

Fonaments de la programació

Programació Orientada a Objectes (1)
Conceptes

`{.js}`
JavaScript



UNIÓ EUROPEA
Fons Europeu de
Desenvolupament Regional

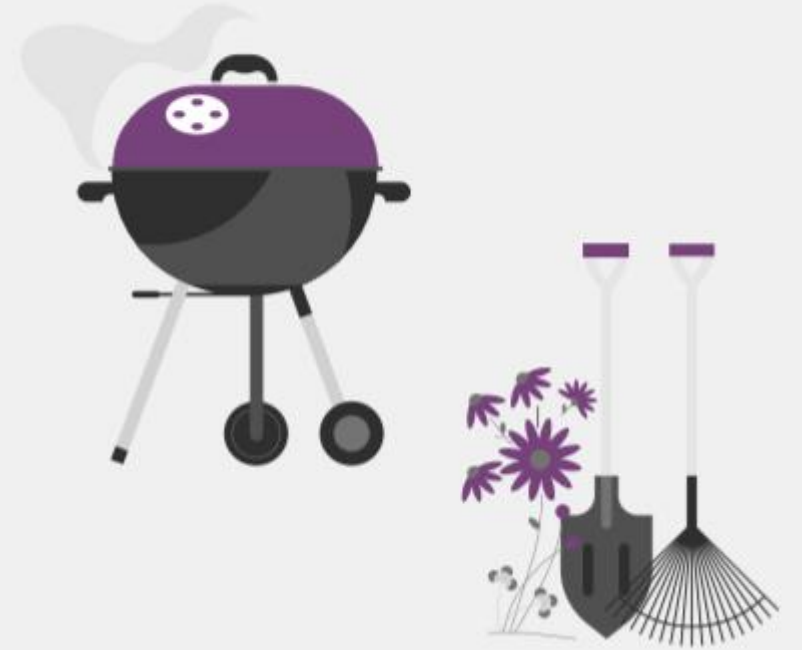
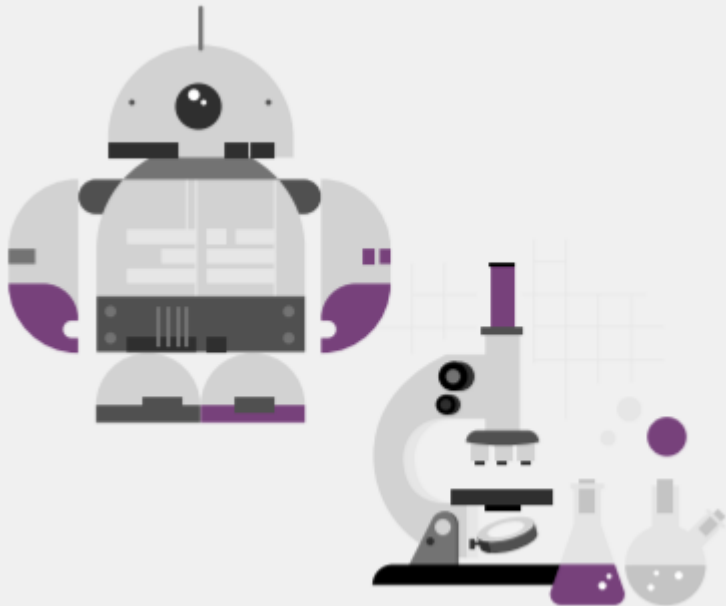
Definició

La Programació Orientada a Objectes (**POO**) és un **paradigma** de programació que fa servir objectes en les seves interaccions, per dissenyar aplicacions i programes informàtics.

Paradigma: una forma de solucionar problemes.

Què és un objecte?

Un **objecte** és una encapsulació abstracta d'informació, juntament amb els procediments per manipular-la.



POO: classes, atributs i mètodes

Un objecte és una **encapsulació** abstracta **d'informació**, juntament amb els procediments per manipular-la.



Un **objecte** conté
atributs (propietats)
que defineixen el seu estat
i
mètodes (operacions)
que indiquen el seu comportament

ATRIBUTS

(característiques, propietats,...)
color, pes, marca, num_cordes...

MÈTODES

(funcions, comportaments...)
tocarNota(), tocar acord(),
afinar(), ajustarClau()

Els objectes es defineixen en **classes...**

Característiques de la POO

- Abstracció
- Encapsulació.
- Polimorfisme.
- Herència.
- Enviament de missatges.
- Reutilització de codi.

Característiques de la POO

- **Abstracció**
- Encapsulació.
- Polimorfisme.
- Herència.
- Enviament de missatges.
- Reutilització de codi.

L'abstracció
consisteix en **aïllar un element**
del seu context
o de la resta dels elements
que l'acompanyen.

Èmfasi en el "què fa?",
no en "com ho fa?"
sent aquesta una característica
de la caixa negra.

Característiques de la POO

- Abstracció
- **Encapsulació.**
- Polimorfisme.
- Herència.
- Enviament de missatges.
- Reutilització de codi.

Es denomina encapsulació a l'**ocultació de l'estat**, és a dir, de les dades membre d'un objecte, de manera que **només es pugui canviar mitjançant les operacions definides per a aquest objecte.**

Característiques de la POO

- Abstracció
- Encapsulació.
- **Polimorfisme.**
- Herència.
- Enviament de missatges.
- Reutilització de codi.

