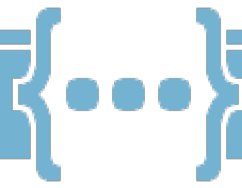


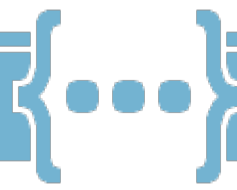
# Cours 12

Introduction aux objets et à leurs attributs

Intro. à la programmation - Aut. 2021

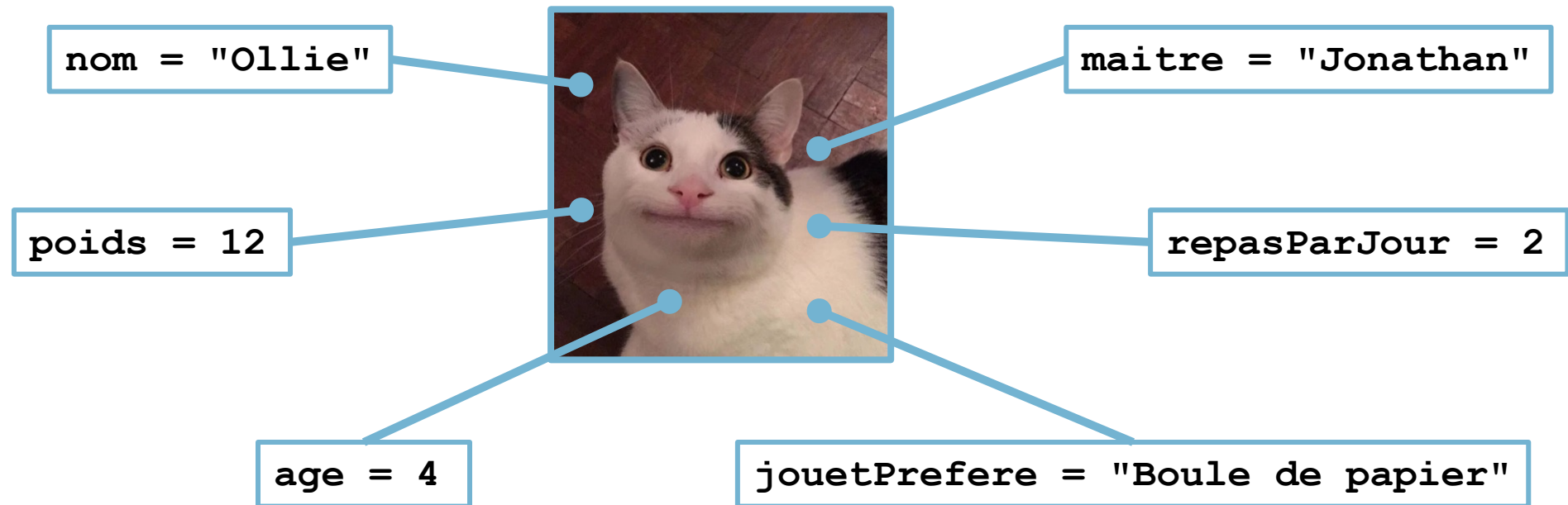


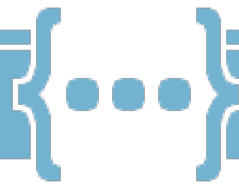
- ❖ Introduction aux objets
- ❖ Propriétés d'objet



## ❖ Un ... objet ?

- ◆ Une entité pour laquelle on souhaite stocker plusieurs données dans un programme.
- ◆ Exemple : Un chat 🐱





