

A stylized illustration of a desk setup. On the left, a desk lamp with a brown shade is lit. In the center, a large computer monitor displays a web browser interface with a 'W' logo and some text. On the desk in front of the monitor is a keyboard, a mouse, a pair of glasses, a small red object, and a white cup of coffee with steam rising from it. To the right of the monitor, there's a small blue object and a red object. The background shows a window with a plant and some architectural details.

SAE Lectures

WebDev | ARCADE

Creating a browser game

timor kodal

A person with a backpack is seen from behind, looking out over a vast, hazy mountain landscape. The person is wearing a dark jacket and has a large, rolled-up mat strapped to their back. The landscape features rolling hills, a dirt road, and some small buildings in the distance. A blue rectangular box is overlaid on the upper part of the image, containing text.

Webdev | PHP.CMS OOP Recap

PHP | OOP.1
timor kodal

OOP Recap | Klassen und Objekte

Objektorientierung in der Programmierung kennzeichnet die **Organisation** von Programmcode in **Klassen und Objekten**.

Eine **Klasse** entspricht einem abstrakten Konzept, das als Blaupause für die Erstellung realer, von ihr abgeleiteter **Objekte** dient. Bei einem Bauprojekt entspricht die Klasse somit dem Bauplan und ein Objekt dem fertigen Gebäude.

Eine Klasse entspricht zugleich dem **Datentyp eines Objekts**.

Vergleicht ihr dies mit der realen Welt, so wäre das Konzept eines Tieres eine Klasse. Tiere haben Gemeinsamkeiten, verfügen z.B. meist über Gliedmaßen und betreiben Stoffwechsel. Ein konkretes Tier (eine Hauskatze) würde dann einem Objekt der Klasse Tier entsprechen.

OOP Recap | Eigenschaften und Methoden

Eine Klasse definiert in der Regel charakteristische Attribute als **Eigenschaften**.

Üblicherweise stellt sie darüber hinaus Möglichkeiten zu **Interaktion** und **Tätigkeitsausübung** bereit und stellt diese allen von ihr abgeleiteten Objekten als **Methoden** zur Verfügung.

Zu den **Eigenschaften** der Klasse Tier zählen **Farbe** und **Beschaffenheit des Fells**

Typische **Methoden** eines Tieres umfassen Tätigkeiten wie z.B. **Fortbewegung**, **Klettern**, **Fressen**, **Schlafen** und **Reproduktion**.

OOP Recap | Vererbung (is-a relationship)

Klassen können Eigenschaften und Methoden von übergeordneten Klassen ableiten (**erben**). Die Attribute der abgeleiteten Klasse können unverändert übernommen oder aber auch verändert und erweitert werden.

Auf diese Weise können gemeinsame Attribute von verschiedenartigen Tieren in **hierarchischen Strukturen** (**Vererbungsbäume**) organisiert werden.

Ein **Hai** betreibt Kiemenatmung, die in der Klasse **Fisch** spezifiziert ist.
Da eine **Katze** von ihrer Mutter gesäugt wird, zählt sie zur Klasse **Säugetier**.
Ein **Hai** und eine **Katze** teilen zusätzlich jedoch auch gemeinsame Attribute, die in der für beide Klassen übergeordneten Klasse **Wirbeltier** hinterlegt sind.

OOP Recap | Komposition (has-a relationship)

Alternativ zur Vererbung können Klassen auch durch **Komposition** Attribute anderer, nicht übergeordneter Klassen erhalten, wodurch sich eine freie Kombination verschiedener Eigenschaften und Methoden erreichen lässt.

In PHP lässt sich Komposition durch **Interfaces** und **Traits** realisieren.

Im Unterschied zur Vererbung ist die Komposition **flexibler** und bietet sich immer dann an, wenn eine streng hierarchische Strukturierung nicht zielführend erscheint.

Schmetterlinge zählen zu den Gliederfüßern, Vögel zu den Wirbeltieren. Somit gehören beide Tierarten unterschiedlichen Tierstämmen an. Dennoch besitzen beide die Möglichkeit des Flugs. Für derartige Fälle bietet sich das Konzept der Komposition an, um einzelne analoge Attribute verschiedenen Klassen zuzuordnen.

