

Advanced JavaScript

Eng. Niveen Nasr El-Den
SD & Gaming CoE
iTi

The background features a large, dark blue trapezoidal shape on the left side, which tapers towards the right. To the right of this shape is a white area. At the bottom, there is a horizontal orange bar that also tapers towards the right. The overall design is minimalist and geometric.

Day 1



JavaScript is
Multi-paradigm
Programming Language.





**JavaScript supports
programming
in **many** different **styles**.**



Functional Programming

- Functional programming is a style of programming that attempts to
 - ▷ pass functions as arguments (callbacks) and
 - ▷ return functions without side-effects (changes to the program's state).
- **Immutability** is one of functional programming's best features.
 - ▷ An immutable object is an object that can't be modified after it's created.
 - ▷ JavaScript has a method that **freezes** an object **one-level deep**

Functional Programming

- The ability to pass and return functions is done through one of three concepts:
 - ▷ Pure Functions
 - Are the atomic building blocks in functional programming. They are adored for their simplicity and testability.
 - Pure Functions = Consistent Results
 - Same Input => Same Output
 - ▷ Currying
 - ▷ Higher-Order functions
 - Is a function that either takes a function as an argument or returns a function.

Facts #1 About JavaScript Function

- Every thing you can do with other data types can be done with functions
 - ▷ Function can be
 - assigned to
 - ▷ a variable,
 - ▷ an array element,
 - passed as an argument to another function
 - a value returned from a method call
 - created on the fly
- This makes using *functions* a very handy and flexible, but also a confusing one.

