

# FEVAL

- Feedback

# Vorlesung 10

- Komplexe Datentypen: Arrays und Objekte

# Zusammenfassung

- Variablen, Datentypen und Operatoren

# Variablen

- Deklaration

# Deklaration – in TypeScript

var	bezeichnung	:	typ
-----	-------------	---	-----



**Schlüsselwort zur Deklaration einer Variable**

Ähnliche Schlüsselwörter: let, const

```
/* Codebeispiel */
```

```
var name : string;  
var age : number;  
var likesFish : boolean;  
var unsureWhichType: any;
```

# Deklaration – in TypeScript

var    bezeichnung    :    typ

```
/* Codebeispiel */
```

```
var name : string;  
var age : number;  
var likesFish : boolean;  
var unsureWhichType: any;
```

**Freie Bezeichnung der Variablen durch den Programmierer  
mit folgenden syntaktischen Einschränkungen**

- Ausschließlich Verwendung von Ziffern (0-9), Buchstaben (a-z, A-Z), sowie zwei weitere Zeichen (Unterstrich und \$), die aber besser nicht verwendet werden
- dürfen nicht mit einer Ziffer beginnen
- dürfen keine Leerzeichen enthalten und auch keinen Bindestrich
- keine Verwendung eines Schlüsselworts aus TypeScript/JavaScript (z.B. string, for, while, ...)

# Deklaration – in TypeScript

var	bezeichnung	:	typ
-----	-------------	---	-----

**Syntaktischer  
„Zuordnungsoperator“  
für den Datentyp**

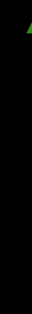


```
/* Codebeispiel */
```

```
var name : string;  
var age : number;  
var likesFish : boolean;  
var unsureWhichType: any;
```

# Deklaration – in TypeScript

var bezeichnung : typ



**Gültiger TypeScript-Datentyp,**  
z.B. string, number, boolean  
oder ein definiertes Objekt

```
/* Codebeispiel */
```

```
var name : string;  
var age : number;  
var likesFish : boolean;  
var unsureWhichType: any;
```



# Variablen

- Wertzuweisung

# Zuweisung

`var bezeichnung : typ`

`=`

`Wert`

**Zuweisungsoperator**

**Gültiger (!) Wert**

```
/* Codebeispiel */
```

```
var name : string = "Pingu";  
var age : number = 2;  
var likesFish : boolean = true;  
var unsureWhichType: any;
```

# Konstanten

# Konstanten

```
/* Codebeispiel */  
const placeOfBirth: string = "Südpol";
```

**Konstanten-Deklaration**  
**Schlüsselwort „const“**

**Besonderheiten im Vergleich zur Variable**

- Konstanten benötigen eine initiale Zuweisung (Initialisierung).
- Konstanten können nicht mehr verändert werden

# Primitive Datentypen

- Number, String, Boolean

# Primitive Datentypen

string    `var name : string = "Pingu";`

Zeichenkette

number    `var age: number = 2;`  
          `var age: number = 2.5;`

Zahl ganzzahlig bzw.  
Fließkommawert

boolean    `var likesFish : boolean = true;`

Bool'sche Aussagenlogik

any        `var ID : any;`

Ohne feste Zuordnung eines  
Datentyps

# Operatoren

- Verarbeitung von Daten

# Zeichenketten Operatoren

Verbindung

+

"Hallo " + "Welt"

"Hallo Welt"

Verbindung string und  
Number

"Hallo Welt " + 1

"Hallo Welt 1"



# Mathematische Operatoren

Addition	+	17+3	20
Subtraktion	-	10-3	7
Multiplikation	*	3*5	15
Division	/	15/3	5
Divisionsrest / Modulo	%	12/5	2
Incrementor	++	<code>var i=10; i++;</code>	11
Decrementor	--	<code>var i=10; i--;</code>	9
Zuweisungsoperator	<code>+= -= *= /=</code>	<code>var i=10; i+=5;</code>	15

