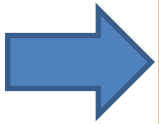
CFGS DAM
Mòdul 5: Entorns de desenvolupament
Prof: Jordi Quesada

INS Joan d'Àustria

Llenguatges de programació

- Existeixen molts llenguatges de programació, alguns especialitzats en algunes tasques i altres més generals.
- Els llenguatges de programació estan classificats segons les seves característiques

Llenguatges de programació



Criteri	Classificació
Segons el nivell d'abstracció	De baix nivell (1a generació) De nivell mig (2a generació) D'alt nivell (3a i 4a generació)
Segons la forma d'execució	Compilats Interpretats Virtualitzats
Segons el paradigma de programació	Funcionals Estructurats Orientats a objectes

Llenguatges de programació

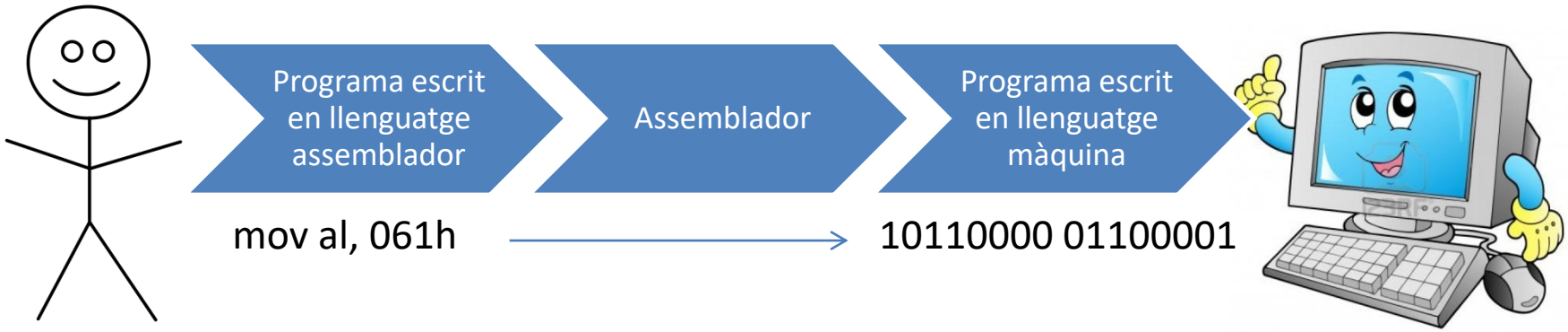
- Segons el nivell d'abstracció
  - De 1a generació:
    - Es programa directament el hardware amb 1 i 0. S'anomena llenguatge màquina
    - És tediós i requereix coneixements profunds
    - Es difícil de mantenir (reutilitzar, compartir, ampliar, corregir...)
    - És específic d'una maquina i no es pot utilitzar en altres

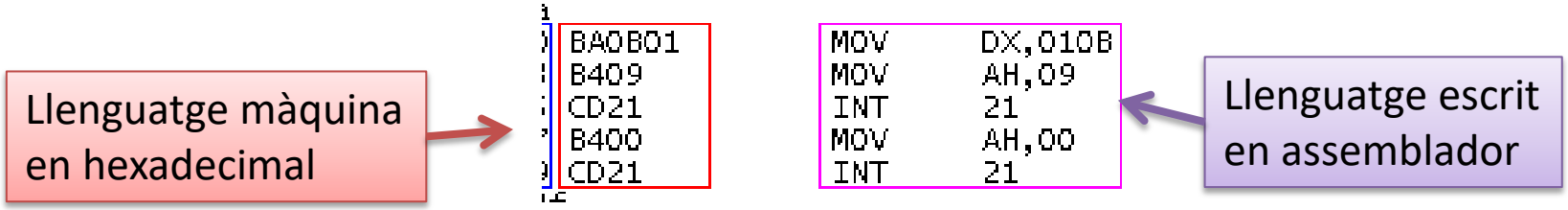
10110000 01100001
Aquesta línia conté una instrucció que mou un valor al registre del processador

