IN1010

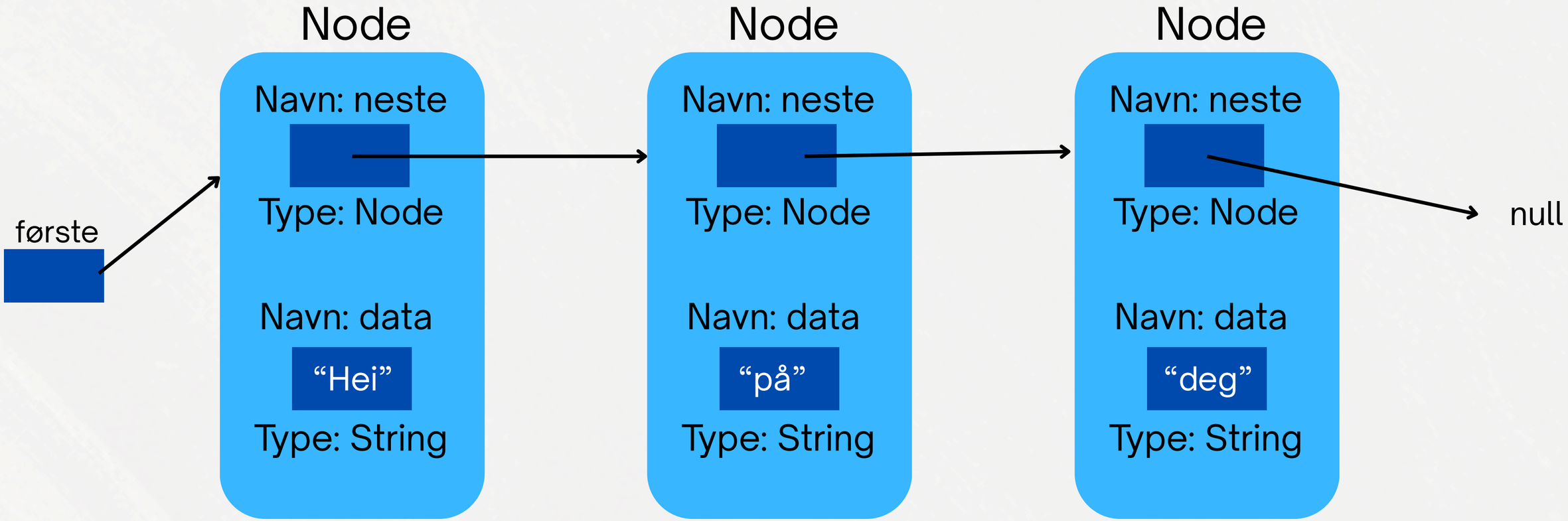
UKE 6 - GRUPPE 3

MIAU@IFI.UIO.NO

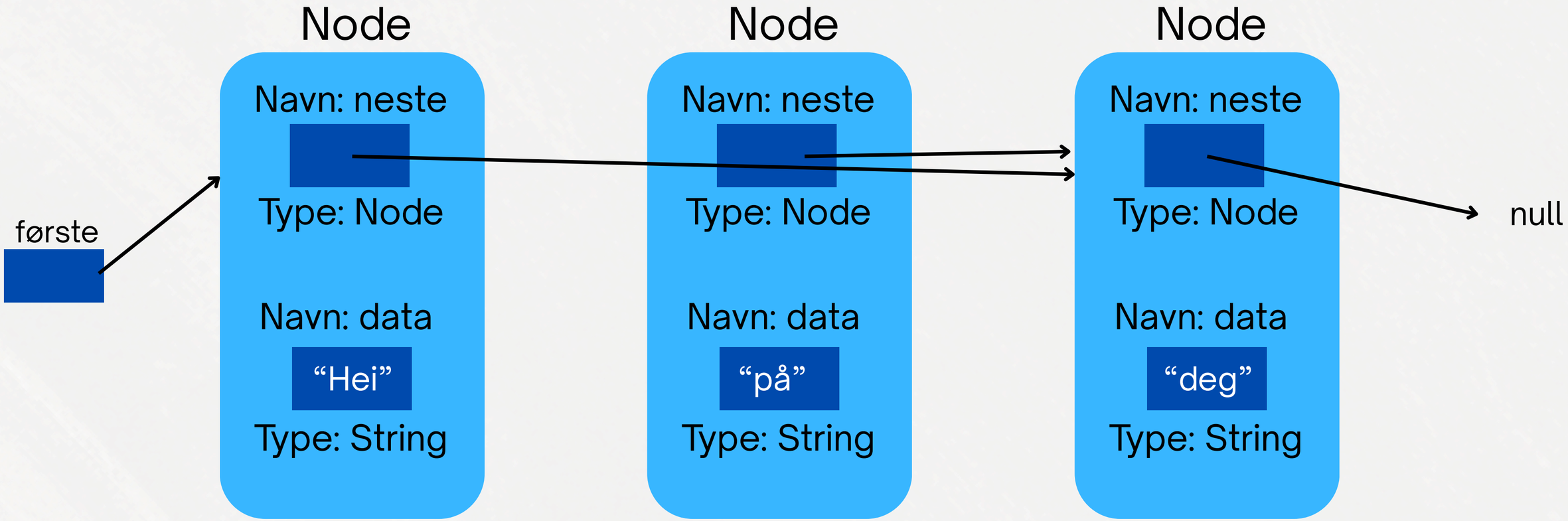
Plan for dagen

- Lenket liste
- Dobbelt lenket liste
- Datastrukturer
  - Kø
  - Stack
  - Prioritetskø
- Iterable
- Comparable