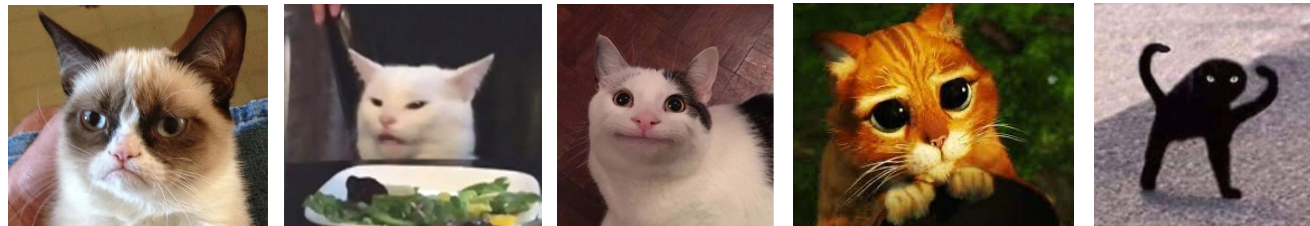
## ❖ Où stocker les propriétés de notre chat ?

### ◆ Dans des variables ?

```
let nomChat = "Ollie";  
let ageChat = 4;  
let poidsChat = 12;  
let jouetPreferChat = "Boule de papier";  
let maitreChat = "Jonathan";  
let repasParJourChat = 2;
```

### ◆ Et si on a plusieurs chats ? Ça devient mélangeant...

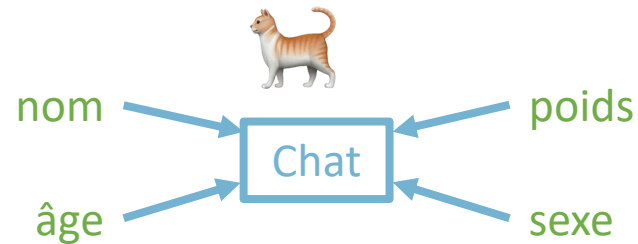
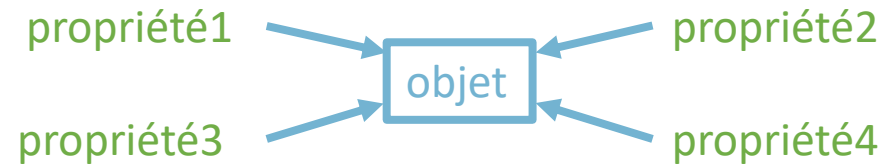
- nomChat1, nomChat2, nomChat3, ageChat1, ageChat2, ageChat3, ... 

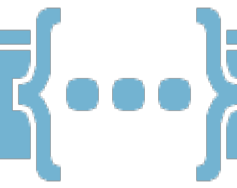




## ❖ Un « objet » en JavaScript

- ◆ Pour le moment, disons que c'est une « donnée qui contient plusieurs **propriétés** ».





## ❖ Comment créer un objet ? 🎩 🔑

- ◆ Similaire à la création d'une variable, sauf que la partie de droite change un peu :

Nom de la variable                      Type d'objet

↓    ↓

```
let monObjet = new Objet();
```

- ◆ Exemple pour un objet de type Chat 🐈

```
let monChat = new Chat();
```

- On a donc désormais une variable de type Chat nommée monChat.



## ❖ Création du chat, puis définir ses propriétés

- ◆ Nous avons une variable nommée « **monChat** » qui est un objet « de **type Chat** » avec les propriétés : **nom**, **age** et **poids**.



monChat

- ◆ Créer le chat :

```
let monChat = new Chat();
```

- Pour le moment, notre chat existe, mais n'a aucune propriété définie.

```
nom ->  
age ->  
poids ->
```

- ◆ Définir ses propriétés

```
monChat.nom = "Ollie";
```

```
monChat.age = 4;
```

```
monChat.poids = 12;
```

- Maintenant, notre chat a des propriétés définies !



monChat

```
nom -> "Ollie"  
age -> 4  
poids -> 12
```



- ❖ Comment savoir quelles propriétés possède un objet ?
  - ◆ Ex : Pour les objets de type **Chat**, qu'est-ce qui détermine qu'ils ont un **nom**, un **âge** et un **poids** ?
  - ◆ Les types d'objets doivent être définis dans le code. (Comme les fonctions)
    - Cela ressemble à ceci :

```
class Chat {  
    constructor(nom, age, poids) {  
        this.nom = nom;  
        this.age = age;  
        this.poids = poids;  
    }  
}
```