# Komplexe Datentypen

- Arrays und Objekte

# Arrays

- Deutsch: „Feld“

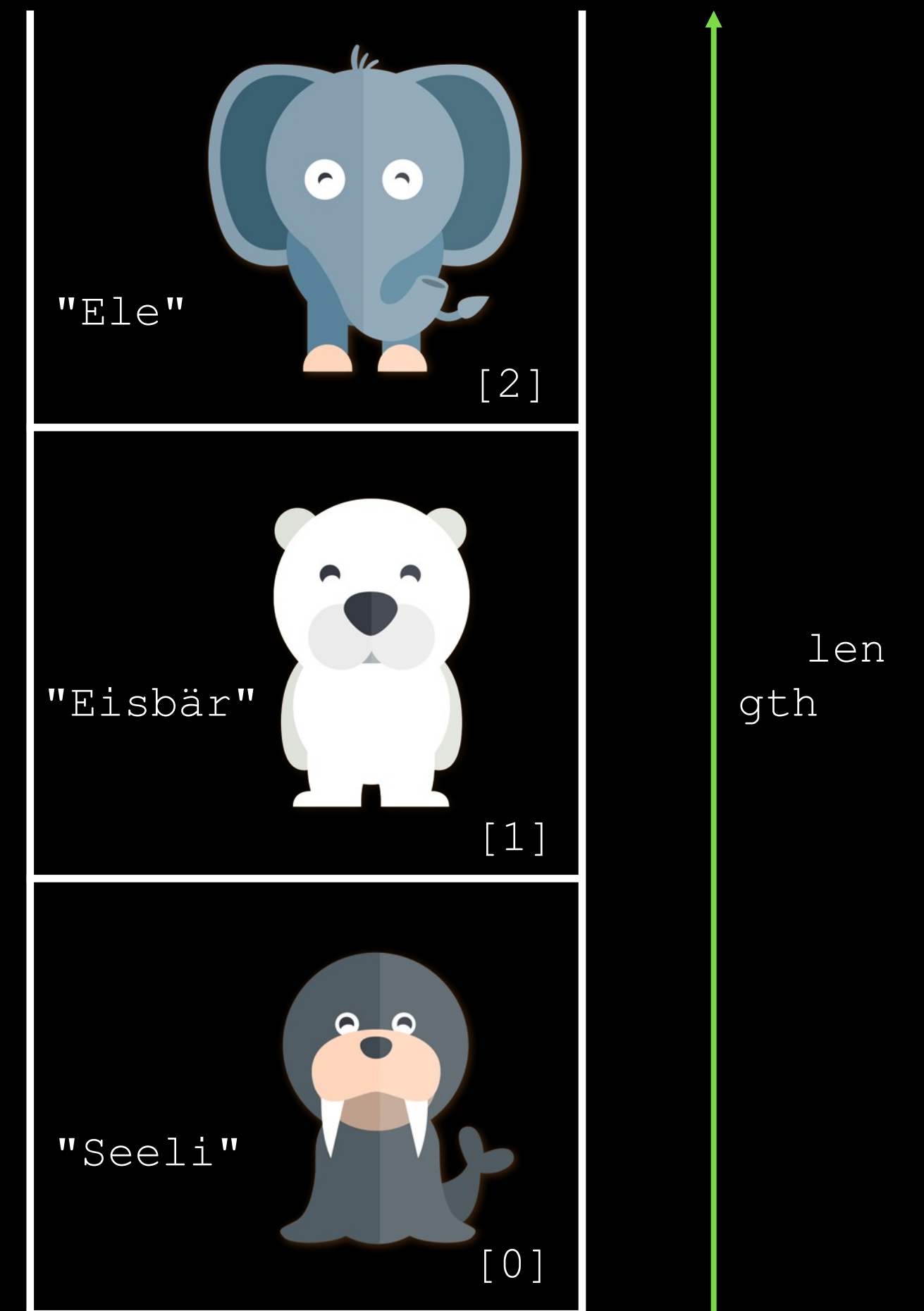
# Arrays

```
/* Codebeispiel */

// Initialisierung Array
var nameFriends: string [] = ["Seeli", "Eisbär","Ele"];

// Zugriff auf einzelne Elemente des Arrays
console.log ( nameFriends[0] ); // Seeli
console.log ( nameFriends[1] ); // Eisbär
console.log ( nameFriends[2] ); // Ele

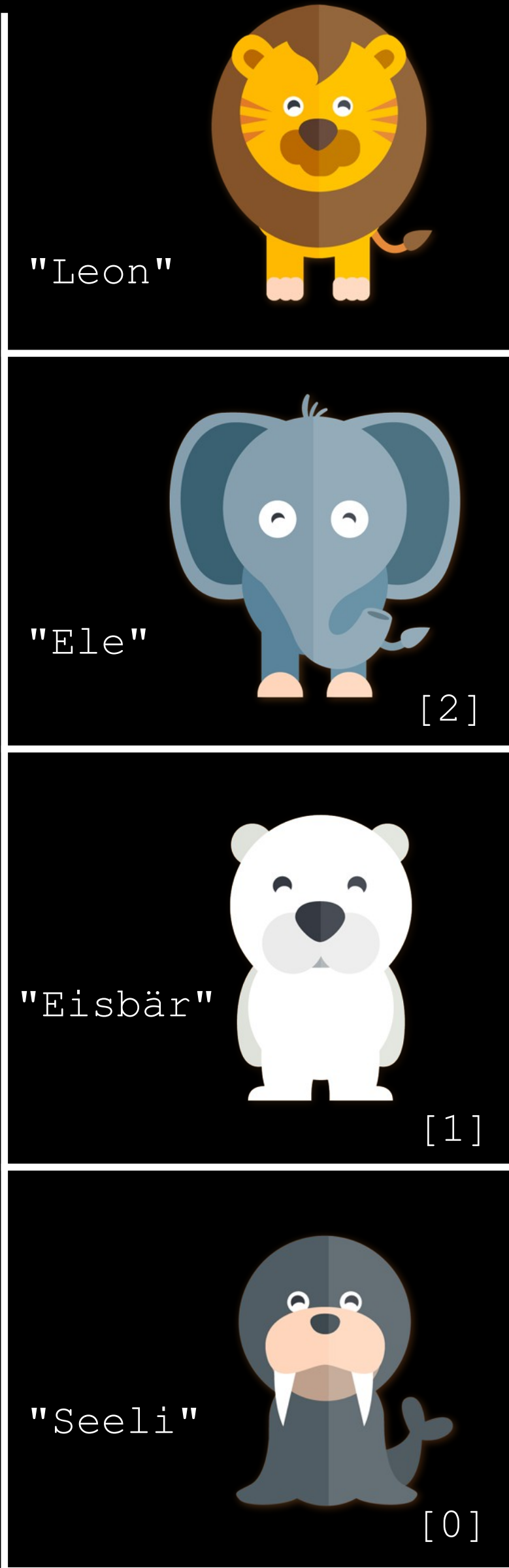
// Ausgabe Anzahl Elemente („Länge“) des Arrays
console.log ( nameFriends.length );
```