Llenguatges de programació

- Segons el nivell d'abstracció
  - De 2a generació:
    - S'anomena llenguatge assemblador
    - S'utilitzen mnemotècnics per tal d'assignar a cada instrucció que és capaç d'executar una màquina una paraula, és a dir, cada paraula correspon a una instrucció
    - Només cal utilitzar una eina anomenada assemblador que fa la traducció
    - S'utilitzen en la programació de drivers i altres dispositus

Llenguatges de programació





https://es.wikipedia.org/wiki/Lenguaje_ensamblador

Llenguatges de programació

- Segons el nivell d'abstracció
  - De 3a generació:
    - N'hi ha molts. S'anomenen llenguatges d'alt nivell
    - El codi és pràcticament anglès i, per tant, són més fàcils d'escriure i mantenir
    - Són independents del hardware
    - Una instrucció en un llenguatge d'alt nivell pot implicar la traducció en moltíssimes instruccions de baix nivell

Llenguatges de programació

- Com treballem amb llenguatges d'alt nivell?

1. El programador fa servir un editor de text i genera un o més fitxers amb les instruccions que vol realitzar. Aquest conjunt de fitxers s'anomena **codi font**.

```
#include <stdio.h>
 int main() {
            printf("Hola mundo");
            return 0;
}
```

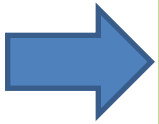
```
public class HolaMundo {
        public static void main(String[] args) {
                System.out.println("Hola Mundo");
        }
 }
```

Llenguatges de programació

- Com treballem amb llenguatges d'alt nivell?

2. Cal traduir aquests fitxers de codi font a llenguatge màquina. Existeixen diferents formes de fer aquesta traducció en funció del llenguatge que utilitzem **i com s'executen al final sobre el sistema operatiu**

Llenguatges de programació



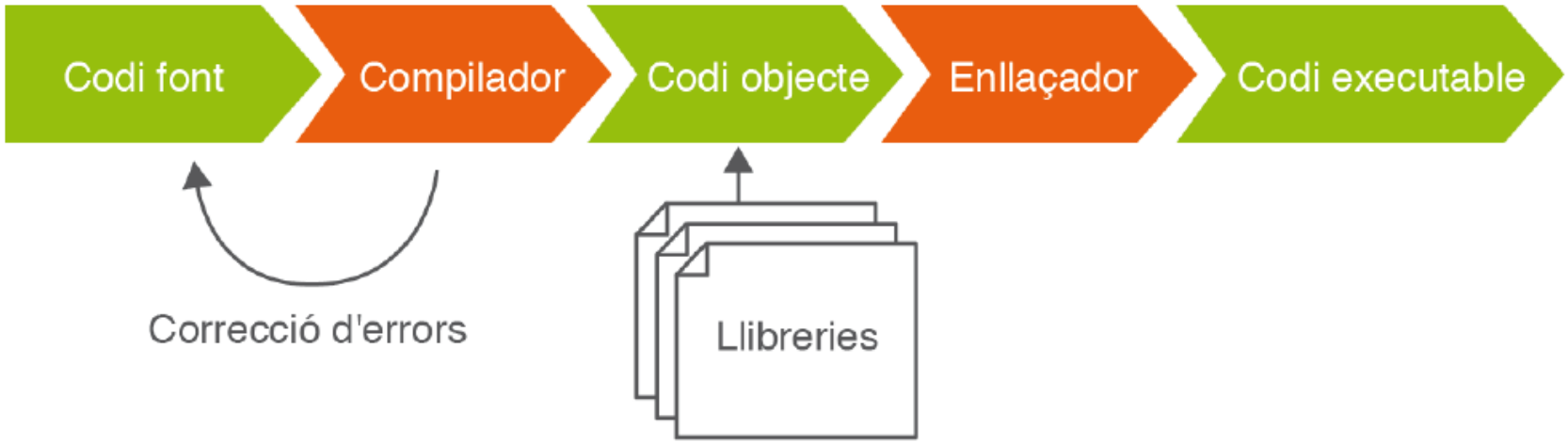
Criteri	Classificació
Segons el nivell d'abstracció	De baix nivell (1a generació) De nivell mig (2a generació) D'alt nivell (3a i 4a generació)
Segons la forma d'execució	Compilats Interpretats Virtualitzats
Segons el paradigma de programació	Funcionals Estructurats Orientats a objectes

