

# Javascript Objects

Our next data structure



# Objectives

- Understand objects conceptually
- Write code using JS objects



Suppose I wanted to model a single person: name, age, and city

```
//I could use an array like this:  
var person = ["Cindy", 32, "Missoula"];  
  
//to retrieve the person's hometown:  
person[2] //this is not very meaningful code  
  
//what if I accidentally reversed the order?  
var person2 = ["Travis", "Los Angeles", 21];
```

This is a perfect use case for an OBJECT

```
var person = {  
  name: "Cindy",  
  age: 32,  
  city: "Missoula"  
};
```



# Objects

Store data in key-value pairs

```
var person = {  
  name: "Travis",  
  age: 21,  
  city: "LA"  
};
```

Note: unlike arrays, objects  
have no order

'Travis'	21	'LA'
name	age	city



# Retrieving Data

You have two choices: bracket and dot notation

```
var person = {  
  name: "Travis",  
  age: 21,  
  city: "LA"  
};
```

```
//bracket notation, similar to arrays:  
console.log(person["name"]);  
//dot notation:  
console.log(person.name);
```



# Retrieving Data

There are a few differences between the 2 notations:

```
//you cannot use dot notation if the property starts with a number  
someObject.1blah    //INVALID  
someObject["1blah"] //VALID!
```

```
//you can lookup using a variable with bracket notation  
var str = "name";  
someObject.str    //doesn't look for "name"  
someObject[str]  //does evaluate str and looks for "name"
```

```
//you cannot use dot notation for property names with spaces  
someObject.fav color    //INVALID  
someObject["fav color"]; //VALID
```



# Updating Data

Just like an array: access a property and reassign it

```
var person = {  
  name: "Travis",  
  age: 21,  
  city: "LA"  
};  
  
//to update age  
person["age"] += 1;  
//to update city  
person.city = "London";
```

'Travis'	22	'London'
name	age	city



# Creating Objects

Like arrays, there are a few methods of initializing objects

```
//make an empty object and then add to it
var person = {}
person.name = "Travis";
person.age = 21;
person.city = "LA";
```

```
//all at once
var person = {
  name: "Travis",
  age: 21,
  city: "LA"
};
```

```
//another way of initializing an Object
var person = new Object();
person.name = "Travis";
person.age = 21;
person.city = "LA";
```





# Objects

Objects can hold all sorts of data

```
var junkObject = {  
  age: 57,  
  color: "purple",  
  isHungry: true,  
  friends: ["Horatio", "Hamlet"],  
  pet: {  
    name: "Rusty",  
    species: "Dog",  
    age: 2  
  }  
};
```



# Objects

## Exercise 1

```
var someObject = {};
```

```
//Which of the following are valid:
```

```
someObject._name = "Hedwig";
```

```
someObject.age = 6;
```

```
var prop = "color"  
someObject[prop] = "red";
```

```
someObject.123 = true;
```



# Objects

## Exercise 2

```
var someObject = {  
  friends: [  
    {name: "Malfoy"},  
    {name: "Crabbe"},  
    {name: "Goyle"}  
  ],  
  color: "baby blue",  
  isEvil: true  
};
```

```
//Write code to retrieve "Malfoy" from someObject  
//Go one "layer" at a time!
```

