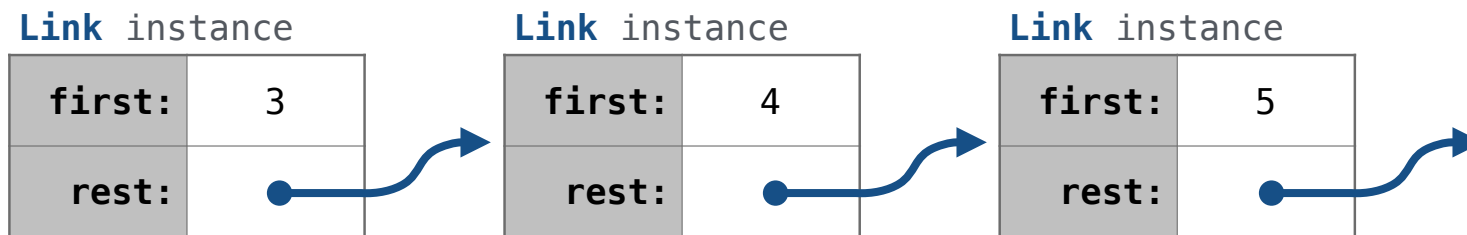
3 , 4 , 5



Linked List Structure

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

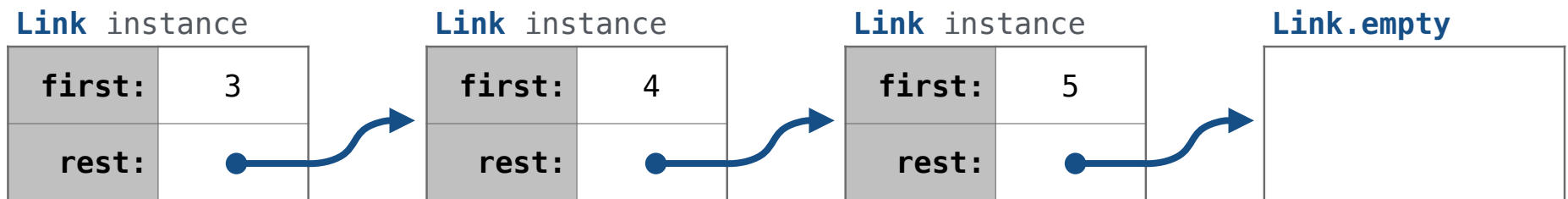
3 , 4 , 5



Linked List Structure

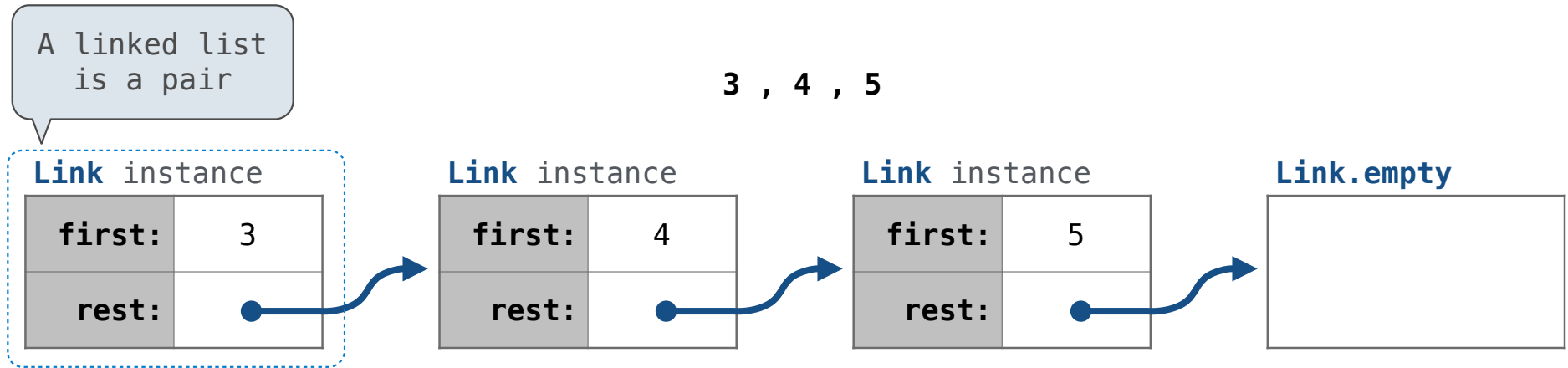
A linked list is either empty **or** a first value and the rest of the linked list