Llenguatges de programació

- Segons la forma d'execució
  - Compilats:
    - La compilació consta de diferents processos que consisteixen en verificar que tot estigui correctament escrit i fer una primera traducció a un codi intermig anomenat **codi objecte**.
    - Al codi objecte resultant, se li afegeixen altres codis objectes de llibreries del llenguatge i es fa la traducció final a llenguatge màquina. Aquest pas s'anomena **enllaçador**.

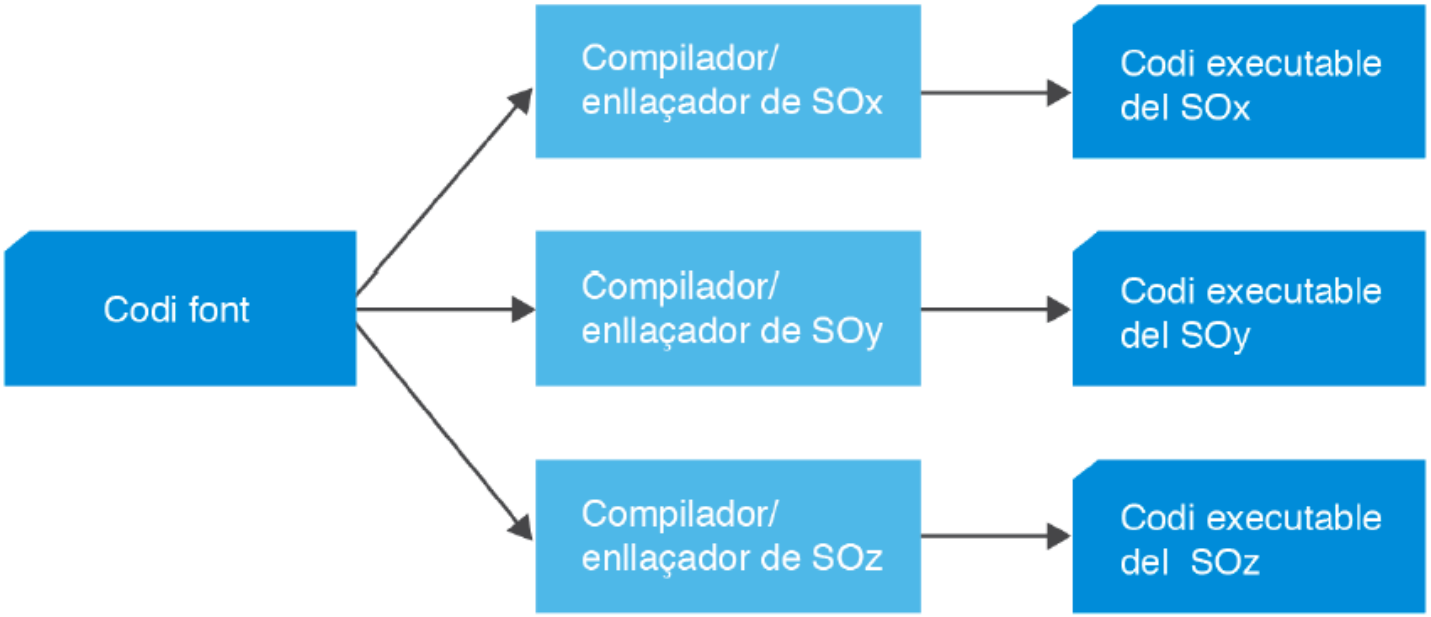
Llenguatges de programació

- Segons la forma d'execució
  - Compilats:
    - Per tant, la traducció a llenguatge màquina es fa abans i la execució del programa es fa després.



