



---

# Bài 1

# Nhập môn lập trình căn bản

Module: BOOTCAMP PREPARATION

- Trình bày được nội dung, yêu cầu, lịch trình và kết quả của module Bootcamp Preparation
- Trình bày được ý nghĩa của lập trình và ngôn ngữ lập trình
- Trình bày được khái niệm thuật toán
- Mô tả được giải thuật sử dụng được pseudo-code và Flowchart
- Cài đặt được WebStorm
- Tạo được ứng dụng JavaScript đầu tiên
- Sử dụng được các lệnh Git cơ bản



---

# Module Bootcamp Preparation

# Module Bootcamp Preparation

---



- **Mục đích:** Giúp học viên làm chủ các kiến thức lập trình cơ bản và tư duy giải quyết vấn đề. Hoàn thành khoá học, học viên có đủ kiến thức và kỹ năng nền tảng về lập trình để bước sang giai đoạn học lập trình chuyên sâu.
- **Thời gian:** 12 bài
- **Đánh giá:**
  - Thi thực hành và lý thuyết cuối module, điểm đạt: 75%
  - Bảng đánh giá kỹ năng theo chuẩn đầu ra
- **Yêu cầu:**
  - Phần mềm WebStorm

# Module Bootcamp Preparation

---



- **Tài liệu học tập:**

- CodeGymX: Bootcamp Preparation
- Source code mẫu trên kênh Github của CodeGym
- Ứng dụng CodeGym Bob gồm các bài luyện tập, bài học, bài kiểm tra.
- Các tài liệu tham chiếu bên ngoài

- **Tài liệu tham khảo:**

- Khoá học Javascript căn bản trên [Codecademy](#)
- Khoá học Javascript căn bản trên [Khanacademy](#)



---

# Thảo luận

Lập trình và ngôn ngữ lập trình



# Lập trình và Ngôn ngữ lập trình

---

- **Lập trình** là quá trình tạo ra tập các chỉ dẫn (instruction) để ra lệnh cho máy tính hoàn thành một công việc (task) nào đó
- Lập trình bao gồm rất nhiều hoạt động: Tìm hiểu yêu cầu, phân tích, thiết kế, viết code, kiểm thử, triển khai, bảo trì, mở rộng...
- **Ngôn ngữ lập trình** là phương tiện để lập trình viên viết ra các chỉ dẫn cho máy tính



# Các loại ngôn ngữ lập trình

---

- Có nhiều loại ngôn ngữ lập trình khác nhau, phục vụ cho các mục đích khác nhau
- Chẳng hạn, chúng ta có các ngôn ngữ lập trình web, ngôn ngữ lập trình desktop, ngôn ngữ lập trình mobile...
- Học lập trình có nghĩa là:
  - Học tư duy giải quyết vấn đề
  - Học một (hoặc một số) ngôn ngữ lập trình
- Học một ngôn ngữ lập trình có nghĩa là:
  - Học tư duy của ngôn ngữ đó
  - Học cú pháp của ngôn ngữ đó



# Học lập trình

---



- Học lập trình có nghĩa là:
  - Học tư duy giải quyết vấn đề
  - Học một (hoặc một số) ngôn ngữ lập trình
- Học một ngôn ngữ lập trình có nghĩa là:
  - Học tư duy của ngôn ngữ đó
  - Học cú pháp của ngôn ngữ đó



---

# Thảo luận

Khái niệm về thuật toán



# Thuật toán (Algorithm)

---

- Thuật toán/giải thuật bao gồm các chỉ thị để giải quyết một vấn đề
- Có thể sử dụng các cách khác nhau để mô tả thuật toán, các cách thông dụng là:
  - Mã giả (pseudo-code)
  - Lưu đồ (flowchart)

# Thuật toán: Ví dụ

---

- Bài toán:
  - Có hai bình A và B đựng hai loại chất lỏng khác nhau, chẳng hạn bình A đựng rượu, bình B đựng nước mắm. Yêu cầu hoán đổi (swap) chất lỏng đựng trong hai bình
- Thuật toán:
  - Yêu cầu phải có thêm một bình thứ ba gọi là bình C
  - Bước 1: Đổ rượu từ bình A sang bình C
  - Bước 2: Đổ nước mắm từ bình B sang bình A
  - Bước 3: Đổ rượu từ bình C sang bình B

# Các cách biểu diễn giải thuật thông dụng

---



- Mã giả (Pseudo-code)
- Lưu đồ (Flowchart)
- Ngôn ngữ lập trình

# Mã giả

---



- Liệt kê tuần tự các bước bằng ngôn ngữ tự nhiên để biểu diễn thuật toán
- Ưu điểm
  - Đơn giản, không cần kiến thức về cách biểu diễn (lưu đồ, ngôn ngữ lập trình)
- Nhược điểm
  - Dài dòng, không cấu trúc
  - Đôi lúc khó hiểu, không diễn đạt được thuật toán

