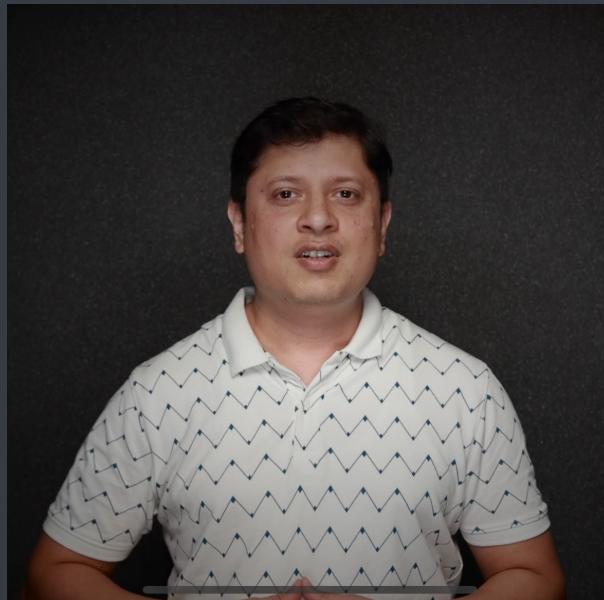


Java for beginners

Iterative control



Saurabh Shukla (MySirG)

Agenda

- ① Iterative Control statements
- ② while
- ③ do-while
- ④ for
- ⑤ break
- ⑥ continue
- ⑦ nested loops
- ⑧ for-each loop

