

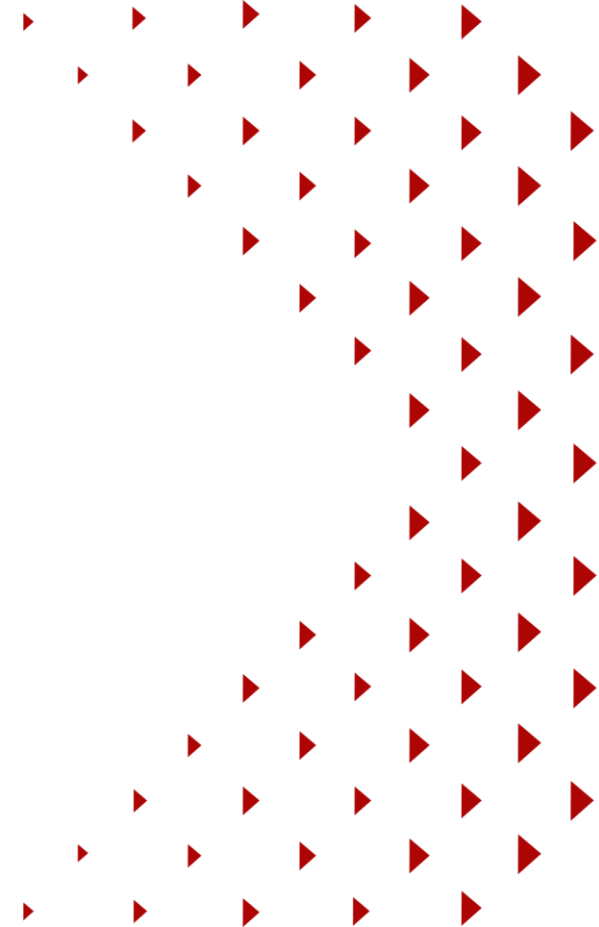


SESSION 15:

OBJECT

Module 1: Web Application UI Design

Version: 1.0



1. OOP – Lập trình hướng đối tượng
2. Object – Đối tượng
3. Truy cập thuộc tính, phương thức

OOP – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG - 1

- **Lập trình hướng đối tượng**
 - Hướng tiếp cận đa dạng gần với thực tế
 - Tìm và phân tích mối quan hệ giữa các đối tượng trong bài toán
 - Mã lệnh và dữ liệu liên kết trong thể thống nhất
 - Javascript được thiết kế dựa trên mô hình đối tượng đơn giản. Mỗi đối tượng chứa nhiều thuộc tính
 - Lập trình hướng đối tượng trong Javascript là xử lý các yêu cầu bằng cách đối tượng hóa các dữ liệu cần xử lý
- **Ưu điểm**
 - Khả năng tái sử dụng cao
 - Ổn định và dễ bảo trì
 - Chi phí giảm dần



OOP – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG - 2

- **Các khái niệm**

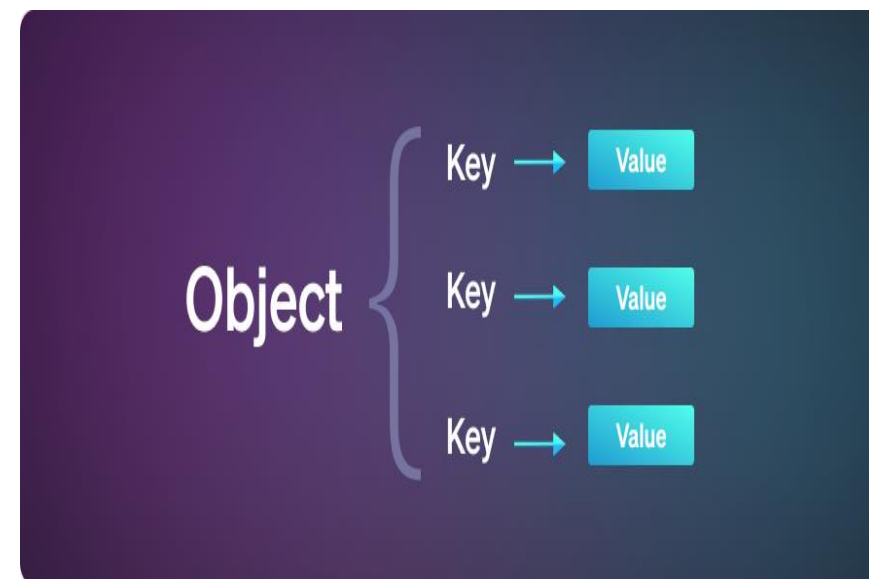
- Object – Đối tượng
- Class – Lớp
- Property – Thuộc tính
- Method – Phương thức, Behavior – Hành vi, Action – Hành động, Capability – Khả năng

