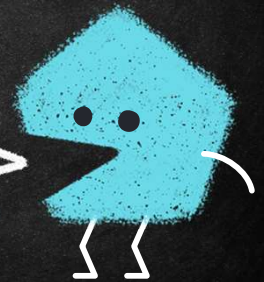
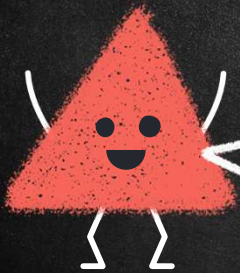




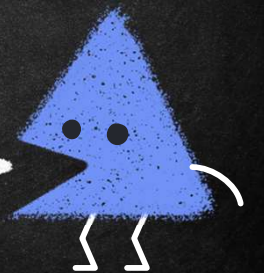
# OPERACIONES

Operadores aritméticos  
Operadores relacionales  
Operadores lógicos



## OPERACIONES ARITMÉTICAS BÁSICAS

+	Suma
-	Resta
*	Multipliación
/	División
%	Módulo: Resto de la división entera

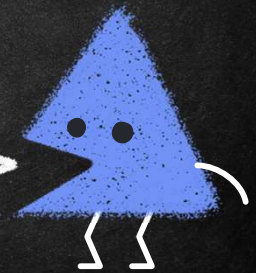




5

## OPERACIONES ARITMÉTICAS

++	Incremento en 1
--	Decremento en 1
+=	Suma y asignación
-=	Resta y asignación
*=	Multiplicación y asignación
/=	División y asignación
%=	Módulo y asignación



6

## EJERCICIO 1 – INDIVIDUAL

Sin ejecutar el programa...

¿Cual es el valor de x cuando se imprime por pantalla?

```
public class OperacionesUno {  
    public static void main(String[] args) {  
  
        int x = 5;  
        int y = 3;  
        x--;  
        x += y;  
        System.out.println(x);  
    }  
}
```



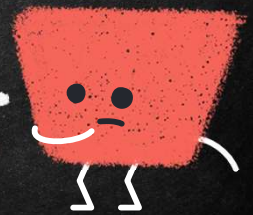
7

## EJERCICIO 2 – INDIVIDUAL

Sin ejecutar el programa...

¿Cual es el valor de y cuando se imprime por pantalla?

```
public class OperacionesDos {  
    public static void main(String[] args) {  
  
        int x = 4;  
        x++;  
        int y = x++ + x; |  
        y += x + x;  
        System.out.println(y);  
    }  
}
```



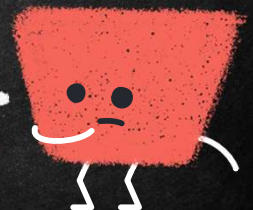
8

## EJERCICIO 3 – GRUPAL

Sin ejecutar el programa...

¿Cual es el valor de y cuando se imprime por pantalla?

```
public class OperacionesTres {  
    public static void main(String[] args) {  
  
        int x = 4;  
        x++;  
        System.out.println(x);  
        int y = x++ + x++;  
        System.out.println(y);  
        y += x + ++x;  
        System.out.println(y); |  
    }  
}
```



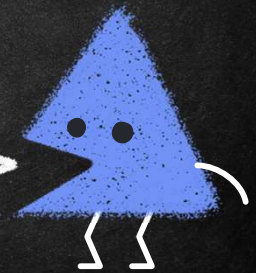


9

## NOTA

```
int n = 10;
int x = ++n; // primer incrementa n, después asigna n a x, n val 11 i x val 11

n = 10;
int x = n++; // primer asigna n a x, después incrementa n, n val 11 i x val 10
```



10

## EJERCICIO 4 – INDIVIDUAL

Sin ejecutar el programa...  
¿Cual es el valor de a y c después de cada operación?

¿¿Y si cambiamos el int por un double??

```
public class OperacionesCuatro {
    public static void main(String[] args) {

        int a = 3;
        int b = -3;
        a = a + 1;
        System.out.println(a);
        a = a + 2 * b;
        System.out.println(a);
        a = a - 2 * b - 2;
        System.out.println(a);
        int c = a / b - 3;
        System.out.println(c);

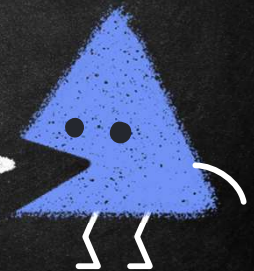
        c = a % b - 3;
        System.out.println(c);
    }
}
```