3 , 4 , 5



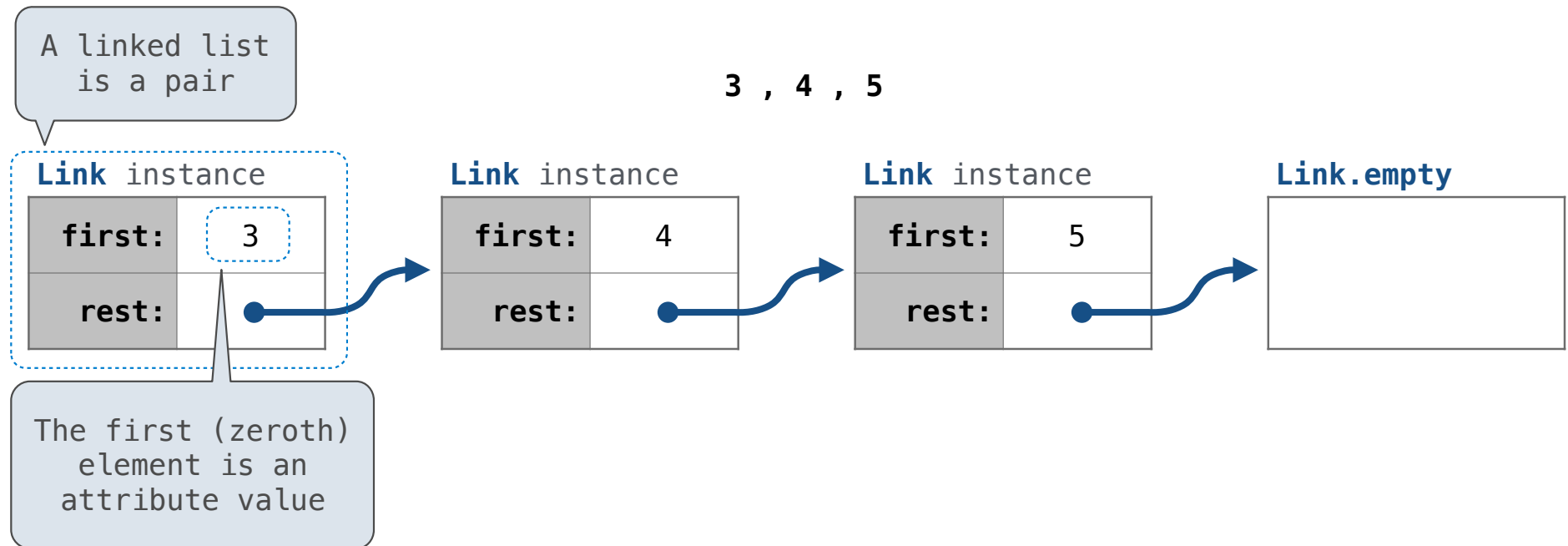
Linked List Structure

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



Linked List Structure

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

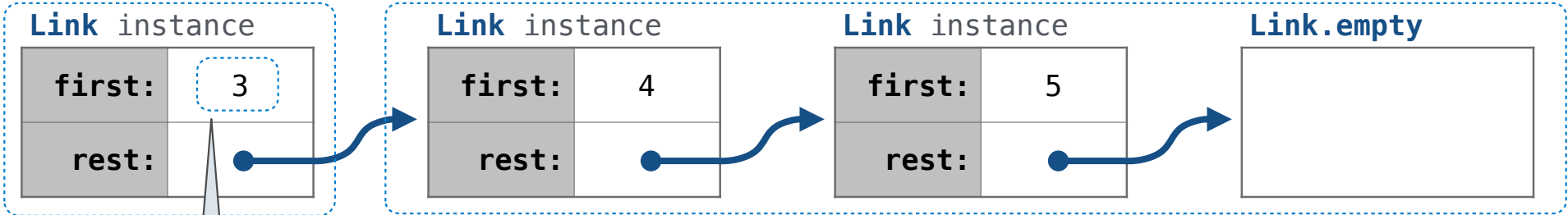


Linked List Structure

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