Iterative Control Statements

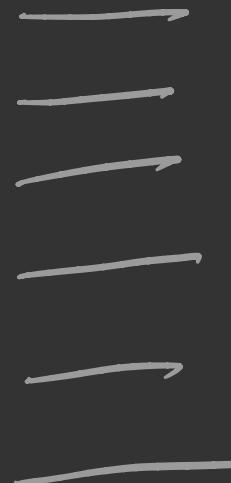
while

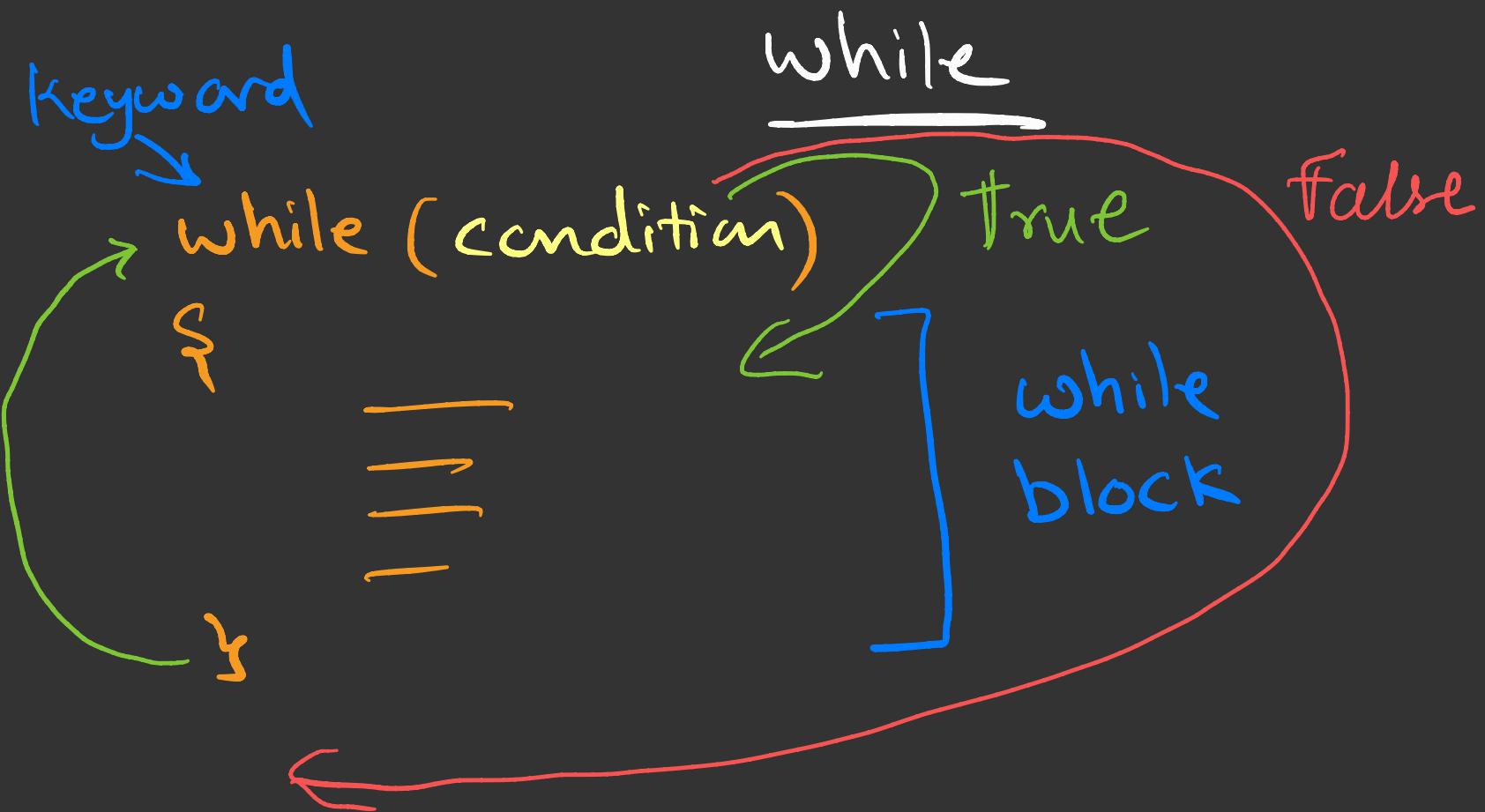
Loop

do while

for

for each





Condition must be evaluated as a boolean value (true or false)

Write a Java program to print MySirG
5 times on the screen.

public class Example {

 public static void main(String[] args) {

 int i=1;

 while(i<=5) {

 System.out.println("MySirG");

 i++;

 }

}

}

- 1) MySirG
- 2) MySirG
- 3) MySirG
- 4) MySirG
- 5) MySirG

Write a Java program to print first 10 natural numbers.

```
public class Example{  
    public static void main(String[] args){  
        int i=1;  
        while( i<=10){  
            System.out.println(i);  
            i++;  
        }  
    }  
}
```

Write a Java program to print first
10 natural numbers in reverse order.

10 9 8 7 6 5 4 3 2 1



int i = 10;

while(i >= 1) {

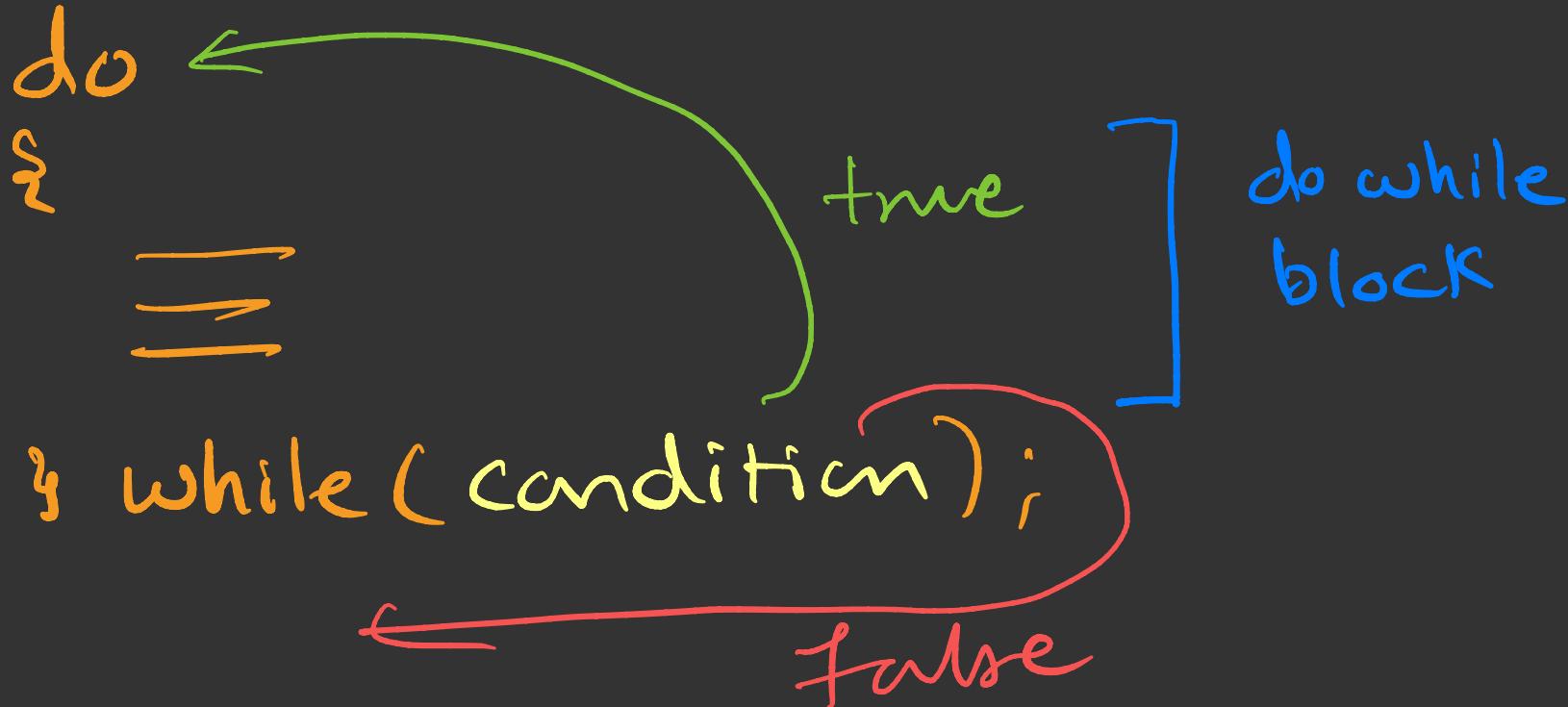
 System.out.println(i);

 i--;