# Arrays

```
/* Codebeispiel */  
// Array Element hinzufügen  
nameFriends.push("Leon");  
console.log (nameFriends,nameFriends.length);
```

push("Leon")  
);

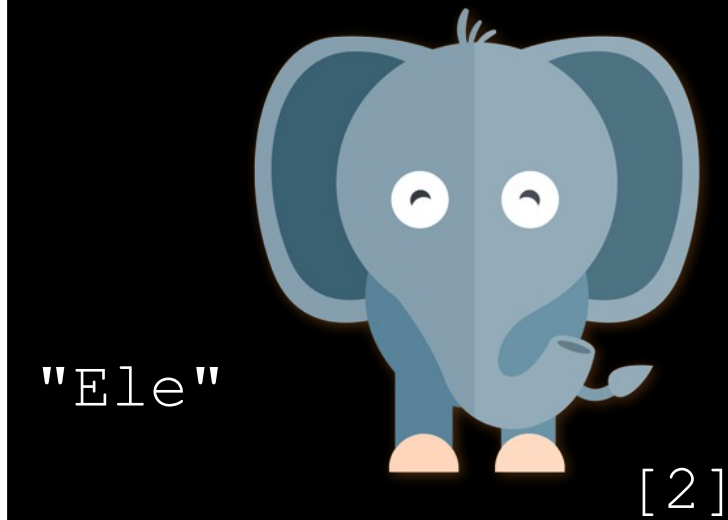
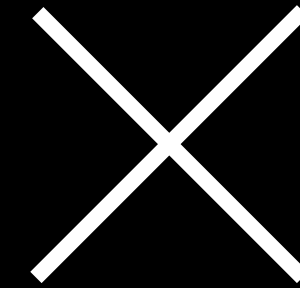


