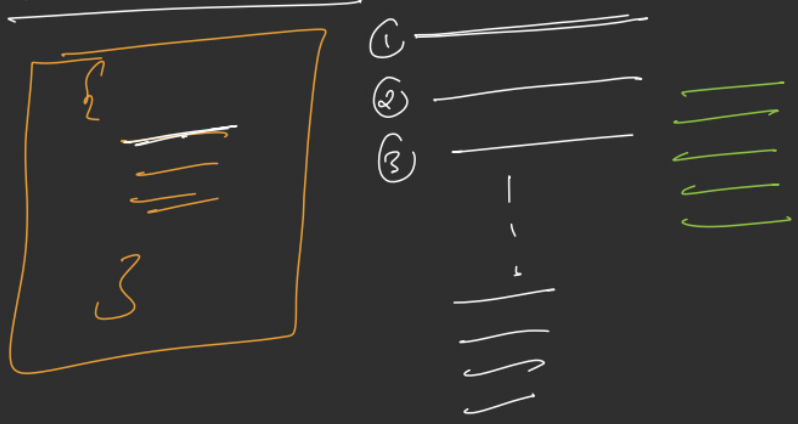
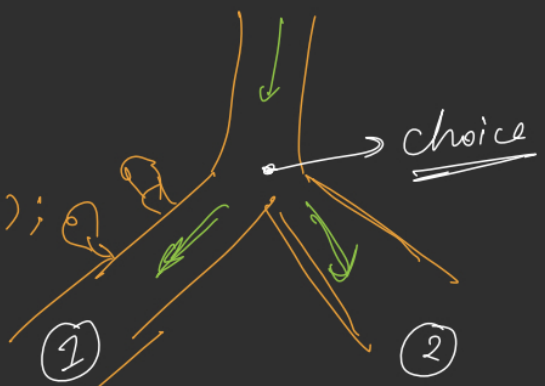
Flow of Control

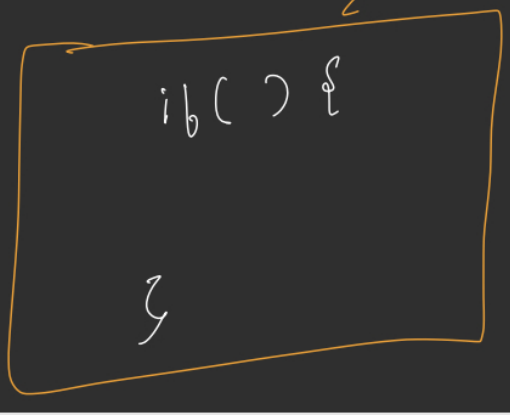


Flow of Control

- Selection ✓
  - if()
  - if-else()
  - switch()
- Iteration
- Jumps

```
{
  int i = 4;
  System.out.println(i);
  int j = 5;
}
```





if (expression) {
    // do something.
}

True / False
boolean.

boolean b = true; → False

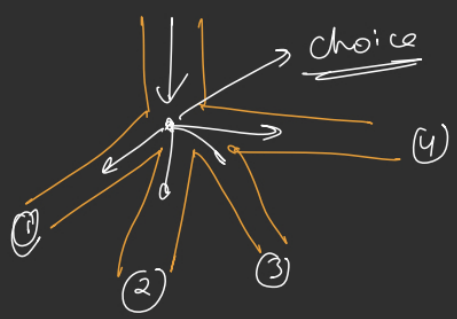
if (b == true) {
    // do something
}

b == true

if () {
}
else {
}

Switch statements

↳ if-else-if ladder
↳ More optimized.



if (choice == 1) {
    // go to 1
}
else if (c == 2) {
    // go to 2
}

Switch v/s if-else-if ladder

1. Switch can only test equality but if-else can test both equality & unequality.
2. Switch is more efficient than if-else-if ladder.

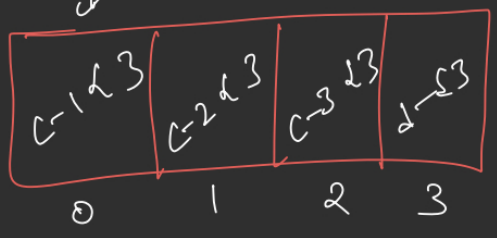
```
int i = 2;
```

```
Switch (i) {
    Case 1 :
        // do something
        break;
    Case 2 :
        // do something
        break;
    Case 3 :
        // do something
        break;
    default :
        // do something
        break;
}
```

```
if (i == 1) {
    // do something
}
else if (i == 2) {
    // do something
}
else if (i == 3) {
    // do something
}
else {
    // do something.
}
```

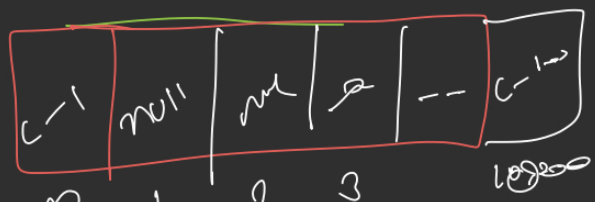
i → 2

Jump tables



Jump table

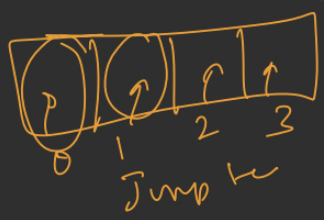
↳ Jump tables are not always efficient

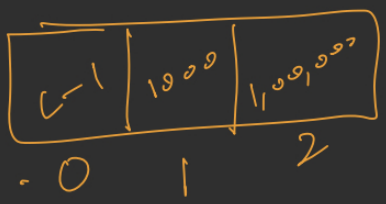



```
i = 3
switch(i){
    case 1:
    case 1000:
    case 1,00,000:
}
```

→ Sparse values

Jump table






```
i = 3000
case 1:
case 1000:
case 3000:
case 10,000:
  :
  :
case 1,00,000
```

Switch Statements

Nested switch.