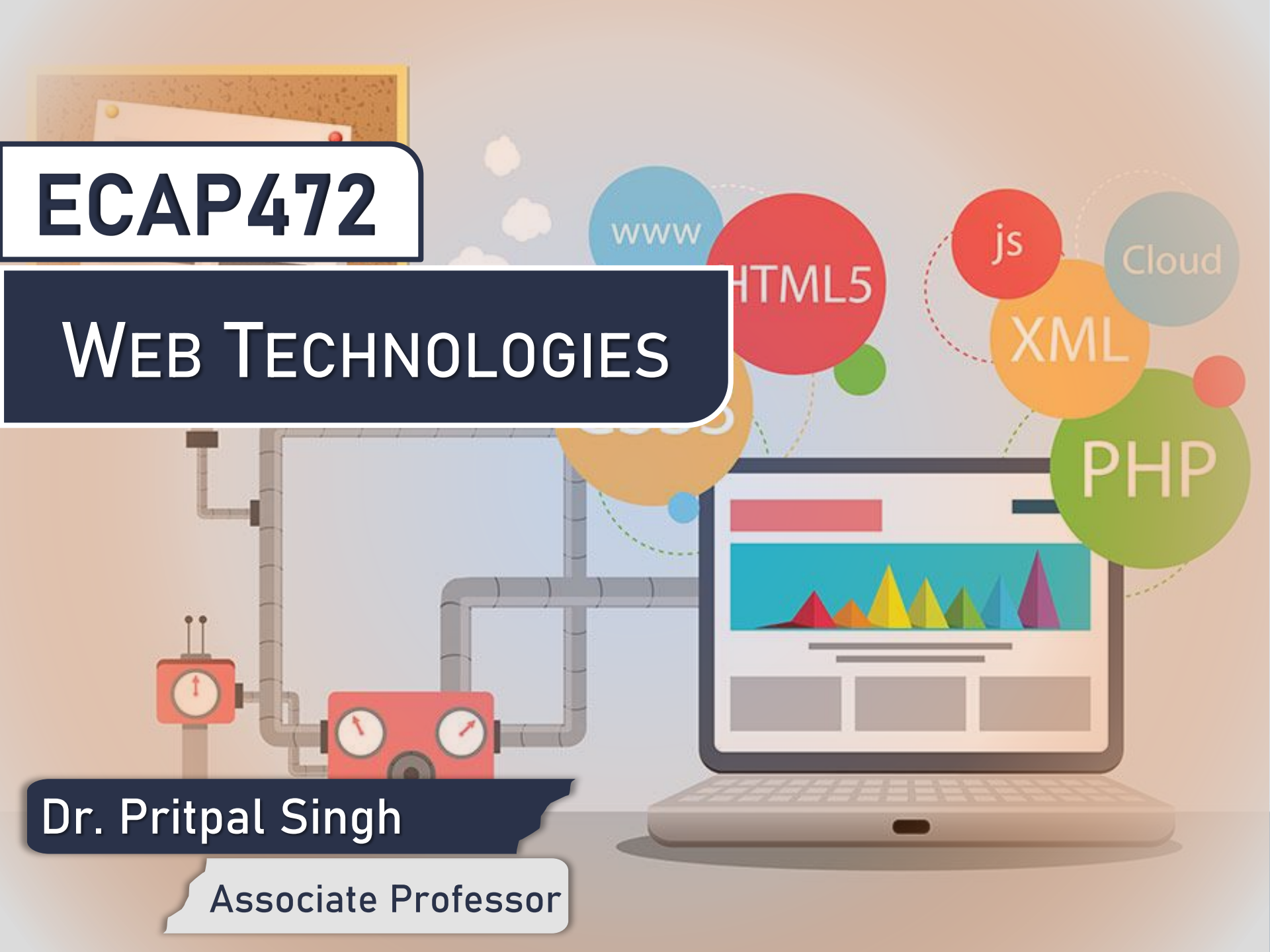
ECAP472

Web Technologies



Dr. Pritpal Singh

Associate Professor

Learning Outcomes

After this lecture, you will be able to

- understand concept of JavaScript Objects.

