

# JavaScript Advanced Course Part 1

[jan.schulz@devugees.org](mailto:jan.schulz@devugees.org)

# Agenda

1. Let, Var and Const
2. Default Arguments
3. Unary Operators
4. Primitives and Objects
5. Object Oriented Paradigm
6. Constructors and Instances
7. Inheritance
8. Prototype
9. Prototype-Chain
10. Class Keyword

# 1. Let, Var and Const

- All of them do the same: create a placeholder for a value
- Difference between var -> let, const
  - Var is valid inside the function scope
  - Let, const is valid inside the block scope
- Difference between const -> let, var
  - Const does not allow to reassign the placeholder

## 2. Default Arguments

```
function multiply(a, b=1) {  
    return a*b;  
}
```

```
console.log(multiply(5, 2));  
console.log(multiply(5));
```

# 3. Unary Operators

Operator	Explanation
+	Tries to convert operand into a number
-	Tries to convert the operand into a number and negates after
!	Converts to boolean value and negates it
++	Adds one to its operand
typeof	Returns a string which is the type of the operand
delete	Deletes specific index of an array or specific property of an object

# 4. Primitives and Objects

## PRIMITIVES

Numbers  
Strings  
Booleans  
Undefined  
Null

## OBJECTS

Arrays  
Functions  
Objects  
Dates  
Wrappers for Numbers,  
Strings, Booleans

... IS AN OBJECT

# 5. Object Oriented Paradigm

OOP (Object Oriented Programming)

- Object interacting with one another through methods and properties
- Used to store data, structure applications into modules and keeping code clean

# 6. Constructors and Instances

```
var john = {  
  Name: 'John',  
  yearOfBirth: 1990,  
  isMarried: false  
}
```

```
var jane = {  
  Name: 'Jane',  
  yearOfBirth: 1991,  
  isMarried: true  
}
```

```
var mark = {  
  Name: 'Mark',  
  yearOfBirth: 1948,  
  isMarried: true  
}
```



# 6. Constructors and Instances

```
var john = {  
  Name: 'John',  
  yearOfBirth: 1990,  
  isMarried: false  
}
```

```
var jane = {  
  Name: 'Jane',  
  yearOfBirth: 1991,  
  isMarried: true  
}
```

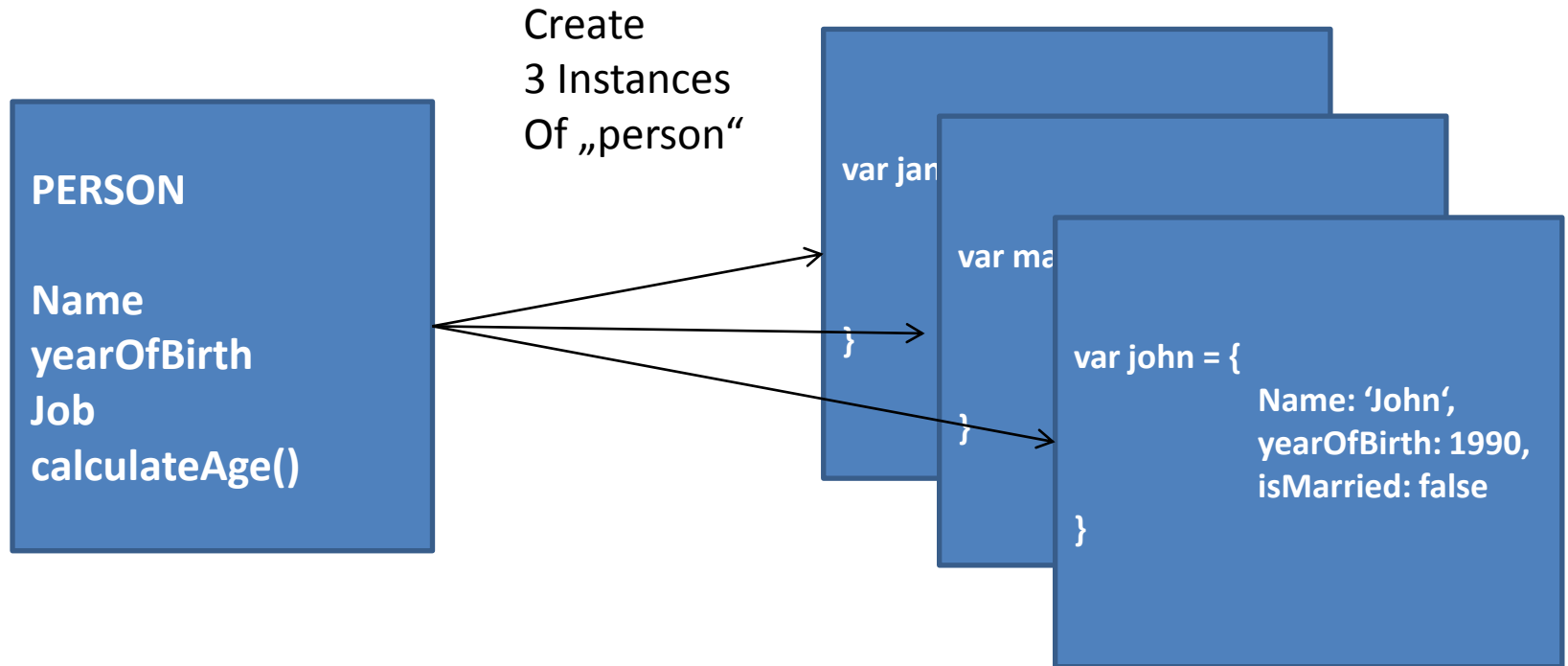
```
var mark = {  
  Name: 'Mark',  
  yearOfBirth: 1948,  
  isMarried: true  
}
```