3 , 4 , 5

A linked list is a pair

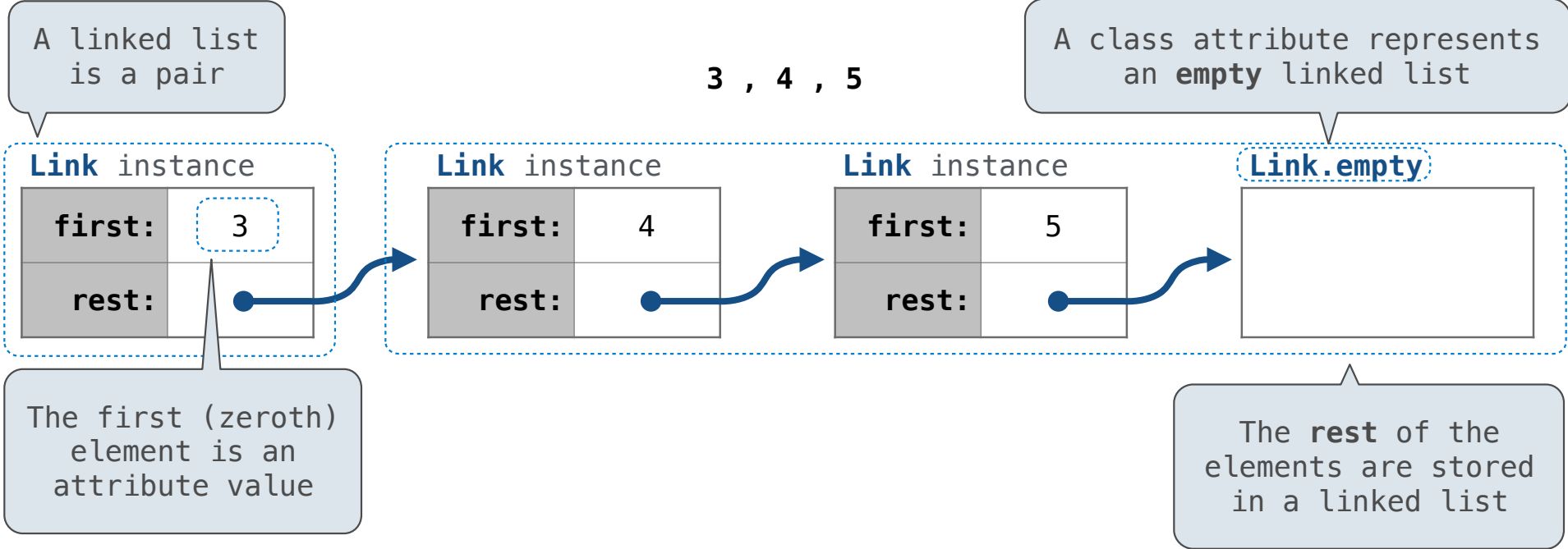


The first (zeroth) element is an attribute value

The rest of the elements are stored in a linked list

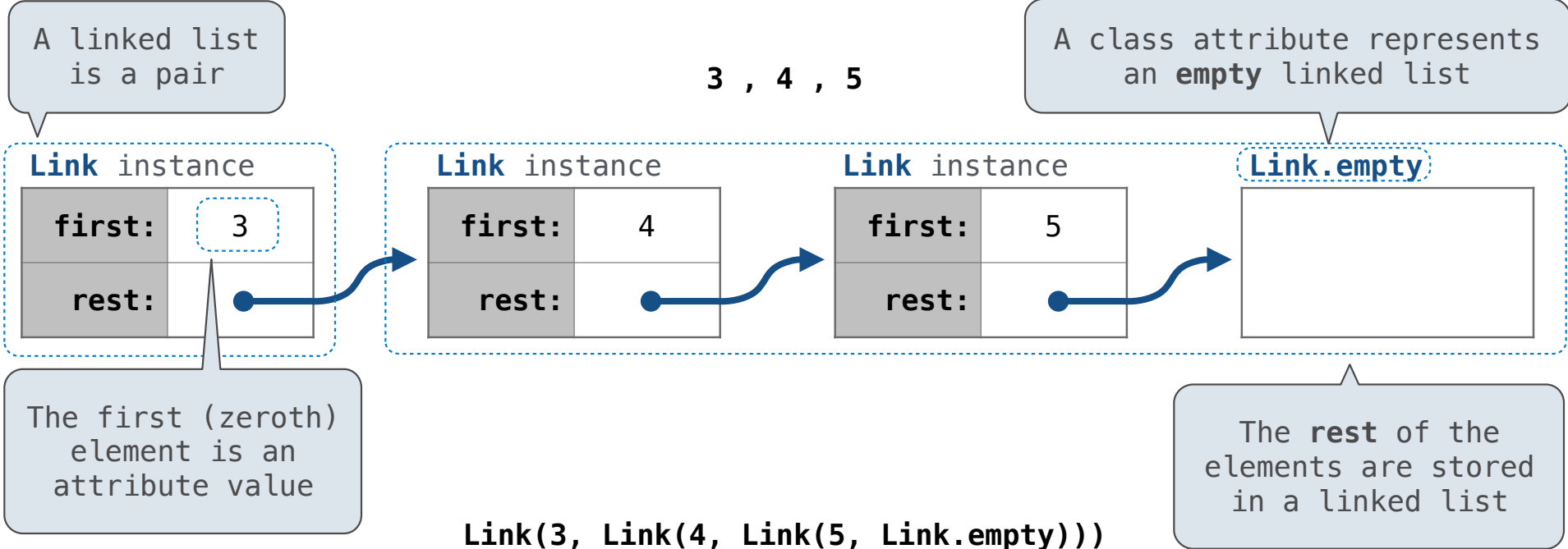
Linked List Structure

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



