

## Java syntax and programming structures

Mitsiu Alejandro Carreño Sarabia

## Starbucks Java Chip Frappuccino Recipe



double  
espresso



2 tbsp  
chocolate  
chips



1/2 cup  
ice



1/2 cup  
whole milk



3 tbsp  
chocolate  
syrup



whipped  
cream

1

Brew the espresso and set it aside to let it cool. You can also use 1 tbsp instant coffee with 2 oz water.

2

Add ice to a blender followed by the milk, chocolate chips, and half of the chocolate syrup.

3

Make sure the coffee is cooled down, and add it to the blender.

4

Blend well.

5

Drizzle some chocolate syrup along the sides of a clear, tall cup or glass.

6

Pour the blended mixture in.

7

Top off with whipped cream and drizzle the rest of the chocolate syrup on.



BIG CUP OF

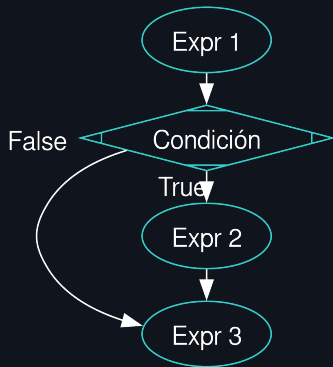
## Agenda

- If
  - If-else
  - If-else if
  - If-else if-else
- For-each
  - Break
  - Continue
- While
- Functions
  - Parameters
- Notes

## ■ If (branching)

---

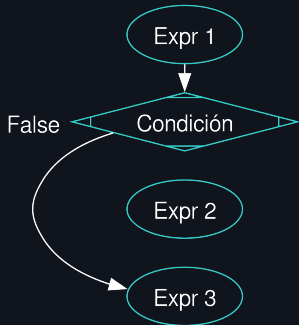
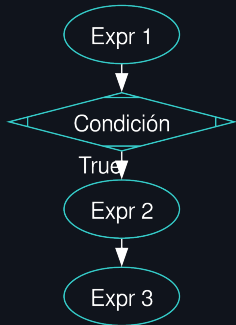
```
1 Expr 1  
2 if (cond) {  
3     Expr 2  
4 }  
5 Expr 3
```



## ■ If (branching)

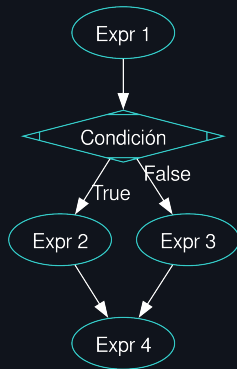
---

If it's false it jumps Expr 3



## ■ If-else

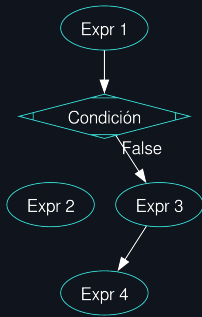
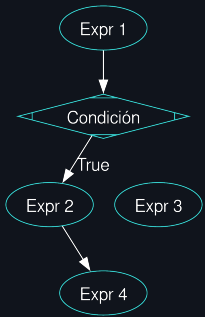
```
1 Expr 1
2 if (cond) {
3     Expr 2
4 } else {
5     Expr 3
6 }
7 Expr 4
```



## ■ If-else

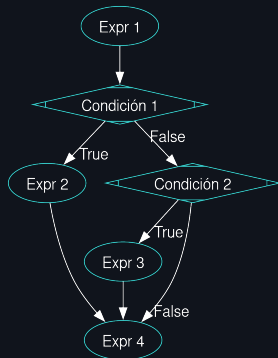
---

Always jumps an expression



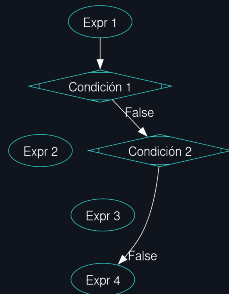
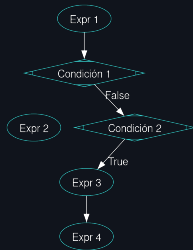
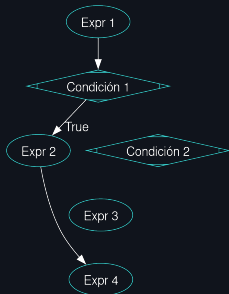
## ■ If-else if

```
1 Expr 1
2 if (cond 1) {
3     Expr 2
4 } else if (cond 2) {
5     Expr 3
6 }
7 Expr 4
```



## ■ If-else if

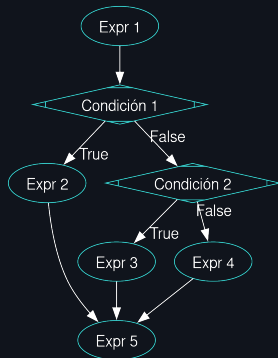
- Either Expr 2 or Expr 3 or none





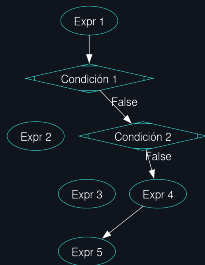
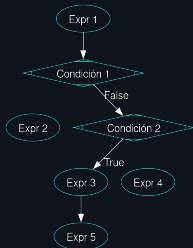
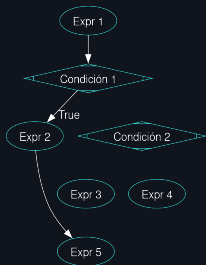
## ■ If-else if-else

```
1 Expr 1
2 if (cond 1) {
3     Expr 2
4 } else if (cond 2) {
5     Expr 3
6 } else {
7     Expr 4
8 }
9 Expr 5
```