OOP Recap | Sichtbarkeit von Attributen

Für Eigenschaften (Properties) und Methoden können folgende Sichtbarkeiten eingestellt werden:

- **public**: immer sichtbar
- **protected**: für die Klasse selbst und alle erbenden Klassen sichtbar
- **private**: nur für die Klasse selbst sichtbar

```
class Animal {  
    public $state = 'STATE';  
    public function getState() {  
        return $this->state;  
    }  
}  
$animal = new Animal();  
echo $animal->getState();  
# outputs: 'STATE'
```

OOP Recap | Statische Variablen und Methoden

In der Regel müssen Klassen instanziiert werden, damit auf Eigenschaften und Methoden zugegriffen werden kann.

Es gibt aber auch die Möglichkeit, Variablen und Methoden als **static** zu deklarieren. Dann kann über den Gültigkeits-Operator **::** auch über den Klassennamen direkt auf sie zugegriffen werden.

```
class Animal {  
    public static $state = 'STATE';  
    public static function getState() {  
        return self::$state;  
    }  
}  
  
echo Animal::getState();  
# outputs: 'STATE'
```

OOP Recap | Vertiefende Lektüre

- Objektorientierte Programmierung
- Objektorientiertes Design



Webdev | ARCADE Setup

IDE Setup
timor kodal

Abbildung 1 (Hintergrund): Female Dev in a modern office setting (Kodal, 2023)

WebDev | ARCADE

Installation und Setup

Benötigte Programme installieren

GIT

Versionierung

git-scm.com/downloads

Docker

Virtualisierung

docker.com/products/docker-desktop

Visual Studio Code

Entwicklungsumgebung

code.visualstudio.com/download

WebDev | ARCADE

Installation und Setup

Optionale Programme installieren

Sublime Text

Textbearbeitung

sublimetext.com/download

TotalCommander (PC)

Dateiverwaltung

ghisler.com/download.htm

Cyberduck

Dateiübertragung

cyberduck.io/download

Paketmanager

homebrew (MAC)

winget (Windows)

apt (Ubuntu)