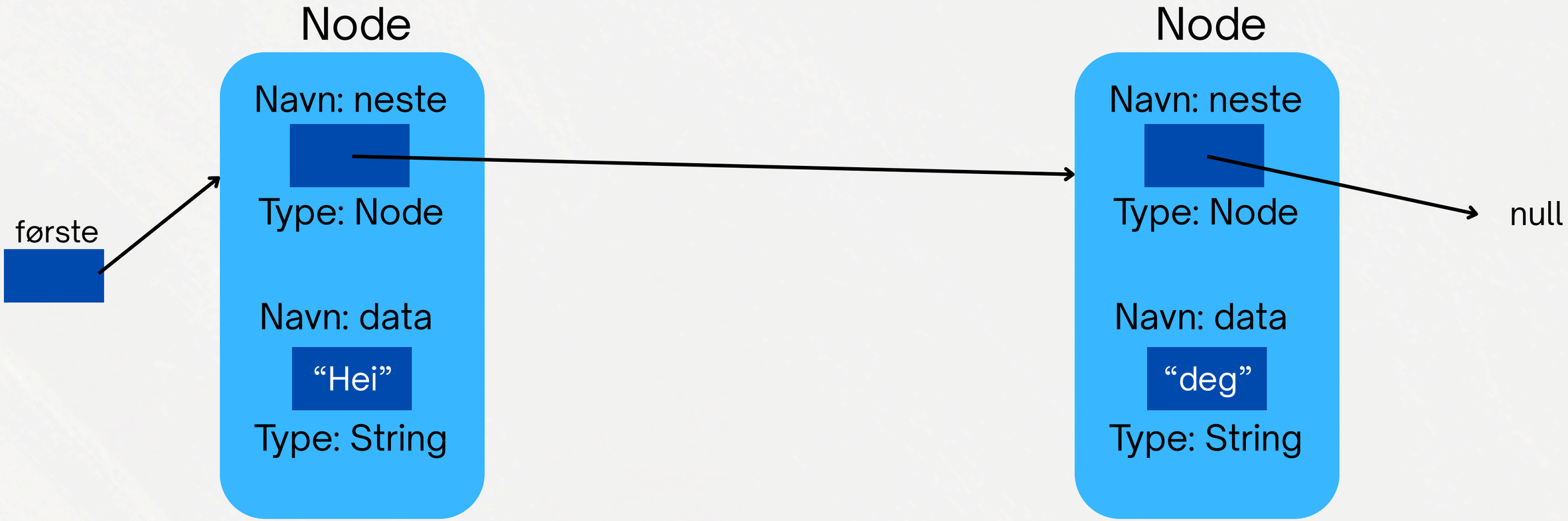
Lenket liste



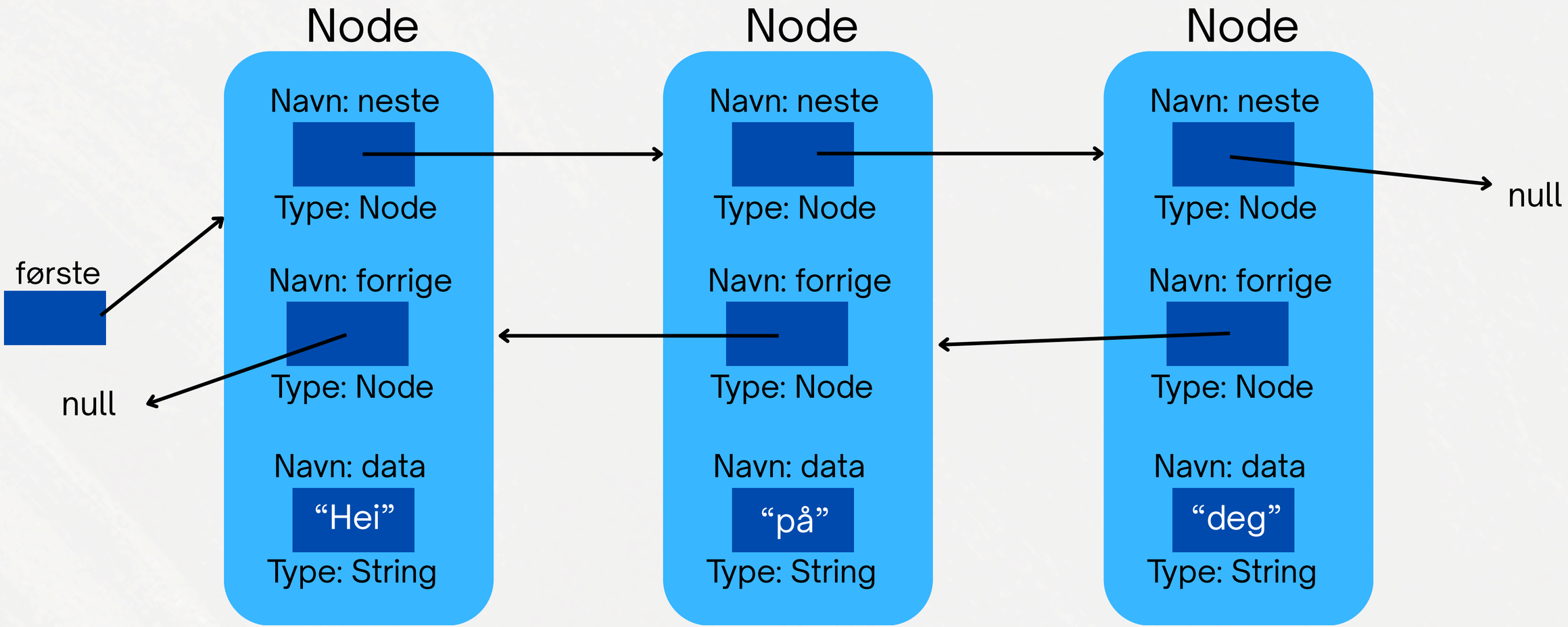
Lenket liste



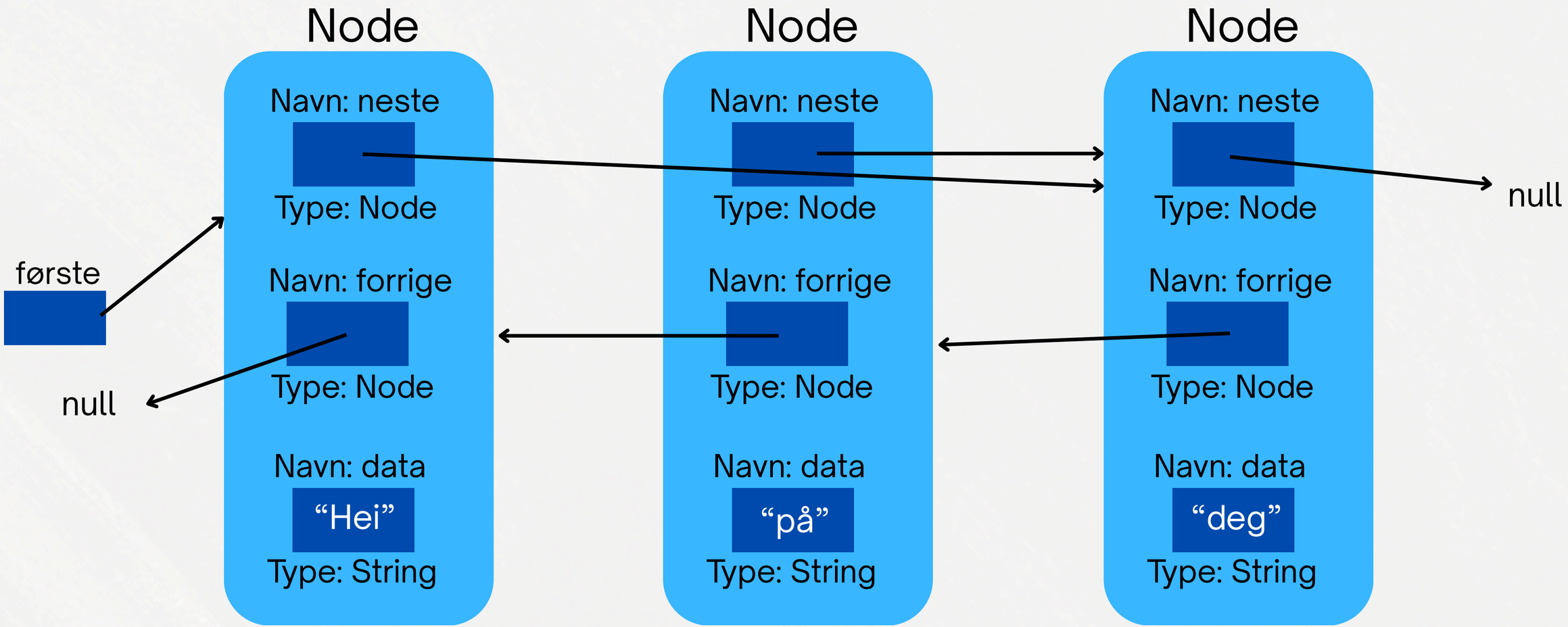
Lenket liste



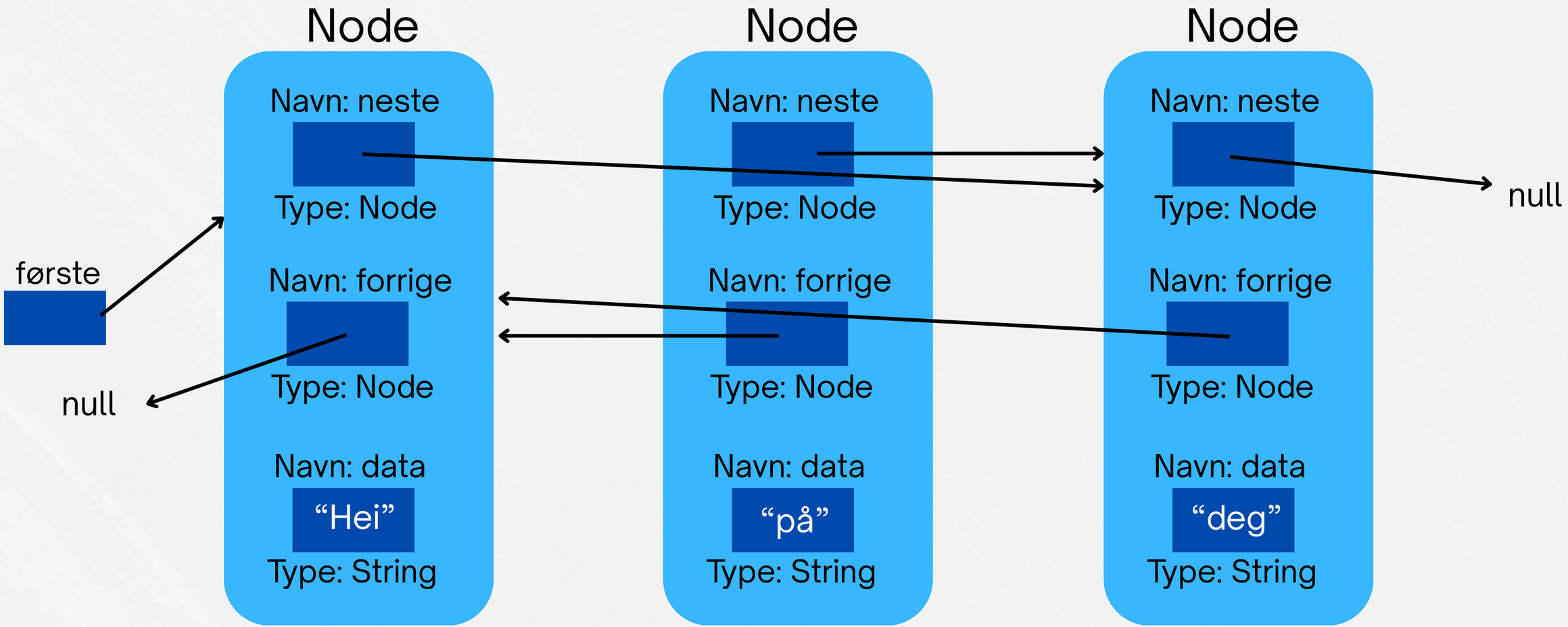
Dobbelt lenket liste



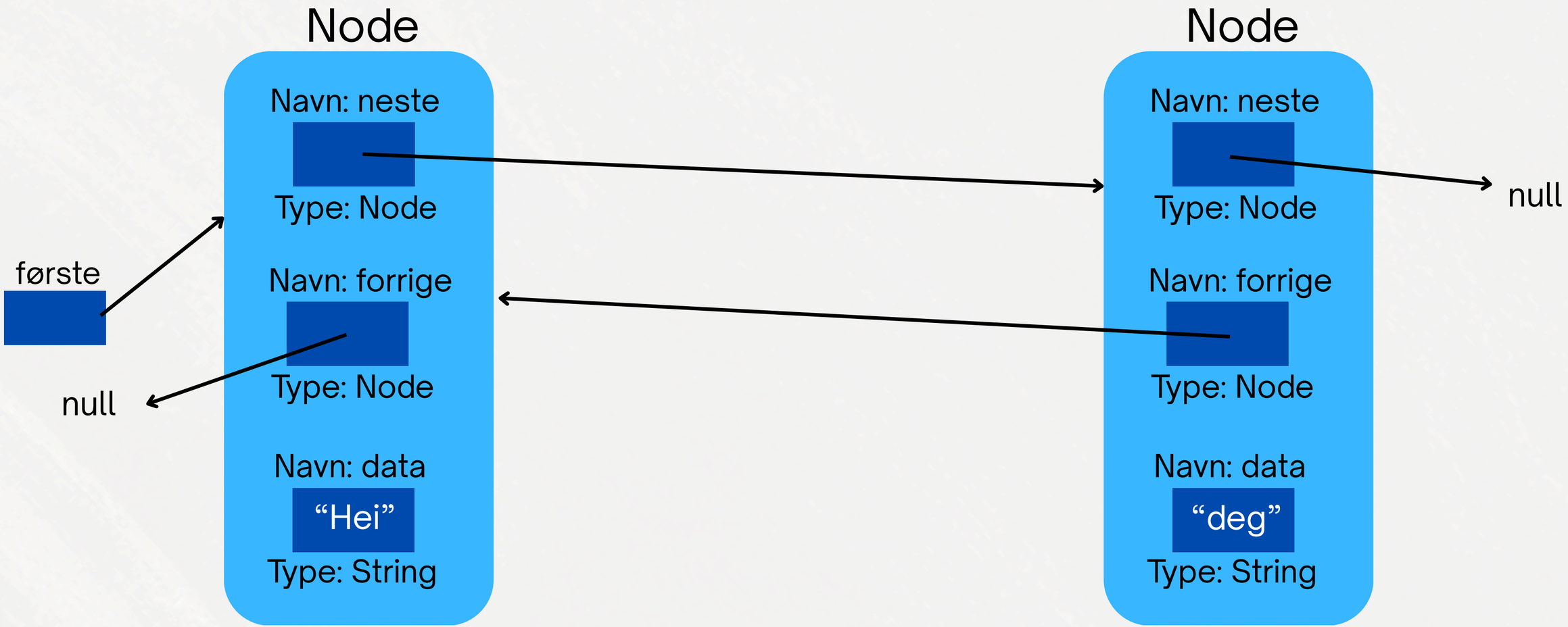
Dobbelt lenket liste



Dobbelt lenket liste

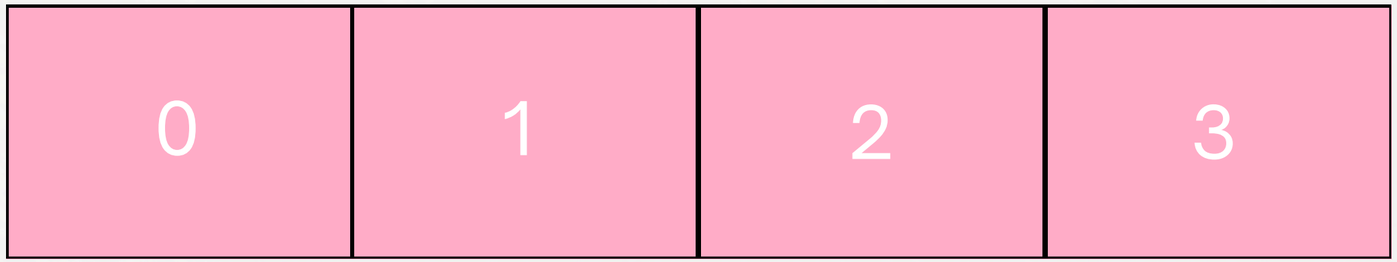


Dobbelt lenket liste

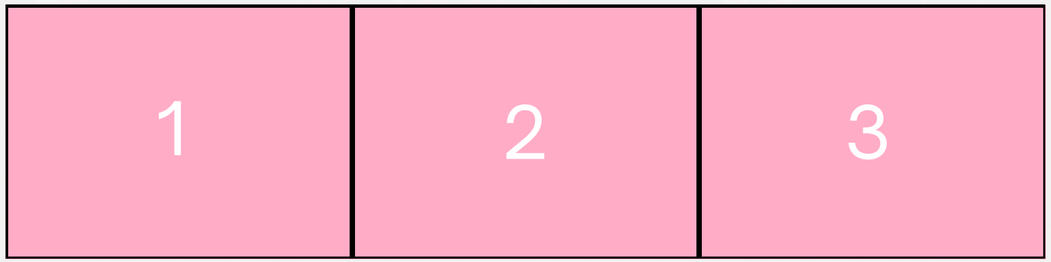


—

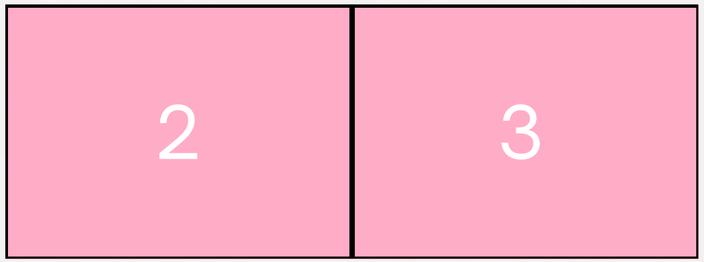
Kø