**3 Objects = A lot of typing**

# 6. Constructors and Instances

CONSTRUCTOR

INSTANCES



# 7. Inheritance

**PERSON**

**Name**

**yearOfBirth**

**Job**

**calculateAge()**

# 7. Inheritance

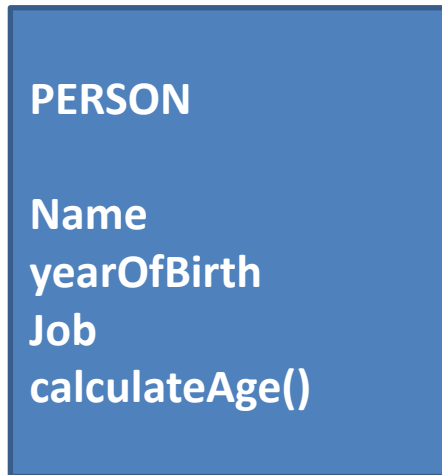
## PERSON

Name  
yearOfBirth  
Job  
calculateAge()

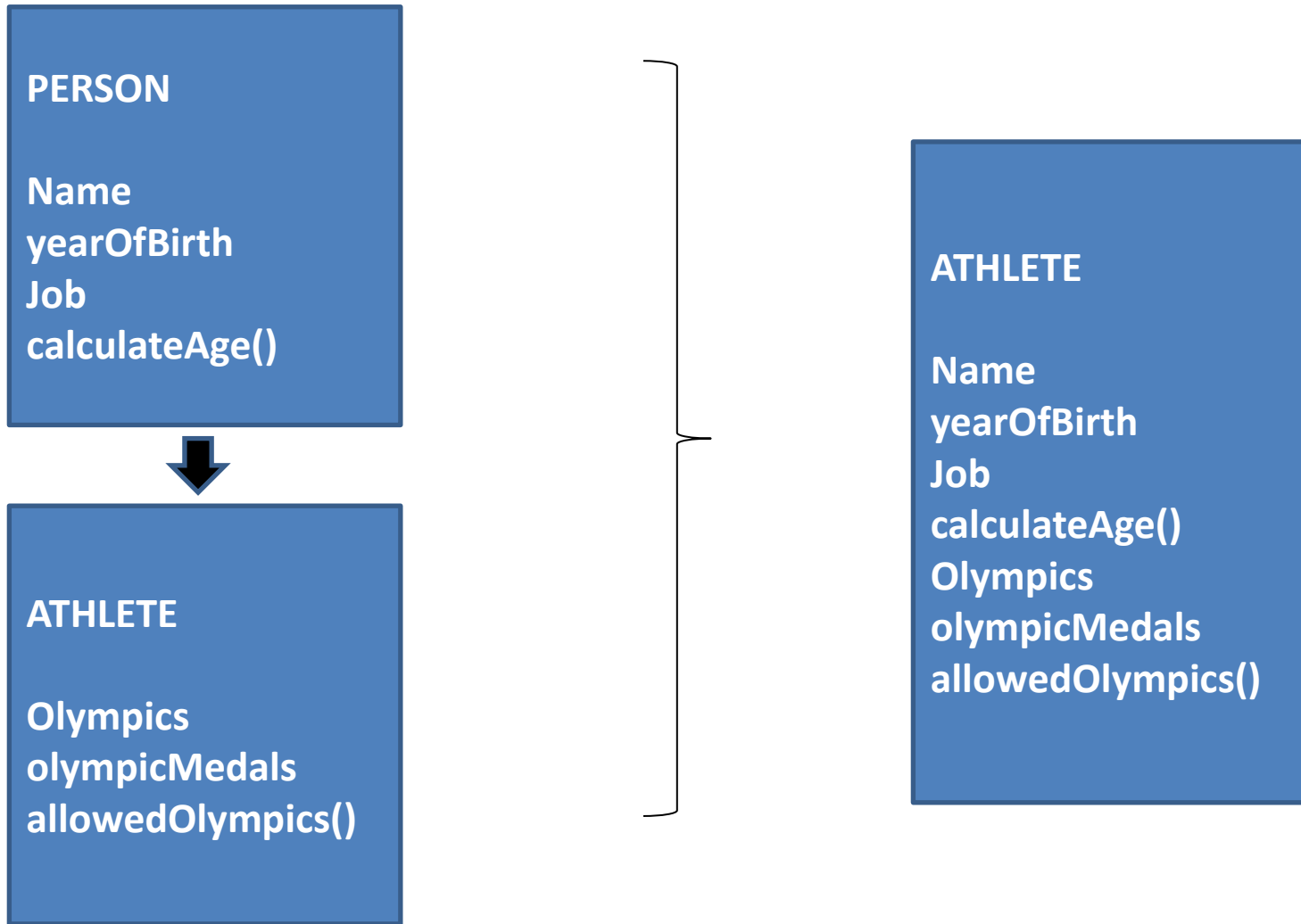
## ATHLETE

Olympics  
