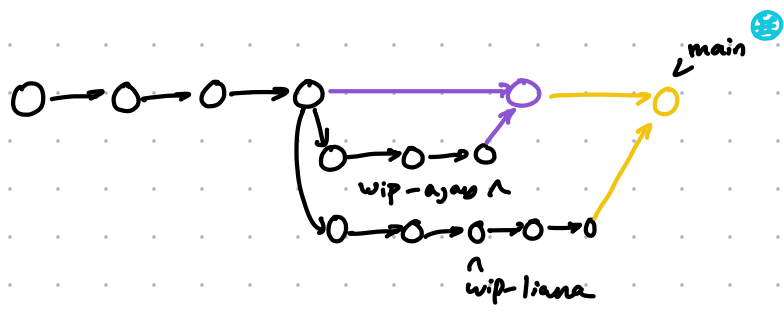
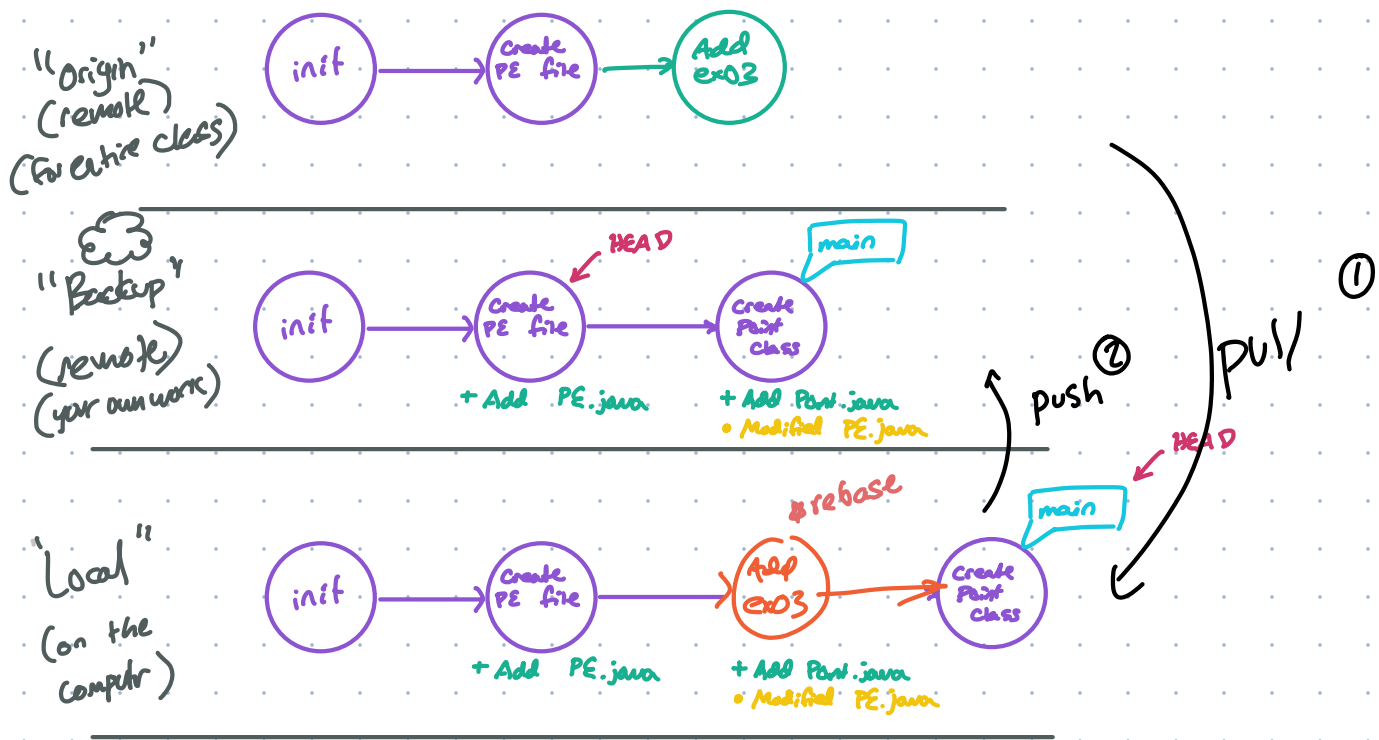


# Intro to Git



```
public interface Employee {
```

```
void introduce();
```

```
} double saySalary();
```

return type

```
public class Professor implements Employee {
```

```
String name; double salary;
```

```
public Professor (String name, double salary) {
```

```
this.name = name;
```

```
this.salary = salary;
```

```
}
```

```
public void introduce() {
```

```
System.out.println("Hi, I am Dr." + this.name);
```

```
}
```

```
public double saySalary() {
```

```
return this.salary;
```

```
}  
public void sayResearchArea() {}
```

```
public class Staff implements Employee {
```

```
String name; double salary;
```

```
public Staff (String name, double salary) {
```

```
this.name = name;
```

```
this.salary = salary;
```

```
}
```

```
public void introduce() {
```

```
S.o.println("Hi, I am " + this.name);
```

```
}
```

```
}
```

```
ArrayList < Employee > roster = new ArrayList <> ();
```

```
Professor kati = new Professor ("kati", 10000000000);
```

```
Staff erin = new Staff ("erin", 999999999999);
```

```
roster.add(kati);
```

```
roster.add(erin);
```

```
for (Employee e : roster) {
```

```
e.introduce();
```

→ "Hi, I am Dr. Kati"

→ "Hi, I am Erin"

```
}
```

Employee e = roster.get(1);

Professor p = (Professor) e;