- **Đặc tính của lập trình hướng đối tượng**

- Tính bao đóng
- Tính kế thừa
- Tính đa hình
- Tính trừu tượng

OBJECT - 1

- Object trong Javascript là một khái niệm trừu tượng dùng để **biểu diễn một vật thể** (cụ thể).
- Các thuộc tính của đối tượng dùng để miêu tả **đặc điểm, tính chất** của đối tượng
- Tập hợp của các **Key** và **Value**, với Key là thuộc tính và Value là giá trị tương ứng của thuộc tính
- **Key** lưu trữ kiểu dữ liệu primitive hay non_primitive được gọi là thuộc tính
- **Key** lưu trữ một hàm được gọi là phương thức



OBJECT - 2

- **Khởi tạo đối tượng**
 - Sử dụng Object Literals
 - Tạo đối tượng với Object constructor functions
 - Tạo đối tượng từ lớp Object
 - Tạo đối tượng với Javascript Class
- **Sử dụng Object Literals**

```
var student = {  
  // Thuộc tính  
  studentId: "SV001",  
  studentName: "Nguyễn Văn A",  
  academy: "Rikkei Academy",  
  // Phương thức  
  study: function(){  
    return "Learn HTML, CSS, JS for frontend programming at "+this.academy;  
  }  
}
```

OBJECT - 3

- Tạo đối tượng với Object constructor functions

```
function Student(studentId,studentName,academy){
  this.studentId = studentId,
  this.studentName = studentName,
  this.academy = academy,
  this.study = function(){
    return "Learn HTML, CSS, JS for frontend programming at "+this.academy;
  }
}
var student = new Student("SV001","Nguyễn Văn A","Rikkei Academy");
```

```
function Student(studentId, studentName, academy) {
  return {
    studentId: studentId,
    studentName: studentName,
    academy: academy,
    study: function () {
      return "Learn HTML, CSS, JS for frontend programming at " + this.academy;
    }
  }
}
var student = new Student("SV001", "Nguyễn Văn A", "Rikkei Academy");
```

```
function Student(studentId, studentName, academy) {
  return {
    studentId, studentName, academy,
    study: function () {
      return "Learn HTML, CSS, JS for frontend programming at " + this.academy;
    }
  }
}
var student = new Student("SV001", "Nguyễn Văn A", "Rikkei Academy");
```

OBJECT - 4

- Tạo đối tượng với đối tượng Object

```
var student = new Object();  
// Thêm thuộc tính của đối tượng  
student.studentId = "SV001";  
student.studentName = "Nguyễn Văn A";  
student.academy = "Rikkei Academy";  
// Thêm phương thức của đối tượng  
student.study = function () {  
    return "Learn HTML, CSS, JS for frontend programming at " + this.academy;  
}
```


OBJECT - 5

- Tạo đối tượng với Javascript Class

```
class Student {  
    //constructor - hàm tạo  
    constructor(studentId, studentName, academy) {  
        this.studentId = studentId;  
        this.studentName = studentName;  
        this.academy = academy;  
    }  
    study() {  
        return "Learn HTML, CSS, JS for frontend programming at " + this.academy;  
    }  
}  
var student = new Student("SV001", "Nguyễn Văn A", "Rikkei Academy");
```

TRUY CẬP THUỘC TÍNH, PHƯƠNG THỨC

- Truy cập bằng toán tử .
 ObjectName.PropertyName
- Truy cập bằng []
 ObjectName["PropertyName"]

```
function Student(studentId, studentName, academy) {  
    return {  
        studentId, studentName, academy,  
        study: function () {  
            return "Learn HTML, CSS, JS for frontend programming at " + this.academy;  
        }  
    }  
}  
  
var student = new Student("SV001", "Nguyễn Văn A", "Rikkei Academy");  
console.log("Thông tin sinh viên:", student.studentId + "-" + student['studentName']);  
// Thêm thuộc tính vào đối tượng  
student.age = 20;  
student.sex = true;  
// Duyệt các thuộc tính của đối tượng  
for (const key in student) {  
    console.log(student[key]);  
}
```

- ❑ **Nắm vững lập trình hướng đối tượng trong javascript**
- ❑ **Nắm vững các cách khởi tạo đối tượng**
- ❑ **Nắm vững cách truy cập vào thuộc tính và phương thức của đối tượng**
- ❑ **Nắm vững cách duyệt các thuộc tính của đối tượng**



KẾT THÚC

HỌC VIỆN ĐÀO TẠO LẬP TRÌNH CHẤT LƯỢNG NHẬT BẢN