

La sintaxi, com saps, són els elements que formen el llenguatge i com s'han de col·locar per formar un programa. A més de les regles, tindrèm un conjunt de convencions per fer-ho més legible, que no afecten a l'execució del programa.

**\* Importació de biblioteques.**

És un mecanisme per afegir noves funcionalitats al repertori bàsic del llenguatge.

**\* Identificadors**

Un identificador és un nom que distingeix a una variable, un mètode, una funció o una classe. Les regles que segueixen són:

- Han de començar amb una lletra, subratllat(\_) o '\$'
- Pot incloure números, però no hi pot començar
- No pot tenir espais en blanc
- És sensible a majúscules
- No es poden utilitzar paraules reservades

A més hi ha unes convencions per fer la lectura més còmode:

- Posar un nom significatiu al que representa
- Noms de classe comencen amb majúscula (Ex. Rectangle)
- Noms de funció comencen en minúscula (Ex. calcularArea)
- Noms de variable comencen en minúscula (Ex. area)
- Noms de constant en majúscules (Ex. PI)

**\* Comentaris**

Són un text enmig del codi que no es compila. Serveix per explicar (o explicar-nos) que fa aquell codi, per justificar alguna operació o per ajudar a la depuració.

En una sola línia	// Això és un comentari
En diverses línies	/* Això és un comentari que ocupa diverses línies */
Comentari de documentació	/** El comentari de documentació * pot ocupar diferents línies */

**\* Sentències**

Són ordres específiques que realitzen tasques concretes. Totes les sentències acaben en un punt i coma (;).

Exemples:

```
int compt=1;  
System.out.println("Escriu això");
```

**\* Blocs de codi**

Són un grup de sentències que es comporten com a una unitat. Estàn determinats per dues claus ({ }). Poden delimitar una classe, una funció, una estructura de control o agrupar sentències.

Les variables que es declaren dins un bloc de codi només són visibles dins el propi bloc de codi. Ho anirem veient.

Exemple:

```
public class HelloWorld{  
  
    public static void main (String[] args){  
        java.lang.System.out.println("Bones Món!");  
        java.lang.System.out.println("Aquest és el meu món!");  
        java.lang.System.out.println("També HI HA ALTRES MONS!");  
    }  
}
```

### \* Expressions

És tot allò que es pot posar a la dreta de l'operador d'assignació (=).

Exemples:

```
compt=35;  
area=(b*h)/2;  
preu_final=aplicar_IVA(preu);
```

### \* Variables

És un nom que té associada una adreça de memòria de l'equip. Hi ha diferents tipus de variables i cada tipus ocupa un espai diferent de memòria.

Les variables s'han de declarar abans de fer-les servir. Aquesta declaració inclou el tipus de variable i el seu nom. Només llavors li podem donar valor.

Exemples:

```
int compt=1;  
String nom="Jordi";  
double base=24.75;  
boolean acabar=false;
```

### \* Literals

Són un conjunt de caràcters que representen un determinat valor dins el codi font.

Exemples:

```
true  
3.1416  
'A'
```

### \* Caràcters

En Java s'utilitza el joc de caràcters Unicode. S'anoten entre cometes símplex: 'X'.

A més dels caràcters convencionals n'hi ha d'altres, anomenats seqüències d'escapament, que serveixen per representar caràcters de control (no imprimibles).

Retrocés	\b
Retorn de carro	\r
Tabulador	\t
Salt de línia	\n
Cometes	\"
Barra invertida	\\

**\* Constants**

Es tracta d'una "variable" de tipus final. Es declara fora del bloc principal. Una vegada inicialitzada no es podrà canviar el seu valor. Per convenció s'escriu en majúscules.

Exemples:

```
final double PI=3.1416;
final int ALUMNES_MAX=24;
```

**\* Cadenes de caràcters (strings)**

Permeten guardar paraules o frases. S'escriuen entre cometes dobles ("Ximbomba").  
Les veurem en profunditat més tard.

**\* Paraules reservades**

Són les paraules bàsiques del llenguatge, no es poden utilitzar com a identificador. Aquestes paraules es poden agrupar en categories. Per exemple:

Tipus de dades	boolean, float, double, int, char
Sentències condicionals	if, else, switch, case, break, default
Sentències iteratives	for, do, while, continue
Tractament d'excepcions	try, catch, finally, throw
Estructura de dades	class, interface, implements, extends
Modificadors i control	public, private, protected, transient
Altres	super, null, this

**\* Operadors**

Permeten definir operacions per conformar expressions. Poden ser **unaris** (un sol operand) o **binaris** (dos operands).

Aritmètics		
+	Suma	8+6
-	Resta	14-8
*	Producte	6*8
/	Divisió	58/3
%	Mòdul	15%3
++	Autoincrement	++i    i++
--	Autodecrement	--i    i--
Assignació		
=		a='G' b=c=d=13
+=	a=a+b	a+=b
-=	a=a-b	a-=b
*=	a=a*b	a*=b
/=	a=a/b	a/=b

Concatenació		
+	Unió de cadenes	"Cala " + "Figuera"
Relacionals		
==	Igual	
!=	Diferent	
>	Major	
<	Menor	
>=	Major o igual	
<=	Menor o igual	
Logics o booleans		
&	I	
	O	
^	O exclussiva	
&&	I lògica	
	O lògica	
!	No	

#### Precedència d'operadors

Postincrement	a++ a--
Resta unaris	++a --a +a -a ! ~
Multiplicatius	* / %
Additius	+ -
Desplaçament	<< >> >>>
Relacional	< > <= >= instanceof
Igualdat	== !=
I	&
O exclussiva	^
O	
I lògica	&&
O lògica	
Ternari	? :
Assignació	= += -= *= /= %= &= ^=  = <<= >>= >>>=