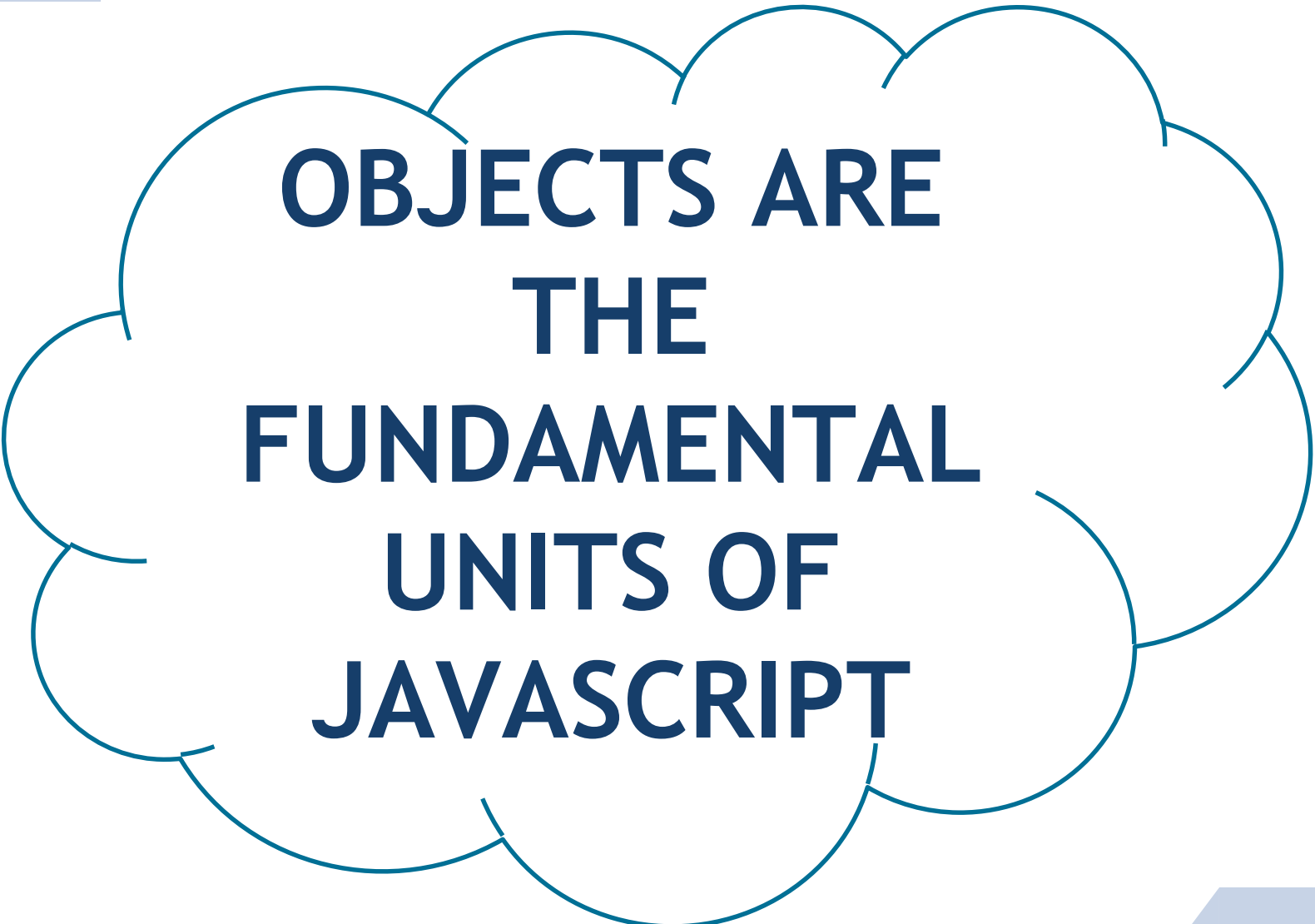

DEMO!!!




**A function always
returns a value**



JavaScript Objects




**OBJECTS ARE
THE
FUNDAMENTAL
UNITS OF
JAVASCRIPT**





An **object** is a collection
of **properties**.

A **property** is an **association**
between a **name**
(or **key**) and a **value**.



JavaScript Built-in Objects

- String

- Number

- Array

- Date

- Math

- Boolean

- RegExp

- Error

- Function

- Object

Creating variables

Literal “Short-Hand”	Constructor
<code>var str = “abc”;</code>	<code>var str = new String();</code>
<code>var arr = [];</code>	<code>var arr = new Array();</code>
<code>var reg = /[a-z]/gmi;</code>	<code>var reg = new RegExp('[a-z]', 'gmi');</code>
<code>var obj = {};</code>	<code>var obj = new Object();</code>
<code>var fn = function(a, b){ return a + b; }</code>	<code>var fn = new Function('a', 'b','return a+b');</code>
	<code>var fn = new Function('a, b','return a+b');</code>

Object Object

- **Object** is the **parent** of all JavaScript objects, which means that every object you create inherits from it
 - ▷ **Reminder** : the **Global** object is **window** object
- To create an object
 - ▷ `var obj = { };` → preferable way
 - ▷ `var obj = new Object();`
- Object object has **constructor** property that used to return the constructor function of the created Object.
- Objects are considered **Associative Arrays** also called a hash (the keys are strings)

JavaScript uses
arrays to represent
indexed arrays
and
objects to represent
associative arrays.

Object Object

//old way of creating an object

```
var obj = new Object();
```

//new way of creating an object (Literal notation)

```
//var obj={ };
```

// adding property to object obj

```
obj.name = "JavaScript"; //dot notation → preferable approach
```

```
//obj["name"] = "JavaScript"; // subscript notation
```

```
var obj = {
```

```
    // adding property to object obj
```

```
    name : "JavaScript",
```