}

10 9 8 7 6 5 4 3 2 1

do while



Comparison between while and do-while

Entry control loop

while (condition)

{
 |=|
 |
}|



```
int i=1;  
while (i<=5)  
{  
  System.out.println(i);  
  i++;  
}
```

1 <= 5 T
2 <= 5 T
3 <= 5 T
4 <= 5 T
5 <= 5 T
6 <= 5 F

Exit control loop

do
{|
 |=|
 |
}|

} while (condition);



```
int i=1;  
do  
{  
  System.out.println(i);  
  i++;  
} while (i<=5);
```

2 <= 5 T
3 <= 5 T
4 <= 5 T
5 <= 5 T
6 <= 5 F

80

A hand-drawn diagram illustrating the components of a for loop. The code is written in orange: `for (; ;)`. A blue bracket on the right is labeled "for block". Three pink arrows point from the words "initialization", "condition", and "flow" to the opening parenthesis, the first semicolon, and the closing parenthesis respectively. A blue arrow points from the word "Keyword" to the "for" keyword. Three yellow double-headed arrows below the loop body represent iteration.

Write a Java program to print first
10 multiples of 5. (using for loop)

```
for (int i=1 ; i<=10 ; i++) {  
    System.out.println( i * 5 )  
}
```

Nested Loop

```
while( )  
{
```

```
    for( ; ; )  
    {
```

```
        while( )  
        {  
            =  
        }
```

```
    while( )  
    {
```

```
}
```

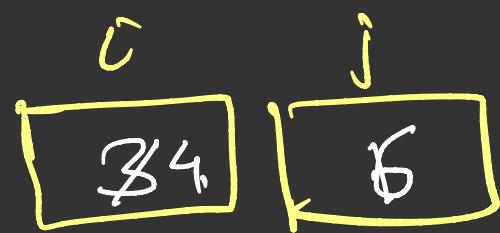
```
    }  
}
```

for ($i=1$; $i \leq 4$; $i++$)

for ($j=6$; $j > 1$; $j--$)

S.O.P (i+j)

7 6 5 4 3
8 7 6 5 4
9 8 7 6 5
10 9 8 7 6



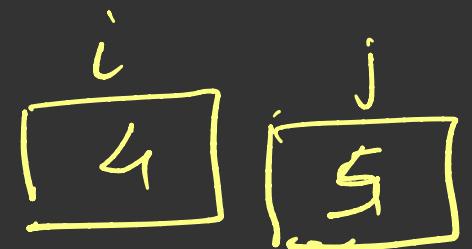
break

- The break is a control transfer statement.
- It is a keyword.
- It is used in the body of loop to terminate the execution of the loop.
- When the break statement is encountered inside a loop, the loop iteration stops and control moves to the first statement after the loop.
- break can also be used in the body of switch.

Find Output of the code

```
int i;  
for (i=1 ; i<=10 ; i++)  
{    if (i==5)  
        break;  
    System.out.println("i=" + i);  
}  
System.out.println("outside loop");
```

There is a need to break out from both the loops when sum of i and j becomes greater than 8.



```
for (int i=1 ; i<=10; i++)
```

```
{
```

```
    for (int j=1 ; j<=5 ; j++)
```

```
{
```

```
        if (i+j > 8)
```

```
            break;
```

```
        System.out.println(i+j);
```

```
}
```

```
}
```

2 3 4 5 6 3 4 5 6 7 4 5 6 7

Labelled break

first :

```
for (int i=1 ; i<=10; i++)
```

```
{
```

```
    for (int j=1 ; j<=5 ; j++)
```

```
{
```

```
        if (i+j > 8)
```

```
            break first;
```

```
        System.out.println(i+j);
```

```
}
```

```
}
```

Continue

- Continue is a keyword
- It is used only in the body of loop
- It is used to skip the current iteration

```
while (condition)
{
    ==
    if (...)

    continue;

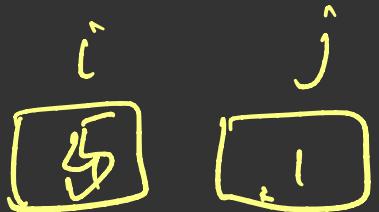
    ==
}

{
```

```
for ( ; ; )
{
    ==
    ==
    continue;
    ==
}

}
```

Find Output



```
for ( int i=1; i<=4; i++ ) {
```

```
    for( int j=1; j<=5 ; j++ ) {
```

```
        if ( j>i)  
            continue;
```

```
        System.out.print( "*" );
```

```
}
```

```
System.out.println( );
```

```
}
```

*
* *
* * *
* * * *

You can use labelled continue as well.

for - each

for each loop is applicable in collection of values to access all the elements one by one from first element to the last element.

for (type var : array)

{ ==
 }

}