Objects

- In JavaScript, an object is a standalone entity, with properties and type.
- Compare it with a cup, for example. A cup is an object, with properties. A cup has a color, a design, weight, a material it is made of, etc. The same way, JavaScript objects can have properties, which define their characteristics.

JavaScript Objects

- In JavaScript, objects are king. If you understand objects, you understand JavaScript.
- In JavaScript, almost "everything" is an object.
- Booleans can be objects (if defined with the new keyword)
- Numbers can be objects (if defined with the new keyword)

JavaScript Objects

- Strings can be objects (if defined with the new keyword)
- Dates are always objects
- Maths are always objects
- Regular expressions are always objects

JavaScript Primitives

- A primitive value is a value that has no properties or methods.
- A primitive data type is data that has a primitive value.
- JavaScript defines 5 types of primitive data types:
  - string
  - number
  - boolean
  - null
  - undefined

Primitive Values are Immutable

- Primitive values are immutable (they are hardcoded and therefore cannot be changed).
- if x = 3.14, you can change the value of x. But you cannot change the value of 3.14.

Objects are Variables

- JavaScript variables can contain single values:
- Example
- let person = "John Doe";
- JavaScript variables can also contain many values.
- Objects are variables too. But objects can contain many values.
- Object values are written as name : value pairs (name and value separated by a colon).

Example

- ```
  let person = {firstName:"John",
  lastName:"Doe", age:50, eyeColor:"blue"};
  ```
  ```

Object Properties

- The named values, in JavaScript objects, are called properties.
- Objects written as name value pairs are similar to:
- Associative arrays in PHP
- Dictionaries in Python
- Hash tables in C
- Hash maps in Java
- Hashes in Ruby and Perl

Object Methods

- Methods are actions that can be performed on objects.
- Object properties can be both primitive values, other objects, and functions.
- An object method is an object property containing a function definition.

Creating a JavaScript Object

- With JavaScript, you can define and create your own objects.
- There are different ways to create new objects:
- Create a single object, using an object literal.
- Create a single object, with the keyword new.
- Define an object constructor, and then create objects of the constructed type.
- Create an object using Object.create().

Using an Object Literal

- This is the easiest way to create a JavaScript Object.
- Using an object literal, you both define and create an object in one statement.
- An object literal is a list of name:value pairs (like age:50) inside curly braces {}.
- The following example creates a new JavaScript object with four properties

Example

- const person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
- Spaces and line breaks are not important. An object definition can span multiple lines:

Using the JavaScript Keyword new

- The following example create a new JavaScript object using new Object(), and then adds 4 properties:
- Example
- const person = new Object();
- person.firstName = "John";
- person.lastName = "Doe";
- person.age = 50;
- person.eyeColor = "blue";

JavaScript Objects are Mutable

- Objects are mutable: They are addressed by reference, not by value.
- If person is an object, the following statement will not create a copy of person:
- const x = person; // Will not create a copy of person.

JavaScript Objects are Mutable

- The object x is not a copy of person. It is person. Both x and person are the same object.
- Any changes to x will also change person, because x and person are the same object.

Example

```
const person
= {
  firstName:"John",
  lastName:"Doe",
  age:50, eyeColor:"blue"
}
```

- const x = person;
- x.age = 10; // Will change both x.age and person.age

How to Display JavaScript Objects?

- Displaying a JavaScript object will output [object Object].

  Example

```
const person = {
  name: "John",
  age: 30,
  city: "New York"
};
```

- document.getElementById("demo").innerHTML = person;

Some Common Solutions to Display JavaScript Objects are:

- Displaying the Object Properties by name
- Displaying the Object Properties in a Loop
- Displaying the Object using Object.values()
- Displaying the Object using JSON.stringify()

What are Advantage of Objects in JavaScript

- The advantages of using object literals to create objects include convenience, flexibility in declaration, and less code during declaration. You can drop an object literal anywhere in your program with no previous setup and it'll work, which can be very handy!

That’s all for now…