# BIBLIOTECA MATH

La clase Math ofrece métodos para realizar operaciones matemáticas

abs(n)	valor absoluto de n (el valor positivo)
asin(a)	arco seno del ángulo a
ceil(n)	redondear n al alza (n = 3.33 -> n = 4.0)
floor(n)	redondear n a la baja (n = 3.63 -> n = 3.0)
round(n)	redondeo de n (n = 3.49 -> n = 3.0, n = 3.50 -> n = 4.0)
max(a, b)	máximo de a y b
min(a, b)	mínimo de a y b
pow(a, b)	a elevado a b
random()	valor aleatorio entre 0 i 1 (0<= valor <1)
sqrt(n)	Raíz cuadrada de n (n tiene que ser positivo)



# EJERCICIO MATH 1 – INDIVIDUAL

Sin ejecutar el programa...  
¿Cual es el valor de max y min?

```
public class OperacionesMathUno {
    public static void main(String[] args) {

        int a = 3;
        int b = -3;
        int max = Math.max(a,b);
        System.out.println(max);

        int min = Math.min(a,b);
        System.out.println(min);

    }
}
```

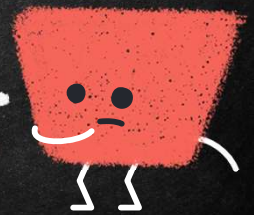




13

## EJERCICIO MATH 2 – INDIVIDUAL

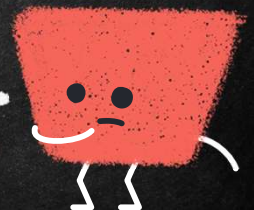
Obtener un número aleatorio entre 0 (incluido) y 1 (excluido)



14

## EJERCICIO MATH 3 – GRUPAL

Obtener un número aleatorio entre 0 (incluido) y 10 (excluido)

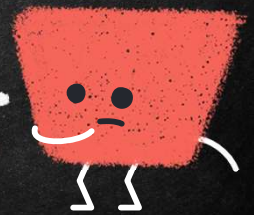




15

## EJERCICIO MATH 4 – GRUPAL

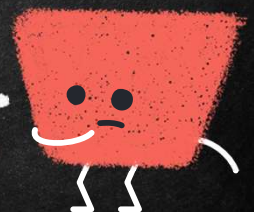
Obtener un número aleatorio entre 1 (incluido) y 10 (incluido)



16

## EJERCICIO MATH 5 – GRUPAL

Obtener un número aleatorio entre dos valores determinados

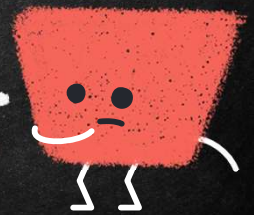




17

## EJERCICIO MATH 6 – INDIVIDUAL

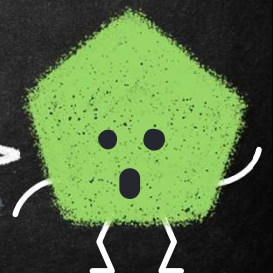
Obtener un número **ENTERO** aleatorio entre dos valores determinados



18

## OPERADORES RELACIONALES

==	Igual
!=	Distinto
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menos o igual que



19

## EJ 1 – OP.REL – INDIVIDUAL

Sin ejecutar el programa...

¿Cual es el valor del boolean que se imprime por pantalla?

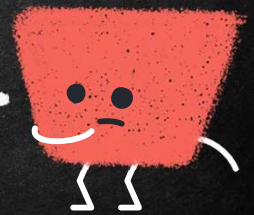
```
public class OperadoresRelacionalesUno {
    public static void main(String[] args) {

        int x = 5;
        int y = 10;

        //Fijemonos además en la concatenación siguiente
        System.out.println("x == y = " + (x == y));

        //Tambien podemos hacer
        boolean resultado = x != y;
        System.out.println("x != y = " + resultado);

        System.out.println("x > y = " + (x > y));
        System.out.println("x < y = " + (x < y));
        System.out.println("x >= y = " + (x >= y));
        System.out.println("x <= y = " + (x <= y));
    }
}
```



20

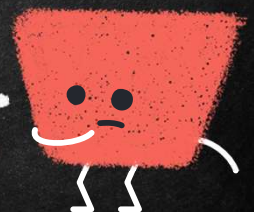
## EJ 2 – OP.REL – INDIVIDUAL

Sin ejecutar el programa...