Linked List Structure

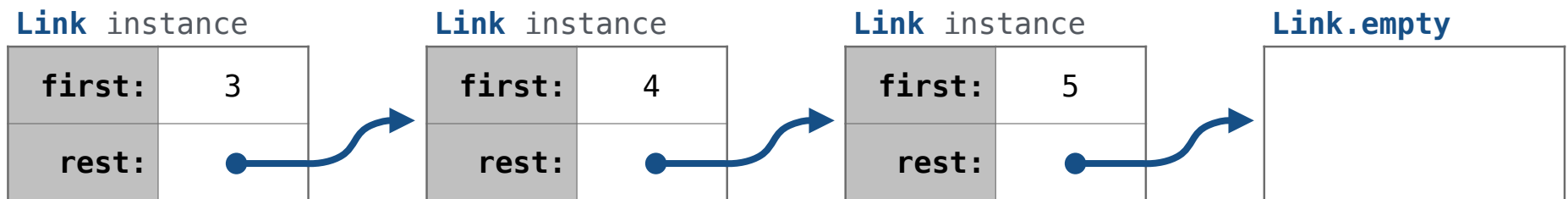
A linked list is either empty or a first value and the rest of the linked list



Linked List Structure

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

3 , 4 , 5

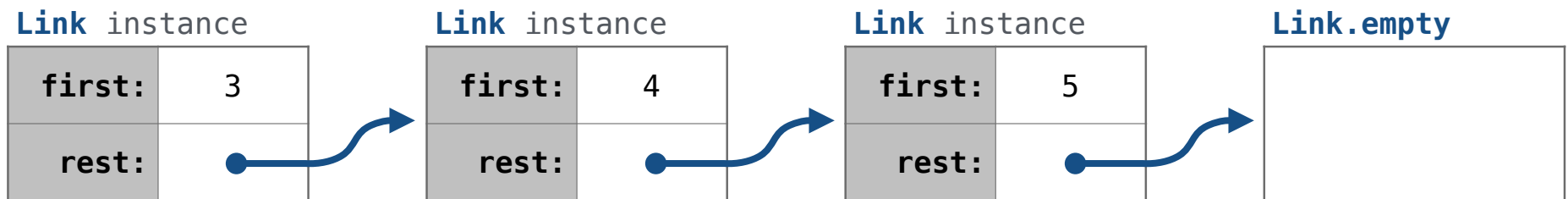


```
Link(3, Link(4, Link(5, Link.empty)))
```

