Un ordinador no fa res més que seguir les instruccions que li donam per tant és necessari saber com donar-les-hi. Això és pot fer des de programes d'aplicació (que ja ha programat algú) o bé fer les nostre pròpies aplicacions.

1 Definició

Un llenguatge de programació és un conjunt de regles i elements que permeten descriure algorismes que seran executats per una màquina.

Aquests llenguatges tenen formalment la mateixa estructura que qualsevol altre:
- Lèxic: un conjunt de paraules permeses
- Sintaxi: les regles de formació d'expressions
- Semàntica: el significat de les expressions i per tant els efectes que tendran.

2 Classificacions

2.1 Segons el nivell d'abstracció

Llenguatge màquina. No hi ha cap abstracció. Estan en codi binari i són propis de cada microprocessador, això condiciona la ***portabilitat***. En darrer terme, qualsevol codi executable està en llenguatge màquina. Quan es visualitza es sol fer en hexadecimal per "facilitar" la lectura.

Cada processador té un repertori d'instruccions. Atenent a la llargària d'aquestes, a la seva quantitat i a la forma d'executar-les podem tenir diferents tipus de màquines.

Ex:

```
48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21
```

Llenguatges de baix nivell. Tot i ser més llegibles que l'anterior es van apropant al llenguatge natural. Com els anteriors són propis de cada processador i cada instrucció del repertori del processador té un equivalent en el llenguatge. Són els anomenats ***assembladors***.

Ex:

```
mov ax, 10 ; posa el valor 10 dins el registre ax
mov bx, 0FA2h ; posa el valor hex. 0FA2 dins el registre bx
add ax, bx ; suma al valor del registre ax el contingut de bx
```

Llenguatges de nivell mig. Tot i presentar característiques del d'alt nivell permeten incursions a baix nivell. Es solen utilitzar per programar sistemes operatius.

Llenguatges d'alt nivell. Són els més propers al llenguatge natural (matemàtic). Cada instrucció a aquest nivell correspon a moltes en llenguatge màquina.

2.2 Segons el paradigma/estil

Imperatius o procedurals. Ens permeten genera seqüències d'ordres que permeten implementar un determinat algorisme. Aquestes ordres van canviant l'estat de la màquina fins a la seva finalització.

Exemple en BASIC:

```
INPUT R
A=3.14*R^2
PRINT A
```

Funcionals o declaratius. Es tracta de descriure quin resultat volem sense descriure cap procediment per arribar-hi.

Exemple en SQL:

```
SELECT nom, llinatges, telefon FROM amics WHERE nom="Jordi" ORDER BY llinatges
```

Orientat a objectes. Aquesta forma pot estar mesclada amb les anteriors. Junten dades i procediments en un sol element anomenat objecte. Aquests s'estructuren de forma jeràrquica i formen classes que comparteixen propietats.

Exemple: En Java podem si tenim definit un objecte cercle amb la propietat àrea podem crear una instància i demanar la seva àrea.

```
new Cercle cercle(r);
System.out.println(cercle.area);
```

2.3 Segons la forma d'execució

Compilats. El programa passa per un procés de traducció que el transforma en llenguatge màquina. D'aquesta manera tenim un fitxer executable independent. Ex: C, Pascal.

Interpretat. Tenen un sistema que va traduint a codi màquina instrucció per instrucció cada vegada que s'executa. Ex: BASIC, JavaScript.

Mixt. El programa original passa per un procés de compilació que genera un codi que s'ha d'interpretar en cada execució. Ex: Java.

2.4 Altres

Visuals. Tant la interfície com el propi llenguatge està orientat d'eines orientades a la presentació en pantalla.

Metallenguatges. No són pròpiament llenguatges, són formes d'organitzar informació i poden incorporar elements amb altres llenguatges. Ex: HTML.

Script. Són llenguatges interpretats que es poden incrustar dins altres elements. Ex. JavaScript.

De propòsit específic.

Ex 1. Fes una taula comparativa dels avantatges i inconvenients de cadascuna de les categories de les classificacions anteriors (4 taules).

Ex 2. Cerca alguns exemples de cadascuna de les categories vistes. Posa'ls en comú amb els companys a classe.