



# **Objects, Constructors and Error Handling**



# LESSON OBJECTIVES

- 
- Understand and implement objects
  - Implement Constructor functions
  - Understand and Implement error handling
- 

# Objects(Object Literal)

---

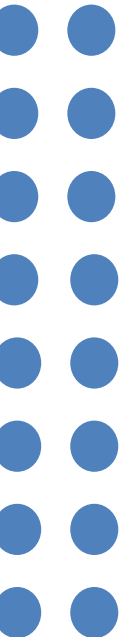
- An object in JavaScript (and many other languages), is like a real life object.
- Properties of an object
- Key - Value pairs
- Accessing properties within objects
- Object methods(see code example after)

## Syntax

```
let objectName = {  
  property1:'value',  
  property2:'value',  
  property3:'value'  
}
```

## Example

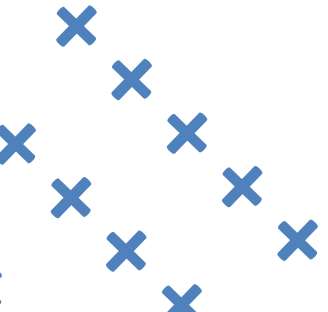
```
let person = {  
  name:'Matthew',  
  surname:'Brown',  
  hairType:'curly'  
}
```



# Factory and Constructor functions

---

- In order to prevent writing duplicate code for objects, we use factory or constructor functions.
- These functions create and return an object to us.
- We will be looking at factory functions first.



# Factory Functions

---

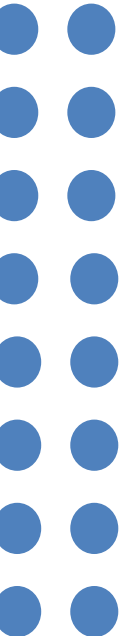
- Returns an object
- Camelcase
- adding methods(see code example after)

Syntax

```
function person(name,last,age,eye) {  
  return{  
    name:name,  
    last:last,  
    age:age,  
    eye:eye  
  }  
}
```

Example

```
let p = person('John','Adams',40,'blue')
```



# Constructor Functions

---

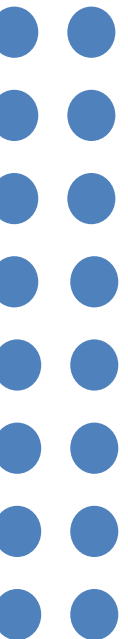
- A constructor is a special function that creates and initializes an object instance of a class.
- Pascalcase
- 'this' keyword
- Key - Value pairs
- adding methods constructors

## Syntax

```
function Person(first, last, age, eye) {  
  this.firstName = first;  
  this.lastName = last;  
  this.age = age;  
  this.eyeColor = eye;  
}
```

## Example to create an object

```
let person1 = new Person('John', 'Adams', '40', 'blue')
```



# Error Handling

- Try and Catch
- Finally
- throw new Error(custom message)

```
try{  
    // type code here  
} catch(error){  
    // handle error  
} finally{  
    // code will alwaysrun regardless  
}  
if(condition){  
    throw new Error('Error message')  
}
```