Llenguatges de programació

- Segons la forma d'execució
  - Compilats:
    - S'ha de realitzar una compilació/traducció diferent per a cada sistema operatiu, **suposant que existeixi traducció per a cada sistema operatiu.** El codi pot requerir de petites variacions per a cada sistema operatiu.



Llenguatges de programació

- Segons la forma d'execució
  - Compilats:
    - Al final del procés de traducció disposarem d'un arxiu que es podrà executar directament sobre el sistema operatiu

Arxiu: font.c

```
#include <stdio.h>
int main() {
   printf("Hola mundo");
   return 0;
}
```

Compilador en Linux:

```
$ sudo apt-get install build-essential
$ gcc font.c -o executable
```

Llenguatges de programació

- Segons la forma d'execució
  - Interpretats:
    - No hi ha cap procés de traducció.
    - El codi font l'interpreta un altre programa que va llegint línia a línia i les va executant.
    - Si es vol tornar a executar, caldrà tornar a interpretar cadascuna de les instruccions
    - Principalment són els que tenen a veure amb la web (javascript, php, asp...) i altres com python, Perl,...
    - Cada sistema operatiu pot tenir el seu propi interpret

Llenguatges de programació

- Segons la forma d'execució
  - Interpretats:

Exemple llenguatge javascript:

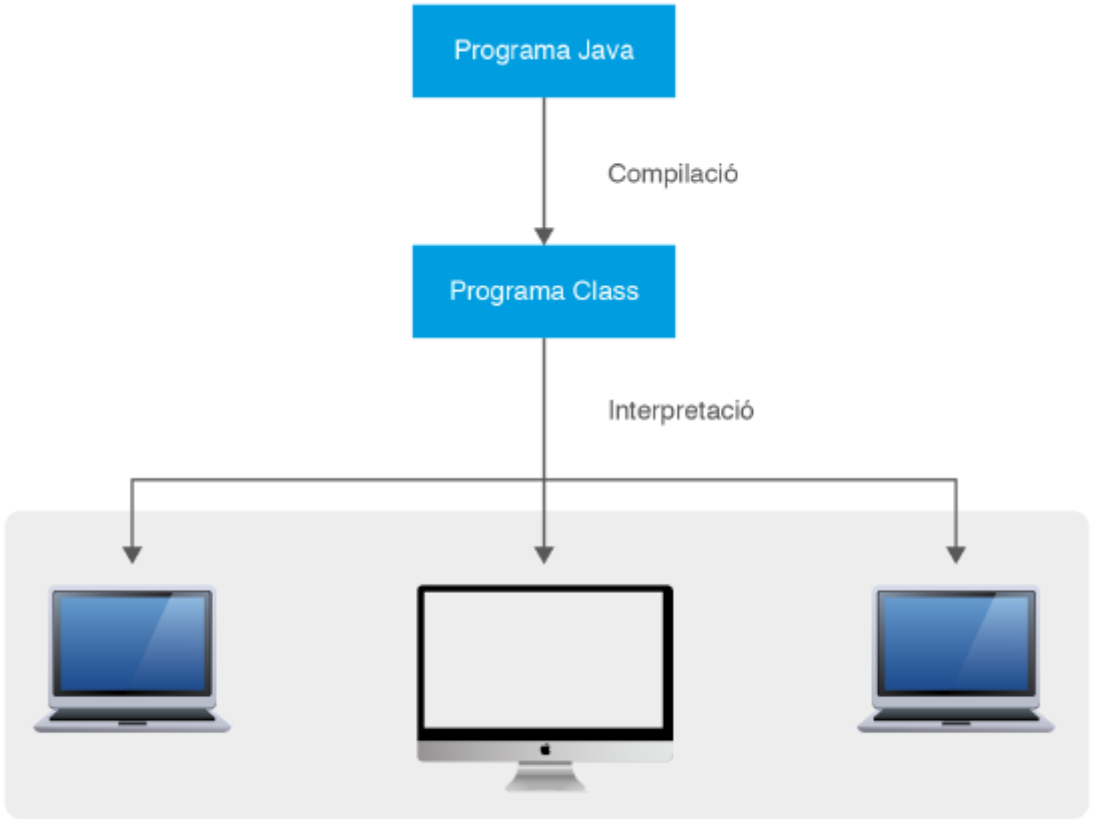
```
<script>
alert('Hola mundo');
</script>
```