# Arrays

```
/* Codebeispiel */  
// Array Element hinzufügen  
nameFriends.push("Leon");  
console.log (nameFriends,nameFriends.length);  
  
// Oberstes Element des Arrays löschen  
nameFriends.pop();  
console.log (nameFriends,nameFriends.length);
```



pop(  
);

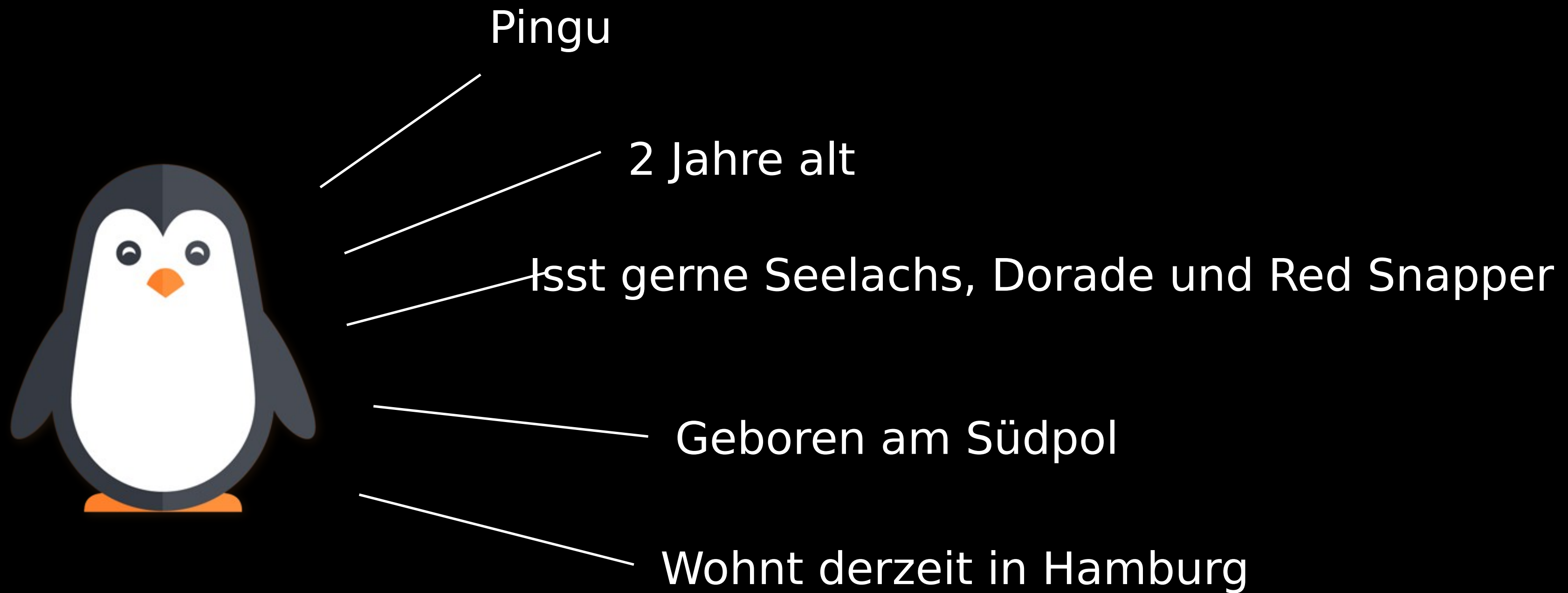




# Objekte



# Objekte



# Objekte



Leon

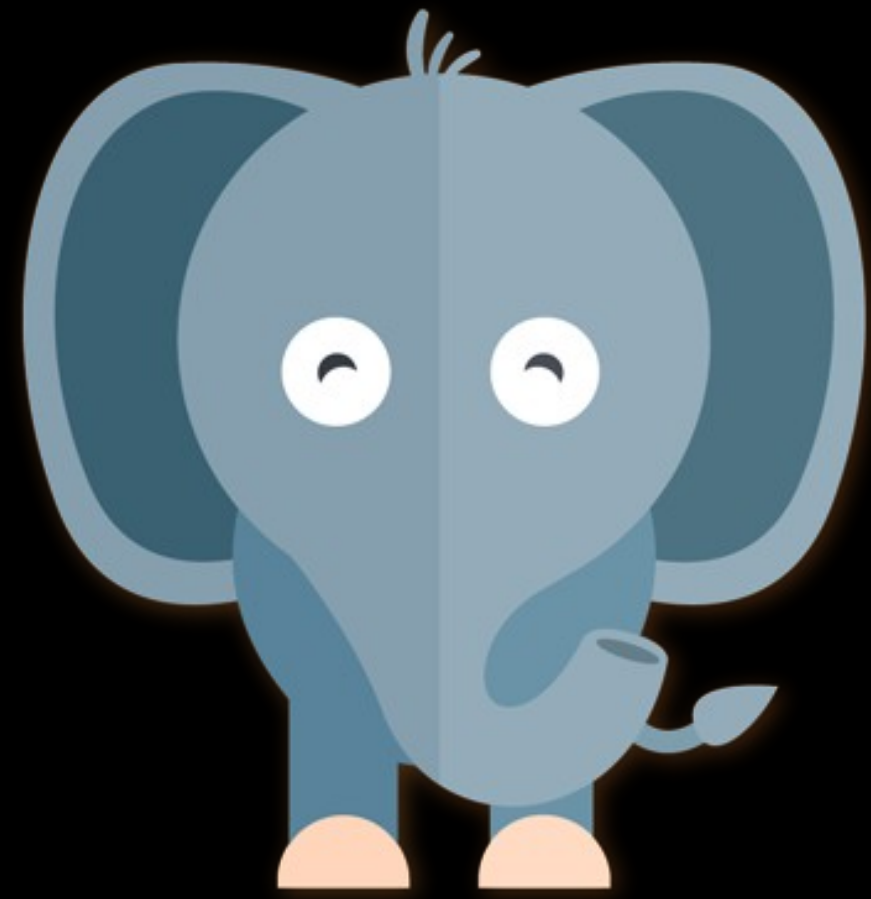
14 Jahre alt

Isst gerne Antilopen, Gazellen, Zebras und Gnus

Geboren in Afrika

Wohnt derzeit in Stuttgart

# Objekte



Ele

28 Jahre alt

Isst gerne Gras, Früchte, Zweige und Rinde

Geboren in Afrika

Wohnt derzeit in München

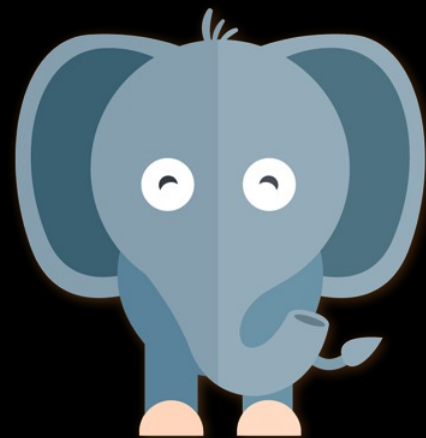
# Objekte



```
var name_1: string = "Pingu";  
var age_1: number = 2;  
var food_1: string [] = ["Seelachs", "Dorade", "Red Snapper"];  
const placeOfBirth1: string = "Südpol";;  
var currentLocation1: string = "Hamburg"=
```