- ❖ Sans ce morceau de code, nous ne pourrions pas créer d'objets de type **Chat**.
- ❖ C'est également ici qu'on peut voir les propriétés des chats : **nom**, **age** et **poids**.
  - ◆ Nous sommes libres de définir toutes les propriétés que l'on souhaite !
- ❖ Pas nécessaire de comprendre tout ce code.
  - ◆ Comprendre que cela permet de créer des « **Chat** » et qu'ils ont les propriétés **nom**, **age** et **poids** est suffisant.





## ❖ Deux manières de créer des objets

- ❖ **Méthode 1** : Créer un objet, puis définir ses propriétés.

```
let monChat = new Chat();  
monChat.nom = "Ollie";  
monChat.age = 4;  
monChat.poids = 12;
```

Attention à bien respecter le même ordre qu'ici.

- ❖ **Méthode 2** : Créer un objet en spécifiant ses propriétés en paramètres.

```
let monChat = new Chat( nom: "Ollie", age: 4, poids: 12);
```

```
class Chat {  
  constructor(nom, age, poids) {  
    this.nom = nom;  
    this.age = age;  
    this.poids = poids;  
  }  
}
```



## ❖ Accéder et modifier une propriété

### ◆ Accéder à une propriété :

```
monChat.nom // Vaut "Ollie" (par exemple)
```

```
monChat.age // Vaut 4 (par exemple)
```

```
monChat.poids // Vaut 12 (par exemple)
```

### ◆ Modifier une propriété :

```
monChat.nom = "Bob";
```

```
monChat.age = 9;
```

```
monChat.poids = 13;
```



- ❖ Accéder et modifier une propriété 🚽
- ◆ Créer puis modifier une toilette

```
class Toilette {  
    constructor(numeroCabine, bouchee) {  
        this.numeroCabine = numeroCabine;  
        this.bouchee = bouchee;  
    }  
}
```

🚽 maToilette

maToilette.bouchee vaut true

maToilette.numeroCabine vaut 5



🚽 maToilette

maToilette.bouchee vaut false

maToilette.numeroCabine vaut 4

```
let maToilette = new Toilette( numeroCabine: 5, bouchee: true);  
// On modifie les propriétés de maToilette  
maToilette.numeroCabine = 4;  
maToilette.bouchee = false;
```



## ❖ Afficher les propriétés d'un objet

```
// Création d'un objet de type Chat  
let chat1 = new Chat(nom: "Ollie", age: 4, poids: 12);
```

```
document.getElementById(elementId: "message").textContent = "Mon chat s'appelle " + chat1.nom;  
// -> "Mon chat s'appelle Ollie"
```

```
document.getElementById(elementId: "message").textContent =  
    chat1.nom + " a " + chat1.age + " ans et pèse " + chat1.poids + " livres."  
// -> "Ollie a 4 ans et pèse 12 livres."
```



## ❖ Afficher les propriétés d'un objet 🚽

```
let toilette1 = new Toilette( numeroCabine: 5, bouchée: true);  
let message;  
  
if(toilette1.bouchée == true){  
    message = "La toilette #" + toilette1.numeroCabine + " est bouchée.";  
}  
else{  
    message = "La toilette #" + toilette1.numeroCabine + " est en bon état.";  
}
```

// "La toilette #5 est bouchée."



## ❖ Modifier une propriété 🚽

```
let gToilette1 = new Toilette( numeroCabine: 5, bouchee: true);
let gMessage;

function deboucher(){
    if(toilette1.bouchee == true){
        gMessage = "La toilette #" + toilette1.numeroCabine + " a été débouchée !";
        gToilette1.bouchee = false;
    }
    else{
        gMessage = "Le concierge s'est déplacé pour rien !";
    }
}
```