

Quan programam ens trobam necessàriament amb la necessitat de guardar la informació que haurem de processar. Farem un comentari ràpid sobre alguns elements que fan aquesta funció. La els descriurem en més profunditat així com els necessitem.

També mostrarem ràpidament els operadors que tenim per tractar amb aquestes dades per anar fent boca.

1 Variables

Una variable representa un valor emmagatzemat en una posició de la memòria de l'equip. S'utilitzen per emmagatzemar valors que necessitarem durant el nostre procés.

Cada variable té un nom. Segons el llenguatge que utilitzem aquests noms poden tenir una sintaxi diferent.

Quan posam un valor en una variable s'anomena **assignació**. Fixa't en l'expressió utilitzada: "posam un valor en una variable".

Exemples en diferents llenguatges:

```
R=1
radi:=3;
AX← a5h
```

Ex 1: Comenta si té cap sentit l'expressió d'assignació $X=X+1$.

Hi ha molts llenguatges que abans d'utilitzar una variable s'ha de **declarar**. Declarar una variable indica al programa quin nom utilitzarem i de quin tipus és per tal de reservar l'espai de memòria necessari.

1.1 Tipus

Els tipus de variable indica quina informació contindran. Encara que cada llenguatge pugui tenir les seves característiques pròpies en podem comentar uns quants:

Enter: Guarden nombres enters. Aquests poden ser $[0, 2^n)$ o bé $[2^{n-1}, 2^n)$, on n representa el nombre de bytes que tenen.

Ex: Byte, Integer, LongInt,...

Real: Permeten representar un rang de nombres reals. Es guarden en notació exponencial.

Ex: Float, Double,...

Caràcters: Ocupen un byte i representen un caràcter ASCII.

Cadenes: Seqüències de caràcters.

Booleà: Guarden un valor de veritat (cert o fals). Ocupen un bit.

2 Estructures

Moltes vegades aquestes variables són molt simples. Els llenguatges es permeten construir estructures complexes.

Cadenes: Seqüències de caràcters.

Vectors. Podem crear un conjunt ordenat d'elements i associar-ho a una sola variable. Aquesta tindrà un índex per indicar la posició.

Exemple: Setmana vector[0..6]

```
llavors podem assignar      Setmana[0]="Dilluns"
                               Setmana[1]="Dimarts"...
```

Matrius. Són vectors en diverses dimensions.

Conjunts. Són contenidors d'elements on no intervé cap ordre.

A més d'aquestes estructures que s'han de declarar quan es construeix el programa s'en poden definir d'altres de dinàmiques (llistes, piles, cues, arbres, grafs) ... Però les veurem en una altra ocasió.

3 Tipus abstractes

Els tipus abstractes són un conjunt d'operacions definits sobre unes dades que serveixen de model. Dins aquesta categoria tenim les **classes** i els **objectes**.

Exemple: En la gestió d'una biblioteca podem tenir una classe `llibre` on hi haurà definides les característiques (autor, editorial, pàgines, estat...) i les operacions que s'hi poden fer (manllevar, retornar, llegir, reservar...). Aquesta classe podrà tenir múltiples instàncies (de fet tantes com llibres).

4 Operadors

Cada llenguatge tindrà formes de manipular els tipus d'informacions que hem vist abans. És molt important estar atents a la precedència d'aquests operadors. Passant per al la sintaxi en general ens trobarem (o no) amb:

Operador d'assignació, per posar valors dins les variables.

Operadors matemàtics (+, -, *, /, ^, \, %, ++, --)

Operadors amb cadenes.

Operadors lògics (=, >, <, ≥, ≤, ≠, i, o, no, xor)

Operadors binaris (desplaçament, complement, xor)

A més cada llenguatge disposa d'un repertori més o menys gran de funcions. Moltes vegades si no disposem d'una determinada funció la podem fer nosaltres mateixos.