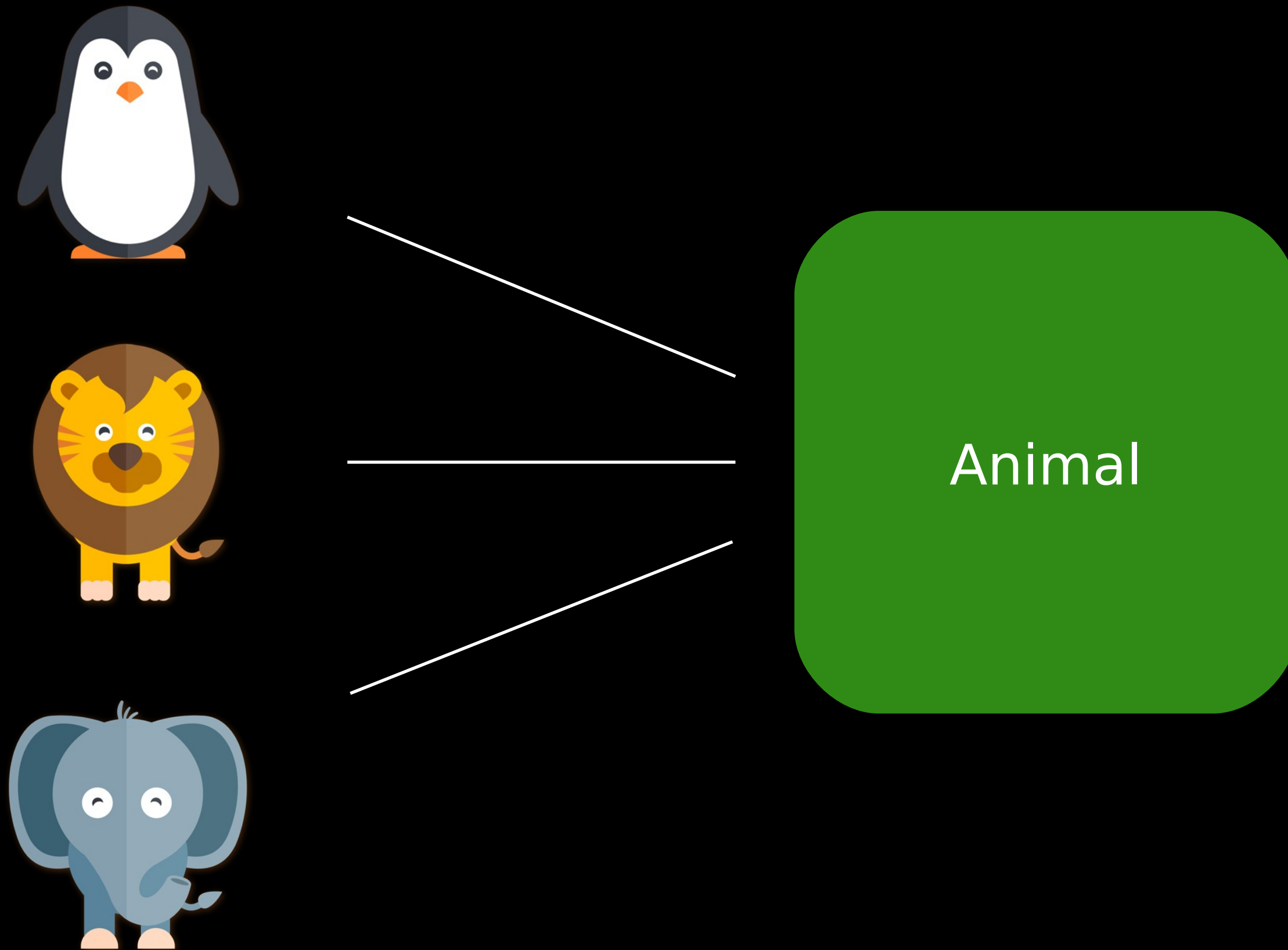


```
var name_2: string = "Leon";  
var age_2: number = 12;  
var food_2: string [] = ["Antilope", "Gazelle", "Zebra", "Gnu"];  
const placeOfBirth_2: string = "Afrika";;  
var currentLocation_2: string = "Stuttgart"=
```