Llenguatges de programació

- Segons la forma d'execució
  - Virtualitzats:
    - Podríem considerar que és un híbrid
    - El codi font es compila a un codi intermig que és comú a tots els sistemes
    - Cada sistema operatiu disposa d'un software anomenat màquina virtual que s'encarrega d'agafar el codi intermig i fer la interpretació final.

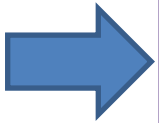
Llenguatges de programació

- Segons la forma d'execució
  - Virtualitzats:



Màquines amb diversos SO

Llenguatges de programació



Criteri	Classificació
Segons el nivell d'abstracció	De baix nivell (1a generació) De nivell mig (2a generació) D'alt nivell (3a i 4a generació)
Segons la forma d'execució	Compilats Interpretats Virtualitzats
Segons el paradigma de programació	Funcionals Estructurats Orientats a objectes

Llenguatges de programació

- Segons el paradigma de programació

  Un paradigma de programació indica una manera particular de construcció de programes. Defineix un conjunt de regles, patrons i estils de programació. Destaquen:

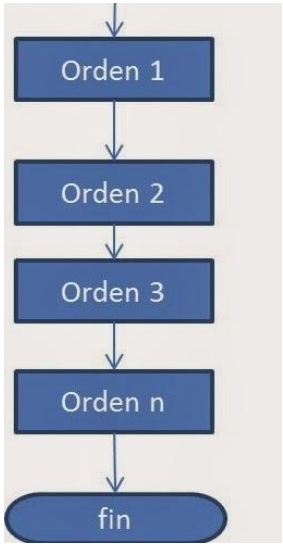
  - Estructurat
  - Orientat a objectes
  - Funcional

Llenguatges de programació

- Segons el paradigma de programació
  - Estructurat:
    - Apareix a finals dels 70's amb el Teorema de Böhm Jacopini
    - Aquest teorema demostra que tot programa de computadora es pot escriure utilitzant únicament tres estructures de control:
      - Estructura seqüencial
      - Estructura condicional
      - Estructura iterativa

Llenguatges de programació

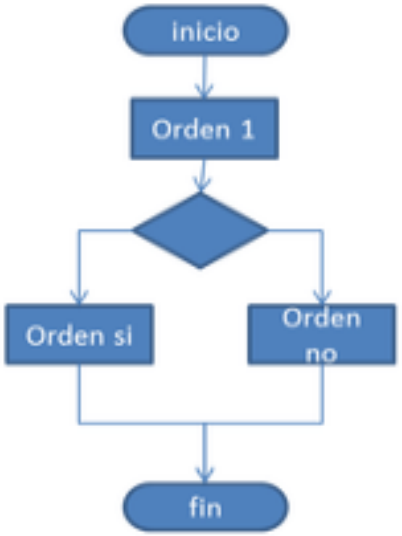
- Segons el paradigma de programació
  - Estructurat:
    - Estructura seqüencial: Les instruccions s'executen una darrera l'altra, és a dir, fins que no acaba la instrucció i no comença la i+1




```
public class Intercambio {
   public static void main(String[] args) {
      int A, B, AUX;
      A = 3;
      B = 5;
      System.out.println("Valores iniciales: A = " + A + "   B = " + B);
      AUX = A;
      A = B;
      B = AUX;
      System.out.println("Valores intercambiados: A = " + A + "   B = " + B);
   }
}
```