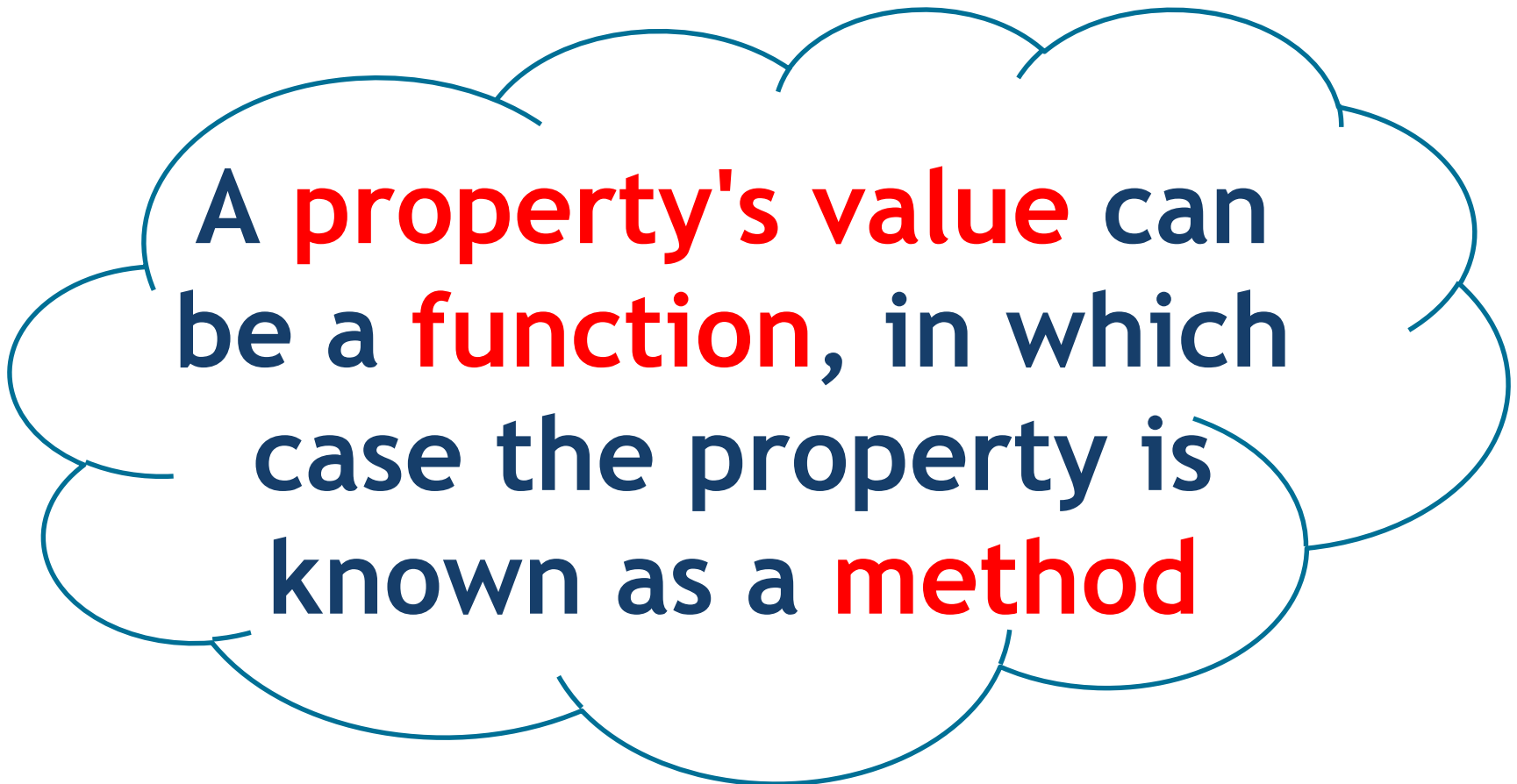

```
    // "name" : "JavaScript",
```

```
};
```


Example!

Object Object Properties & Methods

- `hasOwnProperty("prop")` A method returns true if the current object instance has the property defined in its constructor.
- `.valueOf()`
- `.toString()`
- `Object.keys(obj)`
- `Object.defineProperty(obj, "prop", {})`
- `Object.defineProperties(obj, {})`
- `Object.create(obj [, {}])`
- ...



A **property's value** can
be a **function**, in which
case the property is
known as a **method**



Adding Method to Object

- Method is an action performed by executing a function
- Method is added to an object by assigning a function object as a **value of** an object **property**



Adding Method to Object

- Method is an action performed by executing a function
- Method is added to an object by assigning a function object as a **value of** an object **property**

```
// adding method to object obj  
obj.say = function myFun() //literal function  
{  
  alert( "hello" );  
};  
// “say” is a property that contains a function object
```

Adding Method to Object

- Method is an action performed by executing a function
- Method is added to an object by assigning a function object as a **value of** an object **property**

```
// adding method to object obj
obj.say = function () //literal function
{
  alert( "hello" );
};
// “say” is a property that contains a function object
```

Adding Method to Object

```
var obj = new Object(); //old way of creating an object  
// var obj={ }; //new way of creating an object (Literal)
```

```
// adding property to object obj
```

```
obj.name = "JavaScript"; // dot notation
```

```
//obj["name"] = "JavaScript"; // subscript notation
```

```
// adding method to object obj
```

```
obj.say = function() //literal function
```

```
{
```

```
  alert( "hello" );
```

```
};
```

```
// "say" is a property that contains a function object
```

Adding Method to Object


```
var obj ={  
    // adding property to object obj  
    name : "JavaScript",  
    // "name" : "JavaScript",  
  
    // adding method to object obj  
    say : function() //literal function  
    {  
        alert( "hello" );  
    }  
};
```

Example!


Adding Method to Object

```
var obj ={  
    // adding property to object obj  
    name : "JavaScript",  
    // "name" : "JavaScript",  
  
    // adding method to object obj  
    say : say  
};  
  
function say() {  
    alert( "hello" );  
}
```

Example!



In JavaScript we have
property that contains
Function Object



Assignment