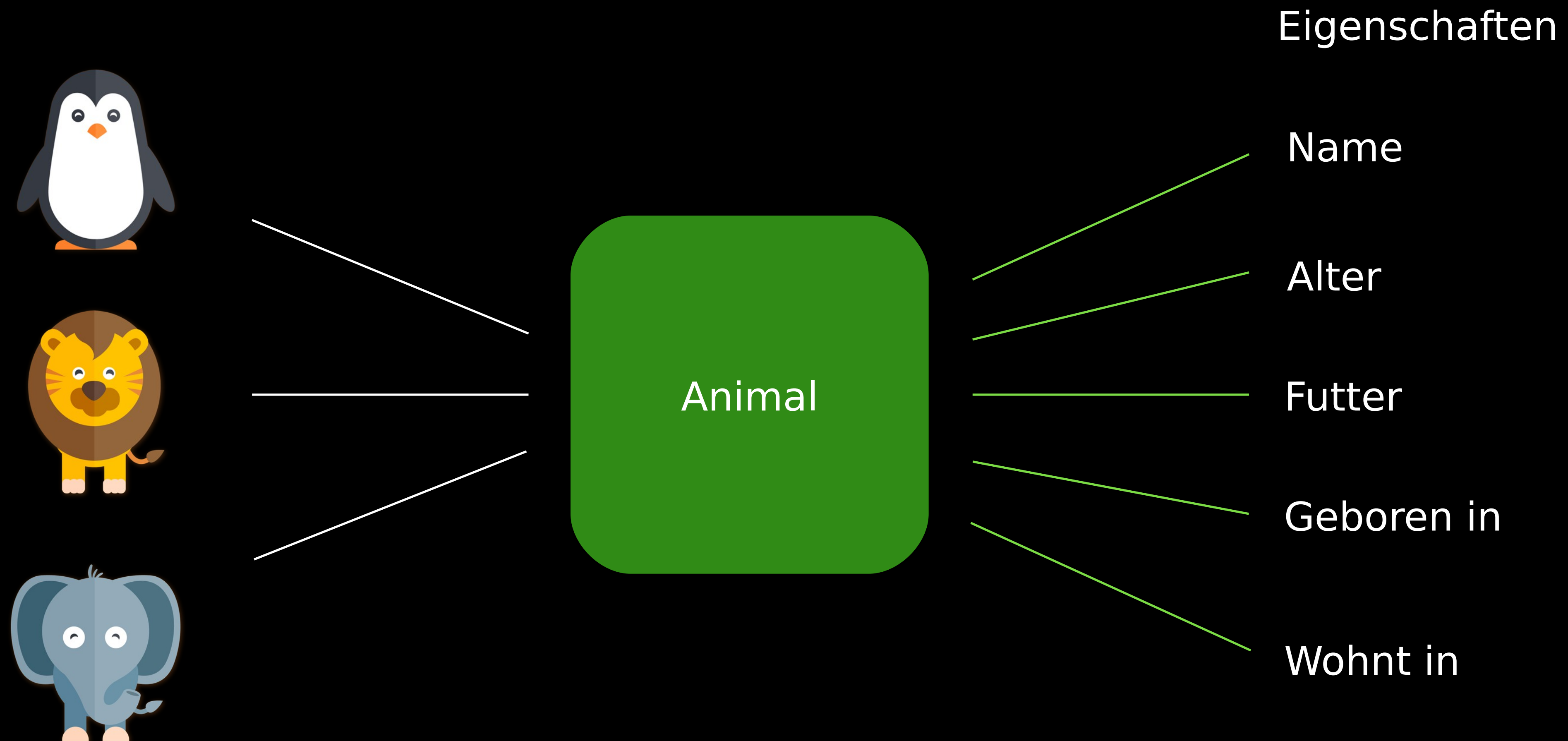


```
var name_3: string = "Ele";  
var age_3: number = 28;  
var food_3: string [] = ["Gras", "Früchte", "Zweige", "Rinde"];  
const placeOfBirth_3: string = "Afrika";;  
var currentLocation_3: string = "München"=
```