¿Cual es el valor del boolean que se imprime por pantalla?

```
public class OperadoresRelacionalesDos {
    public static void main(String[] args) {

        System.out.println(4 == 4);
        System.out.println(4 == 5);
        System.out.println(4 == 4.0);
        System.out.println(0 == false);
        System.out.println("asd" == "asd");
        System.out.println(2 == '2');
        System.out.println(2 == "2");
    }
}
```





# OPERADORES LÓGICOS

!	NOT	Unario	Negación
& (&&)	AND	Binario	Conjunción
(  )	OR	Binario	Disyunción
^	XOR	Binario	Disyunción exclusiva

x	y	x && y	x    y	x ^ y	!x
false	false	false	false	false	true
false	true	false	true	true	true
true	false	false	true	true	false
true	true	true	true	false	false

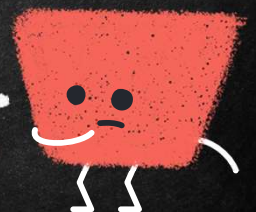


## EJ 3 – OP.REL – INDIVIDUAL

Sin ejecutar en un programa...

Vamos resolver el ejercicio 3 INDIVIDUAL que se publica en aules.

Cuando lo tengamos resuelto mentalmente, pasamos a codificarlo en java para comprobar nuestros resultados.





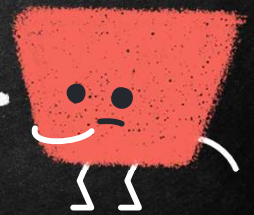
23

## EJ 3 – OP.REL – GRUPAL

Sin ejecutar en un programa...

Vamos resolver el ejercicio 3 GRUPAL que se publica en aules.

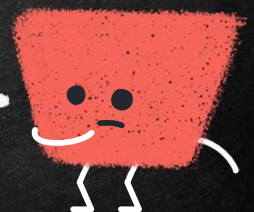
Cuando lo tengamos resuelto mentalmente, pasamos a codificarlo en java para comprobar nuestros resultados.



24

## EJ 4 – OP.REL

Practicar a resolver problemas con programas.  
Disponibles en aules.





25

## OPERANDO CON CADENAS (STRING)

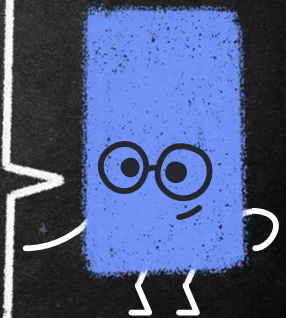
### ASIGNACIÓN

Hasta ahora hemos visto este tipo de asignación en Strings:

```
String txt = "Hola";
```

En realidad String es un Objeto Java, con lo que también podemos verlo declarado así:

```
String txt = new String("Hola");
```



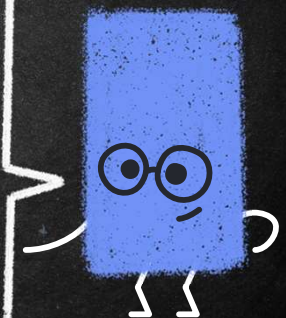
26

## OPERANDO CON CADENAS (STRING)

### CONCATENACIÓN

Podemos juntar dos cadenas simulando una suma, con el operador +

```
//Así  
String txt = "Hola" + "Bea";  
  
//Así  
String txt = "Hola" + " " + "Bea";  
  
//O así  
String nombre = "Bea";  
String txt = "Hola " + nombre;
```



## OPERANDO CON CADENAS (STRING)

Dado: `String saludo = "hola";`

`length()` `int longitudSaludo = saludo.length();`  
`System.out.println(longitudSaludo); // 4`

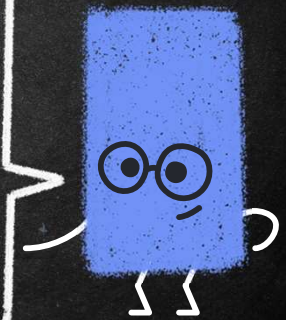
`compareTo(String otroString)`

```
int comparacion = saludo.compareTo("hola");
System.out.println(comparacion); // 0
comparacion = saludo.compareTo("Hola");
System.out.println(comparacion); // 32
```

`lastIndexOf(char a)`

```
int ultimoIndice = saludo.lastIndexOf('a');
System.out.println(ultimoIndice); // 3
ultimoIndice = saludo.lastIndexOf('g');
System.out.println(ultimoIndice); // -1
```

Visitemos String en la API



Fin