Installation und Setup

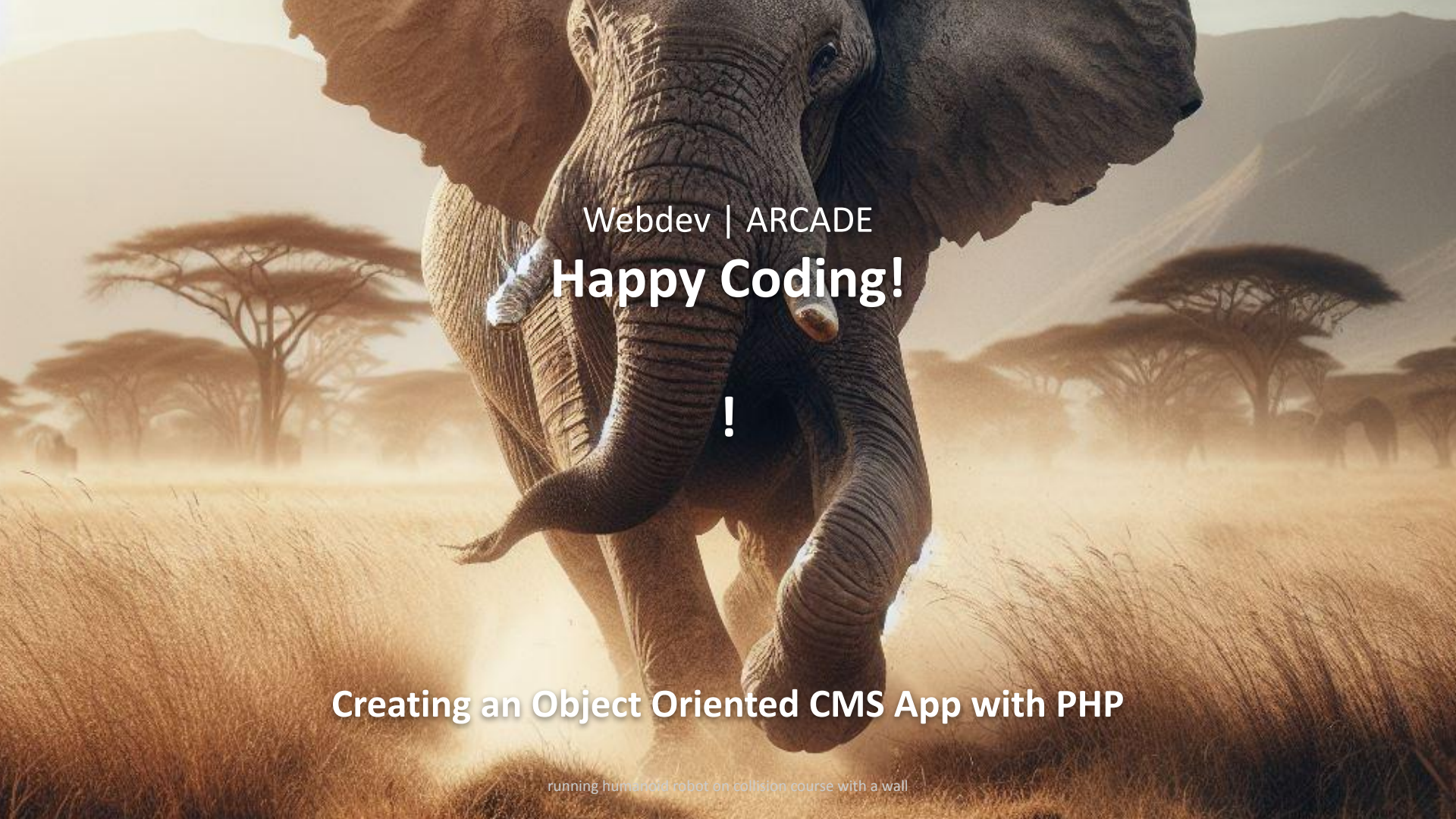
Laufzeitumgebung erstellen und IDE Installieren

DOCKER

- **Docker** von der offiziellen Docker-Seite herunterladen
<https://www.docker.com/products/docker-desktop>
- Installationsanweisungen befolgen

VISUAL STUDIO CODE (Entwicklungsumgebung)

- **Visual Studio Code** herunterladen
<https://code.visualstudio.com/download>
- Installationsanweisungen befolgen
- **LiveShare - Extension installieren** (optional)

A large elephant is running towards the viewer through a savanna landscape. The scene is bathed in the warm, golden light of a sunset or sunrise, with long shadows and a hazy atmosphere. The elephant's trunk is lowered, and its ears are spread wide. In the background, there are acacia trees and distant mountains.

Webdev | ARCADE
Happy Coding!
!

Creating an Object Oriented CMS App with PHP

running humanoid robot on collision course with a wall

Webdev | ARCADE
Happy Coding!

Developing a CMS App with PHP
timor kodal

Abbildung 1 (Hintergrund): Female Dev in a modern office setting (Kodal, 2023)

ARCADE Example

Beispielanwendung auf Basis von OOP

Entwicklungsumgebung einrichten und Anwendung erstellen

- **Repository** klonen
<https://github.com/teamore/arcade-spaceship>
- Quellcode sichten ([LiveShare-Link](#))
- Zum Projektverzeichnis wechseln, Container bauen und starten
`docker compose up`
- Beispieldatei im Browser aufrufen
<http://localhost:3000>

A herd of elephants is walking away from the viewer into a savanna landscape. The lead elephant is in the center, with its back to the camera. Other elephants are visible in the line behind it, and the background shows rolling hills and acacia trees under a cloudy sky.

SAE Lectures WebDev | ARCADE

DANKE und BIS BALD

WebDev | ARCADE
timor kodal