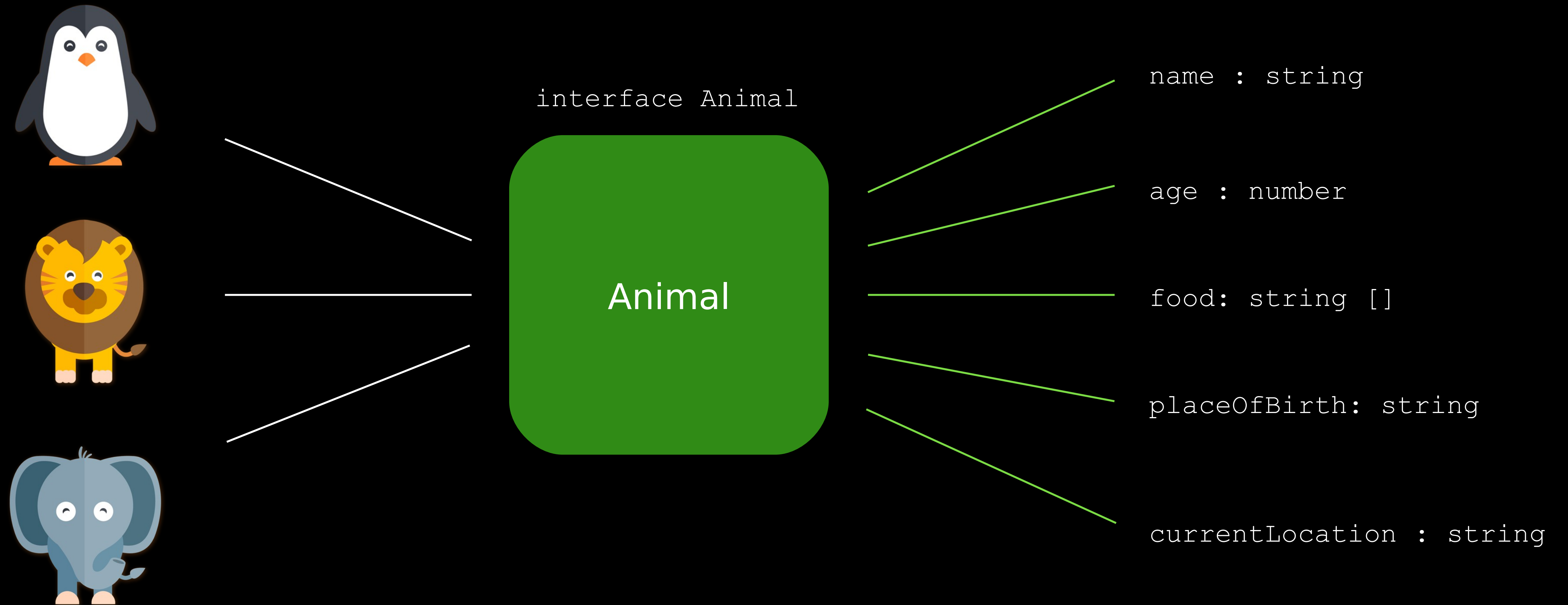
# Objekte



# Objekte



# Objekte



# Objekte – Definition und Initialisierung

```
// Datenstruktur des Objekts festlegen
interface Animal {
    name: string;
    age: number;
    food: string [];
    placeOfBirth: string;
    currentLocation: string;
}

// Objekt initialisieren
var penguin: Animal = {

    /* Schreibweise als sog. "Property-List" */
    name: "Pingu",
    age: 2,
    food: ["Seelachs", "Dorade", "Red Snapper"],
    placeOfBirth: "Südpol",
    currentLocation: "Hamburg"
}
```

**Property List**

```
var propertyList: Typ = {
    variable1 : Wert,
    variable2 : Wert,
    ...
}
```



# Objekte – Zugriff auf Eigenschaften

```
// Objekt initialisieren
var penguin: Animal = {

    /* Schreibweise als sog. "Property-List" */
    name: "Pingu",
    age: 2,
    food: ["Seelachs", "Dorade", "Red Snapper"],
    placeOfBirth: "Südpol",
    currentLocation: "Hamburg"

}
```

```
console.log (penguin.name);
```

```
// Zugriff lesend
```

```
penguin.age++;
```

```
// Zugriff schreibend
```

```
penguin.currentLocation = "Stuttgart";
```

```
// Zugriff schreibend
```

**Punkt-Operator**  
**„Zugriffsoperator“**

`objectVariable.eigenschaft`

# Praktikum

- Aufgabe 10: Shopping Data – Objekte

# Vielen Dank.