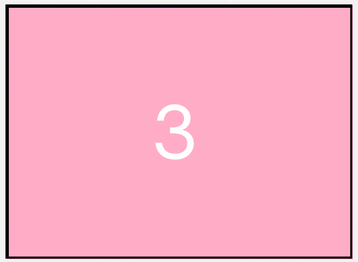
Kø



Kø

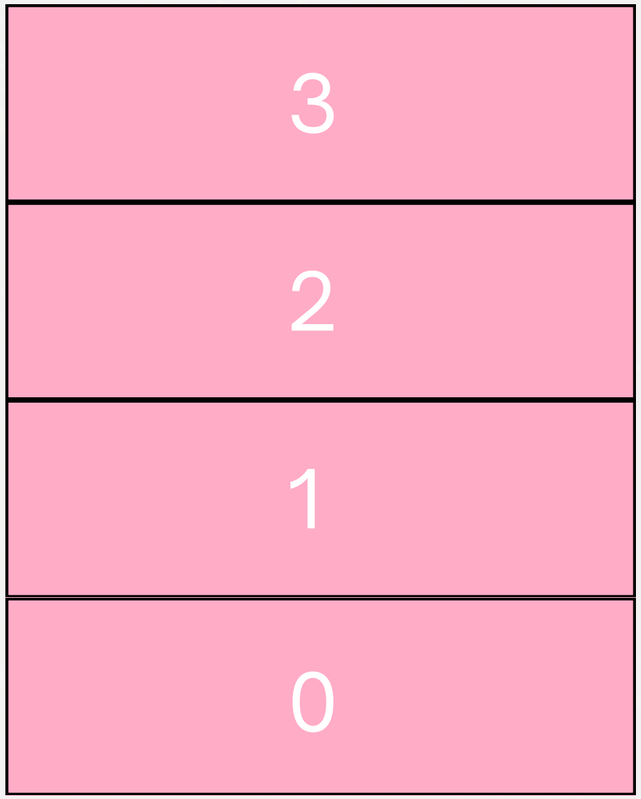


Kø

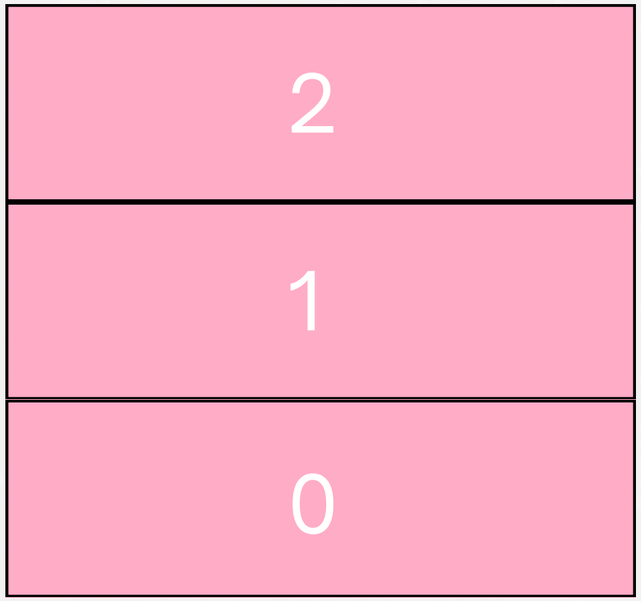


Kø

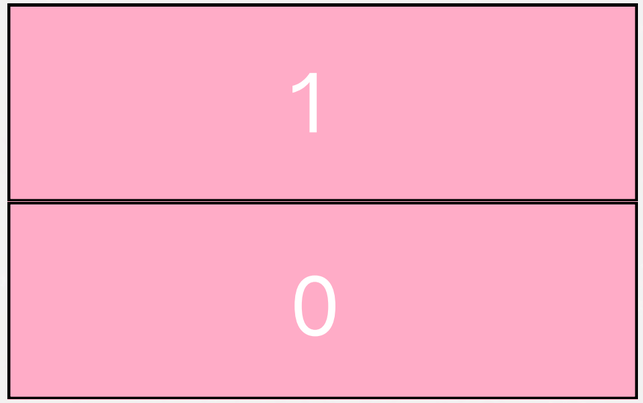
Stack



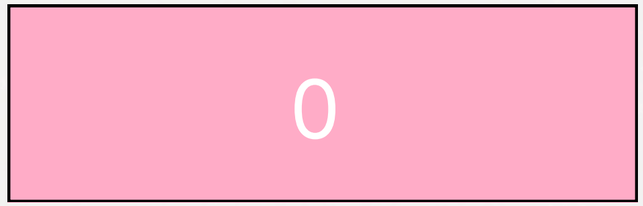
Stack



Stack

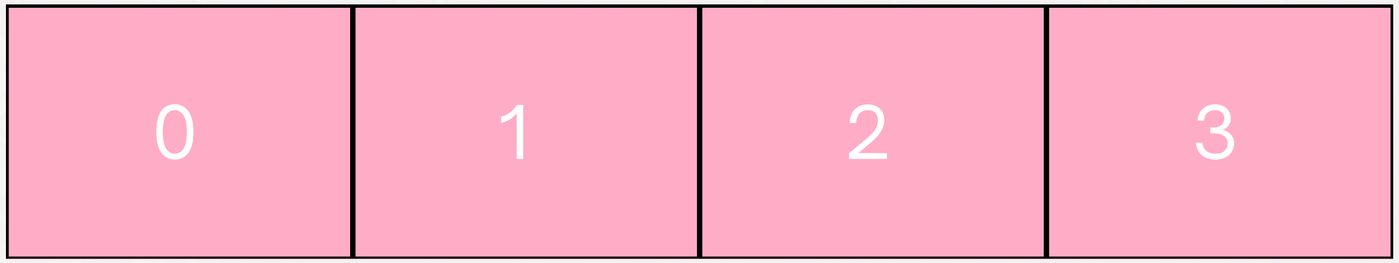


Stack

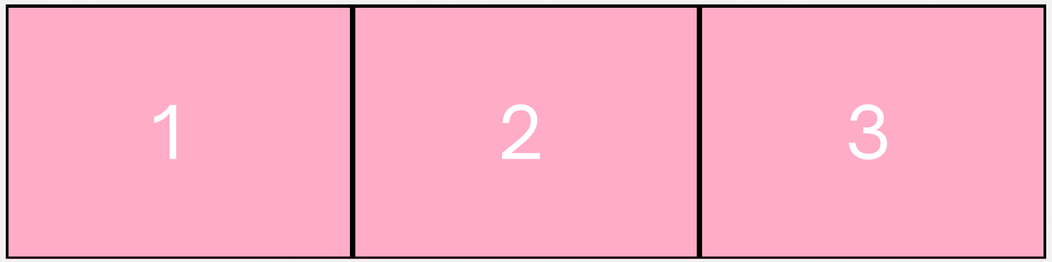


Stack

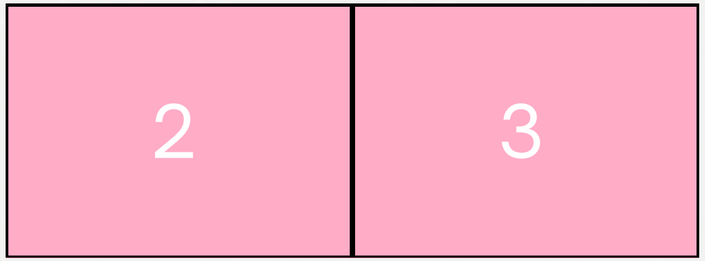
Prioritetskø



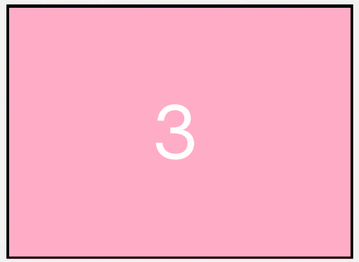
Prioritetskø



Prioritetskø