Polimorfisme
és la propietat
que li permet
a un mètode o operació
tenir el mateix nom
en classes diferents
i actuar de manera diferent
en cadascuna d'elles

Característiques de la POO

- Abstracció
- Encapsulació.
- Polimorfisme.
- **Herència.**
- Enviament de missatges.
- Reutilització de codi.

L'herència
és la transmissió del codi
entre unes classes i altres.

Per suportar un mecanisme
d'herència tenim dos tipus de
classes:

la classe mare
i les classes filles.

La classe mare és la que
transmet el seu codi a les
classes filles.

Característiques de la POO

- Abstracció
- Encapsulació.
- Polimorfisme.
- Herència.
- **Enviament de missatges.**
- Reutilització de codi.

El **pas de missatges** defineix que els mètodes es poden dur a terme mitjançant l'enviament d'un missatge d'un procés emissor a un destinatari.

Característiques de la POO

- Abstracció
- Encapsulació.
- Polimorfisme.
- Herència.
- Enviament de missatges.
- **Reutilització de codi.**

La reutilització de codi es refereix al comportament i a les tècniques que garanteixen que una part o la totalitat d'un programa informàtic es pugui emprar en la construcció d'un altre programa. .

POO: tot és un objecte

Els objectes es classifiquen o s'agrupen per:

- Tipus
- Similitud
- Característiques
- Comportaments.



POO: tot és un objecte

Característiques + Accions	
OBJECTE	CLASSE
Guitarra, cello, violí	Instrument de corda
Pep, Laia, Pau	Persona
HP, Sony, Acer, MSI	Ordinador



També coses abstractes:
idees,
equacions,
conceptes

POO: Classes

Les **classes** són plantilles que defineixen de manera genèrica com seran els objectes d'un tipus determinat.
És a dir, els objectes es creen a partir d'una classe determinada.



InstrumentCorda
-String tipus; -String marca;
+InstrumentCorda(String ref, String marca) +tocarNota(); +tocar acord(); +afinar(); +ajustarClau()

guitarra: InstrumentCorda
-String tipus="Guitarra"; -String marca="Tatay";
+InstrumentCorda(String ref, String marca) +tocarNota(); +tocar acord(); +afinar(); +ajustarClau()

Elements d'una classe?

Una classe està composta per dos elements:

- Una llista de Característiques
- Una llista de Comportaments.



O parlant tècnicament,

Atributs + mètodes	
Atributs (emmagatzemen valors) (tipus, marca)	Característiques
Mètodes (executen instruccions) tocarNota, tocarAcord, afinar(), ajustarClau()	Comportaments

Dades
Propietats
Informació

Accions
Instruccions
Operacions

Estructura d'una classe

```
class MyClass {  
    // métodos de clase  
    constructor() { ... }  
    método1() { ... }  
    método2() { ... }  
    método3() { ... }  
    ...  
}
```


Estructura d'una classe

```
class InstrumentCorda {  
  // métodos de clase  
  constructor(tipus, marca) {  
    this.tipus = tipus;  
    this.marca = marca;  
  }  
  tocarNota(nota) {  
    return "La/el " + this.tipus " + està tocant la nota " + nota;  
  }  
  tocarAcord(acord) {  
    return "La/el " + this.tipus " + està tocant l'acord " + acord;  
  }  
}
```



Com utilitzar els objectes?

Creem la instància guitar per a la Classe `InstrumentCorda` i utilitzem els seus mètodes:

```
let guitar = new InstrumentCorda("Guitar", "Yamaha");  
document.getElementById("demo").innerHTML=  
    guitar.tocarNota("Sol");  
alert(guitar.tocarAcord("Do+"));
```

```
class InstrumentCorda {  
    // métodos de clase  
    constructor(tipus, marca) {  
        this.tipus = tipus;  
        this.marca = marca;  
    }  
}
```

Esta página dice

La/el Guitar està tocant l'acord Do+

Aceptar

```
tocarNota(nota) {  
    return "La/el " + this.tipus + " està tocant la nota " + nota;  
}  
tocarAcord(acord) {  
    return "La/el " + this.tipus + " està tocant l'acord " + acord;  
}
```

Codi complet

**{.js}**

JavaScript

```
<html>
  <body>

    <h2>JavaScript POO1</h2>

    <p>Utilitzant la classe InstrumentCorda</p>
    <p id="demo"></p>

    <script>
      class InstrumentCorda {
        // métodos de clase
        constructor(tipus, marca) {
          this.tipus = tipus;
          this.marca = marca;
        }
        tocarNota(nota) {
          return "La/el " + this.tipus + " està tocant la nota " + nota;
        }
        tocarAcord(acord) {
          return "La/el " + this.tipus + " està tocant l'acord " + acord;
        }
      }

      let guitar = new InstrumentCorda("Guitar", "Yamaha");
      document.getElementById("demo").innerHTML = guitar.tocarNota("Sol");
      alert(guitar.tocarAcord("Do+"));
    </script>
  </body>
</html>
```