



# Pre-Pausa

## Esercizi e Recap



# Esercizio 30 min to

## Html Js e CSS ( tail wind / gasp )

Andate a creare una pagina di login

Credenziali pre-inserite



# Recap Php

## Php e le tre regole fondamentali

***PHP (Hypertext Preprocessor) è un linguaggio di scripting lato server utilizzato principalmente per lo sviluppo web.***

***È integrato nell'HTML, eseguito sul server, e genera contenuti dinamici che vengono inviati al browser dell'utente.***

***PHP è noto per la sua facilità d'uso, flessibilità, e compatibilità con vari database, rendendolo una scelta popolare per creare siti web interattivi e applicazioni web.***

# Recap Php

## Php e le tre regole fondamentali

### Ereditarietà

*L'ereditarietà permette a una classe (classe figlia) di ereditare le proprietà e i metodi di un'altra classe (classe madre).*

*Questo favorisce il riuso del codice e la sua modularità funzionale potendo così specializzare responsabilità ed oggetti.*

```
1.<?php  
2.class Animal {  
3.    public $name;  
4.  
5.    public function __construct($name){  
6.        $this->name = $name;  
7.    }  
8.  
9.    public function makeSound(){  
10.       echo "Some generic sound";  
11.    }  
12.  
13.class Dog extends Animal {  
14.    public function makeSound(){  
15.        echo "Bark";  
16.    }  
17.$dog = new Dog("Rex");  
18.echo $dog->name; // Output: Rex  
19.$dog->makeSound(); // Output: Bark  
20.?>
```

# Recap Php

## Php e le tre regole fondamentali

### Incapsulamento

*L'incapsulamento è il principio per cui i dati (proprietà) e i metodi (funzioni) di un oggetto sono racchiusi all'interno della classe e protetti da accessi non autorizzati.*

*Si utilizzano i modificatori di accesso (public, protected, private) per controllare la visibilità.*

```
1.<?php  
2.class Person {  
3.    private $name;  
4.    private $age;  
5.  
6.    public function __construct($name, $age) {  
7.        $this->name = $name;  
8.        $this->age = $age;  
9.    }  
10.   public function getName() {  
11.       return $this->name;  
12.   }  
13.   public function getAge() {  
14.       return $this->age;  
15.   }  
16.  
17. $person = new Person("John", 30);  
18. echo $person->getName(); // Output: John  
19. echo $person->getAge(); // Output: 30  
20.?>
```

# Recap Php

## Php e le tre regole fondamentali

### Polimorfismo

*Il polimorfismo permette a oggetti di classi diverse di essere trattati come oggetti della stessa classe madre.*

*Questo è realizzato attraverso l'uso di metodi che possono avere lo stesso nome ma comportamenti diversi.*

```
1.<?php
2.interface Shape {
3.    public function area();}
4.
5.class Circle implements Shape {
6.    private $radius;
7.
8.    public function __construct($radius){
9.        $this->radius = $radius; }
10.   public function area() {
11.       return pi() * pow($this->radius, 2);}}
12.
13.class Rectangle implements Shape {
14.    private $width;
15.    private $height;
16.    public function __construct($width, $height) {
17.        $this->width = $width;
18.        $this->height = $height; }
19.
20.    public function area() {
21.        return $this->width * $this->height;
22.   }}
23.
24.$shapes = [
25.    new Circle(5),
26.    new Rectangle(4, 6)
27.];
28.foreach ($shapes as $shape) {
29.    echo $shape->area() . "\n";}
30.?>
```

# Esercizio Php

Php e le tre regole fondamentali

Crea un'applicazione PHP che simuli una piccola biblioteca con libri di diversi generi.

## Classe Base: Libro

- **Crea una classe Libro con le seguenti proprietà private:**
  - titolo (stringa) autore (stringa) anno (intero)
- **Implementa un costruttore per inizializzare queste proprietà.**
- **Fornisci metodi pubblici getter per ottenere il valore di ciascuna proprietà.**

## Classi Derivate:

- **Crea due classi derivate da Libro:**
  - Romanzo - Saggio
- **In ciascuna classe derivata, aggiungi una proprietà privata specifica del genere.**
- **Implementa un costruttore che inizializzi sia le proprietà della classe base che quelle specifiche.**

## Polimorfismo:

- **Crea un'interfaccia SchedaBiblioteca con un metodo visualizzaScheda().**
- **Implementa questa interfaccia in entrambe le classi derivate.**
- **Il metodo visualizzaScheda() dovrebbe restituire una stringa con tutte le informazioni del libro.**

## Obiettivo:

- **Crea un array di oggetti di tipo Romanzo e Saggio.**
- **Utilizza un ciclo per chiamare il metodo visualizzaScheda() su ciascun oggetto e stampa le informazioni.**

# Recap Vue.JS

Vue.js

***Vue.js è un framework JavaScript progressivo per la costruzione di interfacce utente.***

***È progettato per essere incrementale, permettendo di integrare gradualmente la sua funzionalità in progetti esistenti o di usarlo per costruire applicazioni web complesse da zero.***

***Ha nei suoi scopi funzionali Tre Caratteristiche Principali di utilizzo;***

# Recap Vue.JS

## Vue.js

### **Reattività**

- **Vue.js utilizza un sistema reattivo che monitora le modifiche ai dati e aggiorna automaticamente il DOM, garantendo che l'interfaccia utente sia sempre sincronizzata con lo stato dell'applicazione.**

### **Componenti**

- **L'architettura basata sui componenti di Vue.js consente di creare interfacce utente modulari e riutilizzabili, suddividendo l'applicazione in piccoli pezzi indipendenti.**

### **Integrazione Facilitata**

- **Vue.js può essere facilmente integrato in progetti esistenti con altre librerie o framework, e può essere utilizzato sia per arricchire funzionalità specifiche che per sviluppare intere Single Page Applications (SPA).**

# Saluti finali ed esercizi

Per php ci sono già gli es chiamati

**EsPHP(SAVERIO)** e c'è una cartella EsPHP

Per vue.js c'è una cartella EsVue

Per Js c'è una cartella EsJS