olympicMedals  
allowedOlympics()

# 7. Inheritance



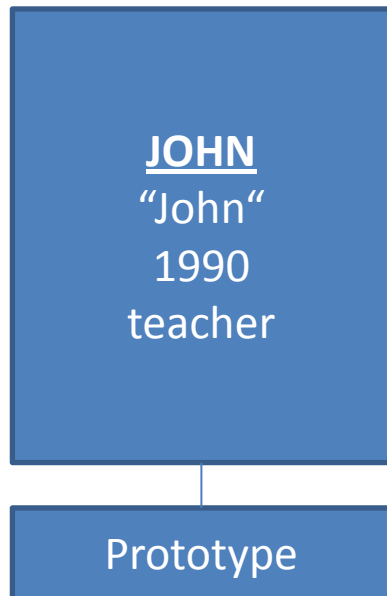
# 7. Inheritance



# 8. Prototype

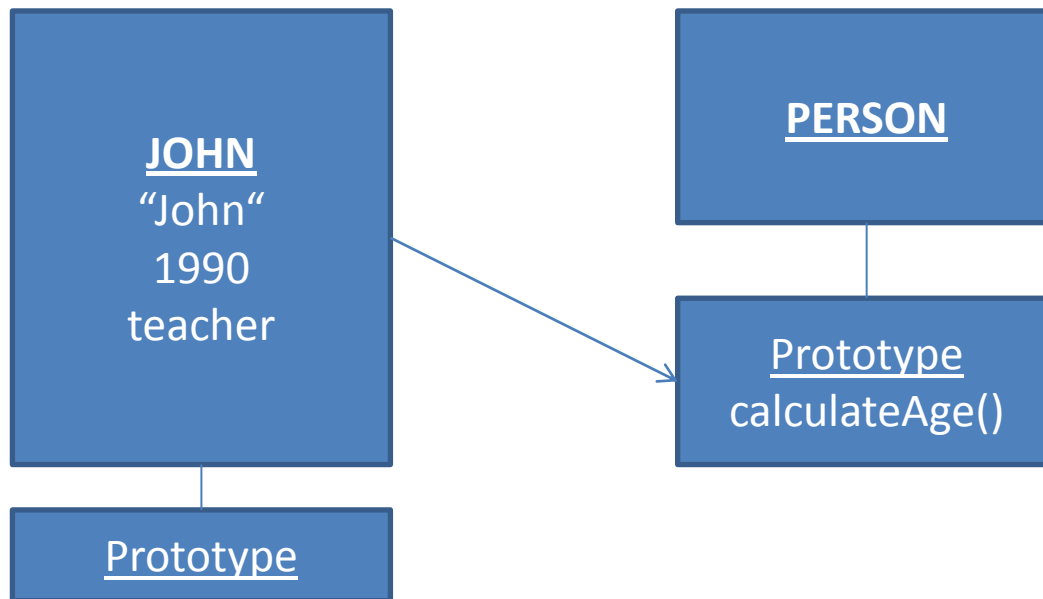
- Every object in JavaScript has an attribute called **prototype**
- Each prototype has an attribute, which itself a prototype
- This goes on, **until prototype is null**