Linked List Structure

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

3 , 4 , 5

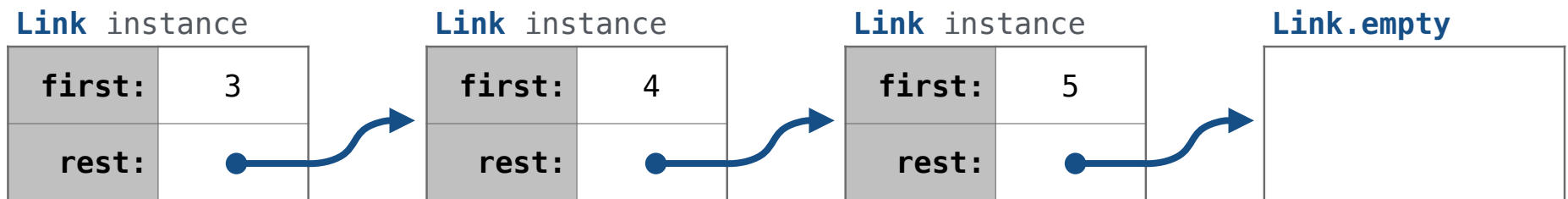


```
Link(3, Link(4, Link(5, Link.empty)))
```


Linked List Structure

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

3 , 4 , 5

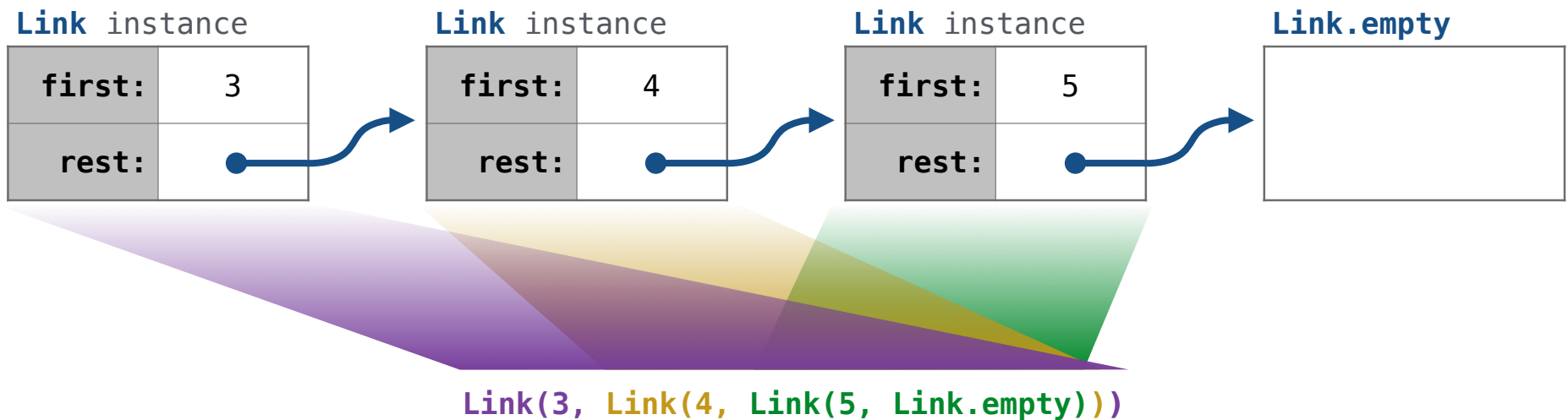


`Link(3, Link(4, Link(5, Link.empty)))`

Linked List Structure

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

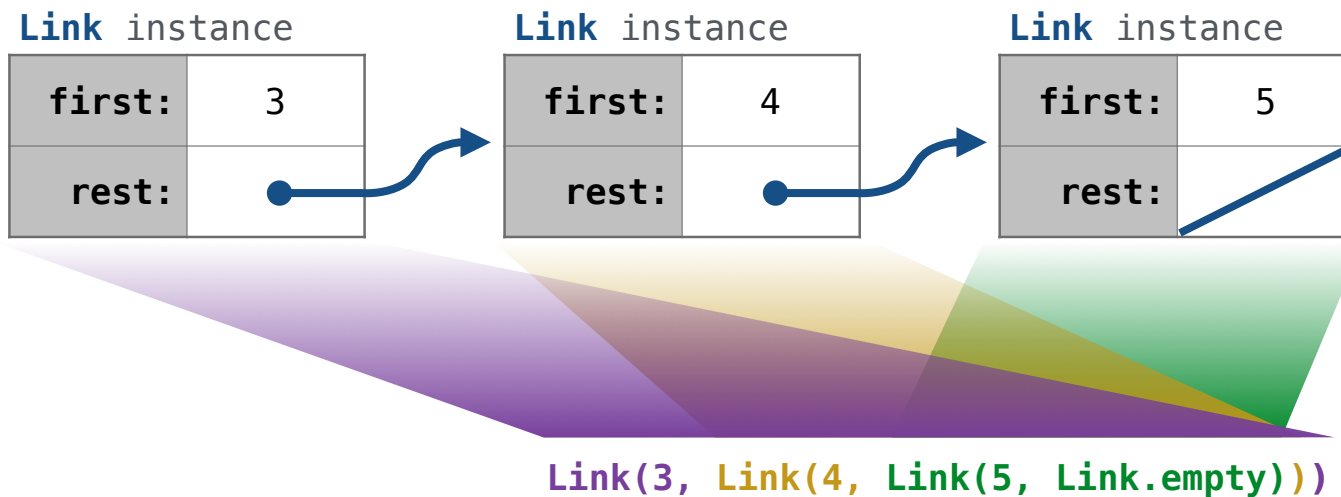
3 , 4 , 5



Linked List Structure

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

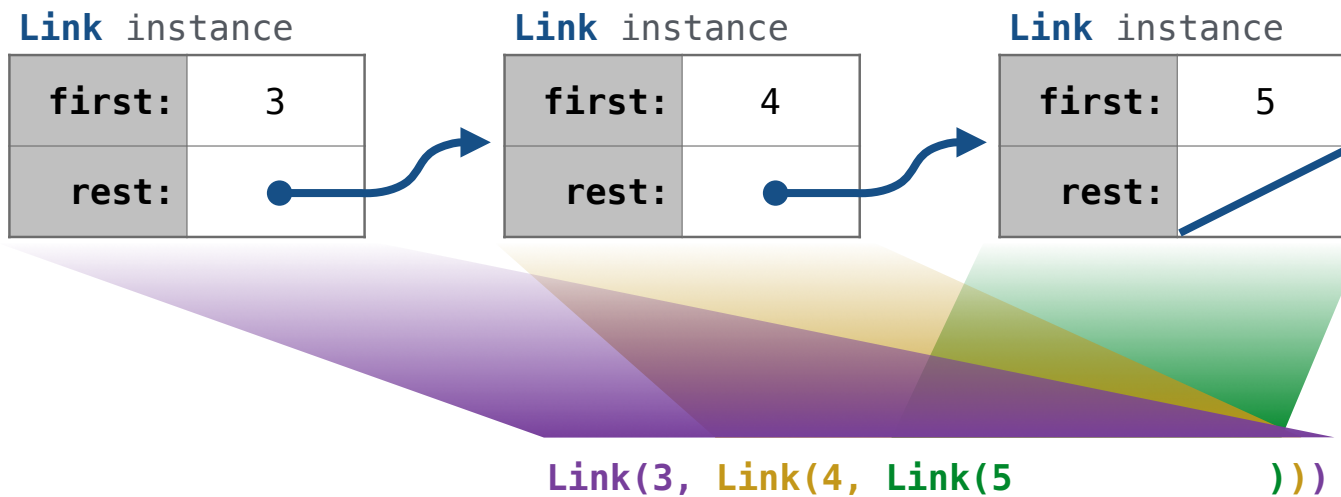
3 , 4 , 5



Linked List Structure

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

3 , 4 , 5



Linked List Class

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

Linked List Class

Linked list class: attributes are passed to `__init__`

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

Linked List Class

Linked list class: attributes are passed to `__init__`

```
class Link:
```

```
    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):
```

```
        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)
```

```
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
```

Returns whether
rest is a Link

```
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
```

Returns whether
rest is a Link

`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
```

Returns whether
rest is a Link

`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
```

Some zero-length sequence

Returns whether rest is a Link

`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
```

Some zero-length sequence

Returns whether rest is a Link

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

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

(Demo)

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:

<code>__getitem__</code>	Element selection []
<code>__len__</code>	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:

<code>__getitem__</code>	Element selection []
<code>__len__</code>	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:

<code>__getitem__</code>	Element selection []
<code>__len__</code>	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]
```

Calls this method

This element
selection syntax

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:

<code>__getitem__</code>	Element selection []
<code>__len__</code>	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)
```

Calls this method

This element selection syntax

Recursive call to __len__

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)
```

Calls this method

This element selection syntax

Recursive call to __len__

More special method names:

`__getitem__` Element selection []

`__len__` Built-in len function

Methods can be recursive too!

(Demo)

Linked List Processing

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

(Demo)