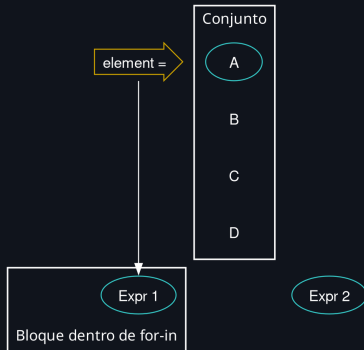
## ■ If-else if-else

- Condition 2 it's just a regular if-else (Expr 3 or Expr 4)
- Condition 1 it's just a regular if-else (Expr 2 or Cond 2)
- Between Expr 2, Expr 3 and Expr 4 only one will execute



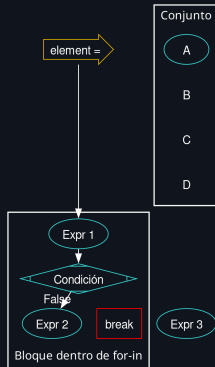
## ■ For-each (Archivo: ./assets/gviz/for-in.gif)

```
1 String[] conjunto =  
2     {"A", "B", "C", "D"};  
3  
4 for (String element :  
5     conjunto){  
6     Expr 1  
7 }  
8 Expr 2
```



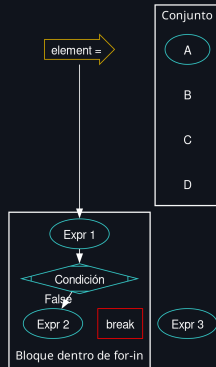
## ■ For-each break (Archivo: ./assets/gviz/for-in-break.gif)

```
1 String[] conjunto =  
2     {"A", "B", "C", "D"};  
3  
4 for (String element :  
5     conjunto){  
6     Expr 1  
7     if (Cond 1) {  
8         break;  
9     }  
10    Expr 2  
11 Expr 3
```



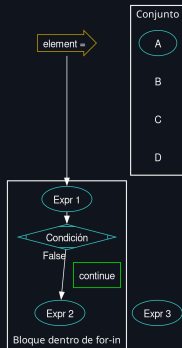
## ■ For-each break (Archivo: ./assets/gviz/for-in-break.gif)

- Let's suppose that condition is true when element is C
- Our loop **does NOT** reach element D



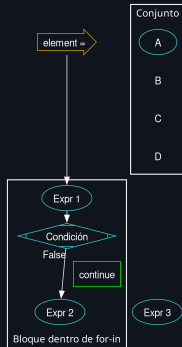
## ■ For-each continue (Archivo: ./assets/gviz/for-in-cont.gif)

```
1 String[] conjunto =  
2     {"A", "B", "C", "D"};  
3  
4 for (String element :  
5     conjunto){  
6     Expr 1  
7     if (Cond 1) {  
8         continue;  
9     }  
10 }  
11 Expr 3
```



## ■ For-each continue (Archivo: ./assets/gviz/for-in-cont.gif)

- Let's suppose that condition is true when element is C, at that iteration Expr 2 is skipped
- Our loop does **reach all** elements

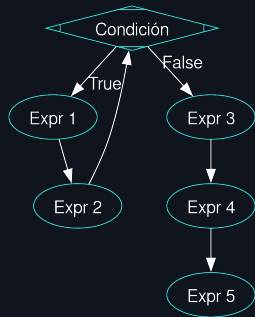


## ■ While

---

A while loop can potentially become infinite

```
1 while (cond) {  
2     Expr 1  
3     Expr 2  
4 }  
5 Expr 3  
6 Expr 4  
7 Expr 5
```





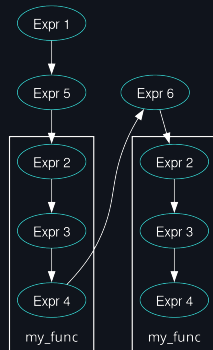
## ■ Functions

---

- A set of expressions executed in **sequential order**
- It's **reusable** and completes a **specific task**
- A function definition (implementation) defines the expressions its composed of **`() {...}`**
- A function **invocation** **`();`** executes the instructions defined **`() {...}`**

## Functions

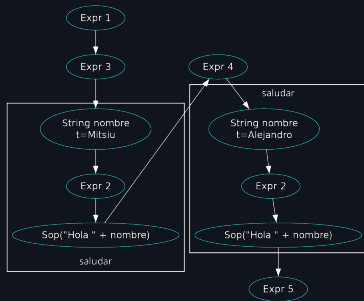
```
1 Expr 1
2
3 void myFunc(){
4     Expr 2
5     Expr 3
6     Expr 4
7 }
8
9 Expr 5
10 myFunc();
11 Expr 6
12 myFunc();
```



## ■ Functions + parameters

- We can alter the function result by sending **parameters**

```
1 Expr 1
2
3 void saludar(String
  nombre) {
4     Expr 2
5     System.out.print("Hola
  " + nombre)
6 }
7 Expr 3
8 saludar("Mitsiu");
9 Expr 4
10 saludar("Alejandro");
11 Expr 5
```





All images and animations can be found at:  
**06-syntax/presenterm/assets/gviz**