Prioritetskø



Prioritetskø

Iterable

- Gjør det mulig å iterere gjennom en beholder
  - F.eks. en (dobbelt) lenket liste
- Går element for element, og sjekker om det har en neste eller ikke

```
public interface Iterator<T> {

    boolean hasNext();
    T next();

}
```

```java
public class LenketListe<E> implements Iterable<E> {
    private class Node {
        E data;
        Node neste;

        private Node(E data) {
            this.data = data;
        }
    }

    private Node første;

    @Override
    public Iterator<E> iterator() {
        return new ListeIterator();
    }

    class ListeIterator implements Iterator<E> {
        private Node denne;

        public ListeIterator() {
            denne = første;
        }

        @Override
        public E next() {
            E temp = denne.data;
            denne = denne.neste;
            return temp;
        }

        @Override
        public boolean hasNext() {
            return denne != null;
        }
    }
}
```

```java
import java.util.Iterator;
```

```java
public static void main(String[] args) {
    LenketListe<String> lenketListe = new LenketListe<>();

    lenketListe.settInn(data: "hei");
    lenketListe.settInn(data: "på");
    lenketListe.settInn(data: "deg");

    for (String ord : lenketListe) {
        System.out.println(ord);
    }
}
```

Comparable

- Lar oss sammenlikne egne objekter
  - Kan bestemme selv hvordan de skal sammenliknes (hvilken egenskap)

```java
public static void main(String[] args) {
    Papegøye blu = new Papegøye(navn: "Blu", antallOrd: 10);
    Papegøye jewel = new Papegøye(navn: "Jewel", antallOrd: 16);

    if (blu.compareTo(jewel) == 1) {
        System.out.println(x: "Blu kan si flere ord.");
    } else if (blu.compareTo(jewel) == -1) {
        System.out.println(x: "Jewel kan si flere ord.");
    } else {
        System.out.println(x: "De kan si like mange ord.");
    }
}
```

```java
public class Papegøye implements Comparable<Papegøye> {
    String navn;
    int antallOrd;

    public Papegøye(String navn, int antallOrd) {
        this.navn = navn;
        this.antallOrd = antallOrd;
    }

    @Override
    public int compareTo(Papegøye papegøye) {
        if (antallOrd > papegøye.antallOrd) {
            return 1;
        }

        else if (antallOrd < papegøye.antallOrd) {
            return -1;
        }

        return 0;
    }
}
```