# 9. Prototype-Chain

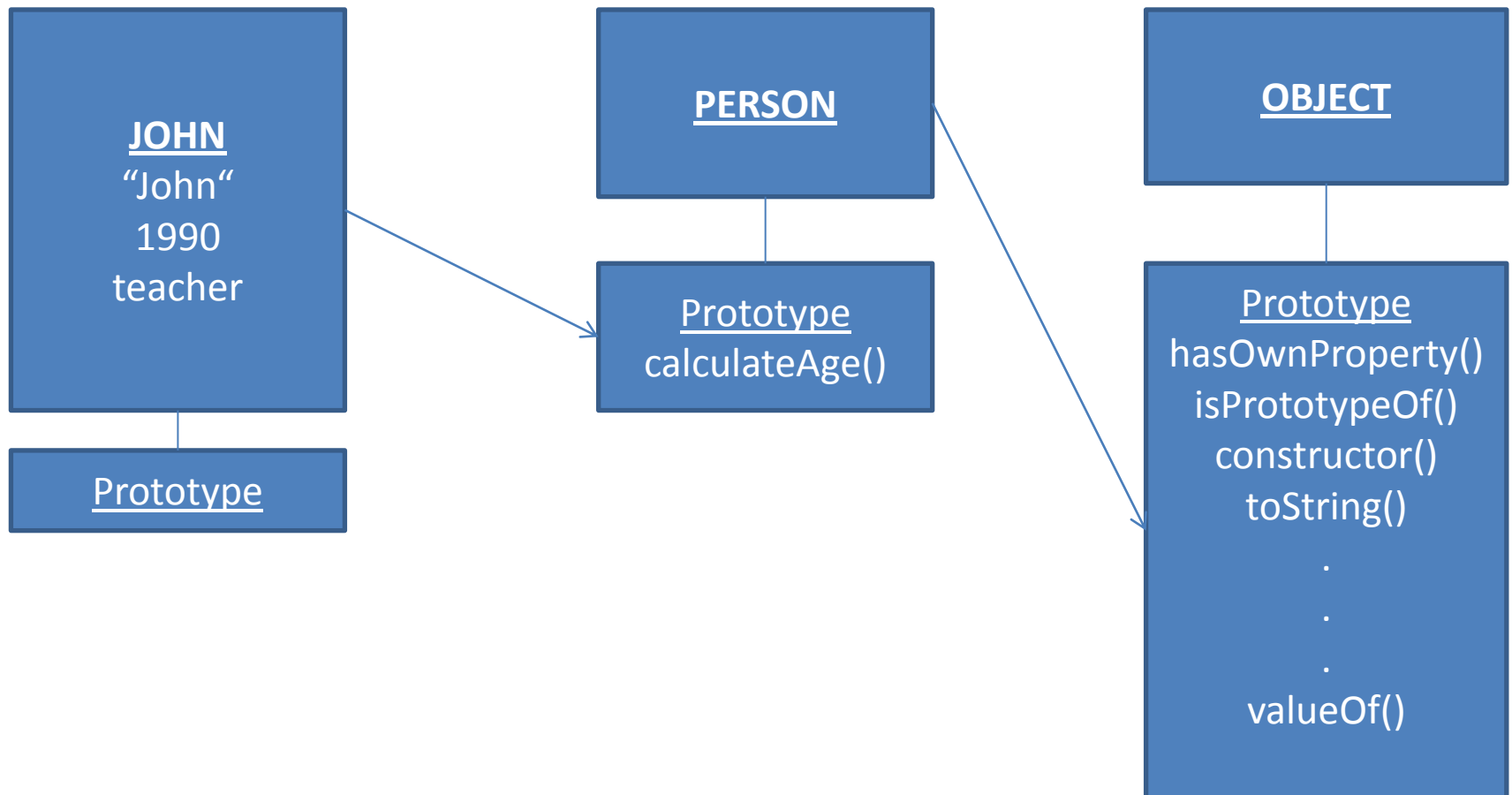




# 9. Prototype-Chain



# 9. Prototype-Chain



# Task

1. Build a function constructor called Question to describe a question. A question should include:  
a) question itself  
b) the answers from which the player can choose the correct one (choose an adequate data structure here, array, object, etc.)  
c) correct answer (I would use a number for this)
2. Create a couple of questions using the constructor
3. Store them all inside an array
4. Select one random question and log it on the console, together with the possible answers (each question should have a number) (Hint: write a method for the Question objects for this task).
5. Use the 'prompt' function to ask the user for the correct answer. The user should input the number of the correct answer such as you displayed it on Task 4.
6. Check if the answer is correct and print to the console whether the answer is correct or not (Hint: write another method for this).
7. Suppose this code would be a plugin for other programmers to use in their code. So make sure that all your code is private and doesn't interfere with the other programmers code (Hint: we learned a special technique to do exactly that).