Llenguatges de programació

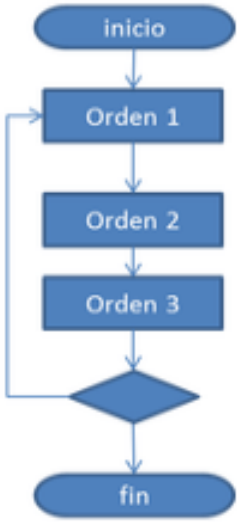
- Segons el paradigma de programació
  - Estructurat:
    - Estructura condicional: Permet que la execució del programa tingui bifurcacions. Segons un criteri o condició lògica només un dels camins s'executarà




```
public class Condicional{
   public static void main(String[] args) {
      int A, B;
      A = 3;
      B = 5;
      if (A>B) {
                System.out.println("A es mayor que B ");
      }
      else {
                System.out.println("A es menor o igual que B");
      }
   }
}
```

Llenguatges de programació

- Segons el paradigma de programació
  - Estructurat:
    - Estructura iterativa: Permet repetir un conjunt d'instruccions fins que es compleixi una determinada condició




```
#include <stdio.h>
 int main() {
            int auxiliar, v1, v2;
            v1=3;
            v2=5;
            /* Intercanvi de variables */
            auxiliar = v1;
            v1 = v2;
            v2 = auxiliar;
            printf( "\n Ara, el valor de v1 es: %d", v1 );
            printf( "\n Ara, el valor de v2 es: %d", v2 );
            return 0;
}
```

Llenguatges de programació

- Segons el paradigma de programació
  - Estructurat:
    - Els principals avantatges de la programació estructura són:
      - El codi font és fàcil de llegir.
      - El manteniment dels programes es senzill.
      - La estructura del programa es senzilla i clara.
    - Exemples de llenguatges estructurats són C, Pascal, Basic, Fortran, etc.

Llenguatges de programació

- Segons el paradigma de programació
  - Orientat a objectes:
    - El seu ús va començar a principis dels 90's.
    - En la programació orientada a objectes, un programa no es composa per un conjunt d'instruccions, sinó per un conjunt d'objectes
    - Un objecte consta d'una estructura de dades anomenades propietats i d'un conjunt d'operacions anomenades mètodes

Llenguatges de programació

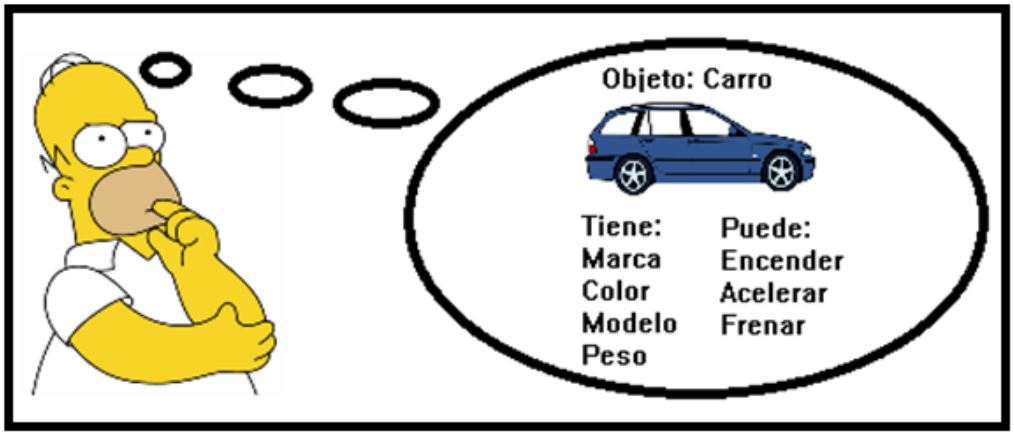
- Segons el paradigma de programació
  - Orientat a objectes:

    Es caracteritza per:

    - Abstracció del món real.
    - Les dades i el procediments que les manipulen són agrupades o encapsulades en objectes.
    - Aquests objectes són una representació directa d'alguna cosa del món real.
    - Integració de 5 conceptes: abstracció, encapsulació, modularitat, jerarquia i polimorfisme.

Llenguatges de programació

- Segons el paradigma de programació
  - Orientat a objectes:




```
public class Carro {
        // Propietats

        private String marca;
        private String modelo;
        private String color;
        private int velocidad;

        // mètodes de clase
        public  void acelerar() {
                this.velocidad++;
        }

        public Carro(String ma, String mo, String co) {
                this.marca = ma;
                this.modelo = mo;
                this.color = co;
                this.velocidad=0;
        }

}
```

Llenguatges de programació

- Segons el paradigma de programació
  - Orientat a objectes:
    - Els principal avantatge és:

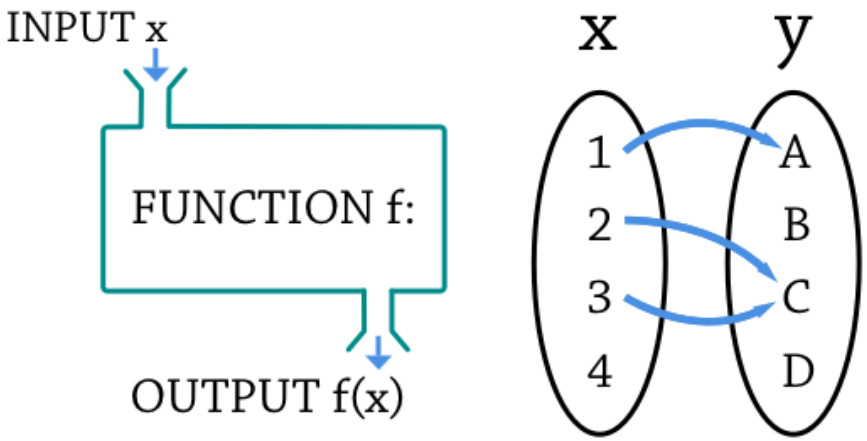
      - Els objectes es poden reutilitzar de manera molt simple

      - És més fàcil detectar un error en el codi d'un objecte que en un programa sencer
    - El principal desavantatge és:

      - No es tant intuïtiva com la estructurada
    - Avui en dia més del 60% de les empreses fan servir aquest paradigma
    - Principals llenguatges de programació: C++, Java, VB.NET, etc.

Llenguatges de programació

- Segons el paradigma de programació
  - Funcionals:
    - Es basa en el concepte matemàtic de funció.
    - No es construeixen en base a instruccions ni objectes sinó en base a funcions
    - La idea és que el resultat d'un càlcul és l'entrada del següent, i així successivament fins que una composició produeixi el resultat desitjat



Llenguatges de programació

- Segons el paradigma de programació
  - Funcionals:
    - Alguns llenguatges funcionals són Lisp, Haskell, Miranda, etc..

```
[1]> (+ 1 2)
3
[2]> (* 1 2)
2
[3]> (cons 1 (cons 2 (cons 3 nil)))
(1 2 3)
[4]> (list 1 2 3)
(1 2 3)
[5]> (first (list 1 2 3))
1
[6]> (rest (list 1 2 3))
(2 3)
[7]> (defun factorial (n)
       (if (> n 0)
          (* n (factorial (- n 1)))
          1))
FACTORIAL
[8]> (factorial 7)
5040
[9]>
```