# Mã giả: Ví dụ

---



- Giải thuật giải phương trình bậc nhất dạng  $ax + b = 0$  như sau:

```
BEGIN
```

```
INPUT a, b
```

```
IF a = 0 THEN
```

```
    IF b = 0 THEN
```

```
        PRINT “Phương trình vô số nghiệm”
```

```
    ELSE
```

```
        PRINT “Phương trình vô nghiệm”
```

```
    END IF
```

```
ELSE
```

```
    PRINT “Phương trình có nghiệm  $x = -b/a$ ”
```


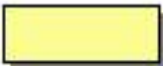

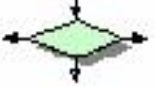


```
END IF
```

```
END
```

# Lưu đồ (Flowchart)

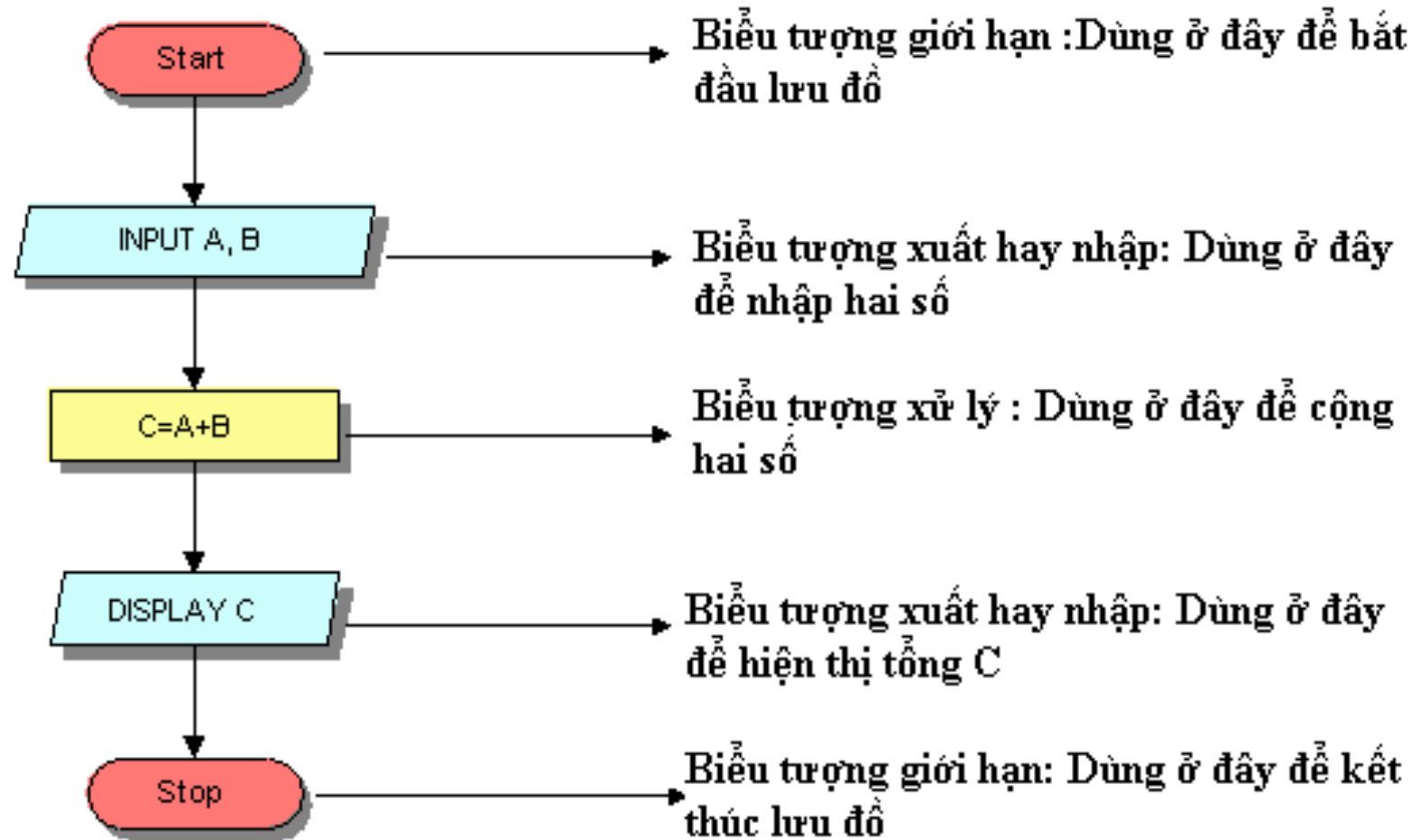


- Lưu đồ mô tả giải thuật bằng các sơ đồ hình khối, mỗi khối quy định một hành động.

Biểu Tượng	Mô Tả
	Bắt đầu hay kết thúc chương trình
	Những bước tính toán
	Các lệnh xuất hay nhập
	Quyết định và rẽ nhánh
	Bộ nối hai phần trong chương trình (đầu nối)
	Dòng chảy



# Lưu đồ: Ví dụ tính tổng hai số





---

# Demo

Cấu trúc lựa chọn

Cấu trúc lặp

# Cấu trúc lựa chọn

---

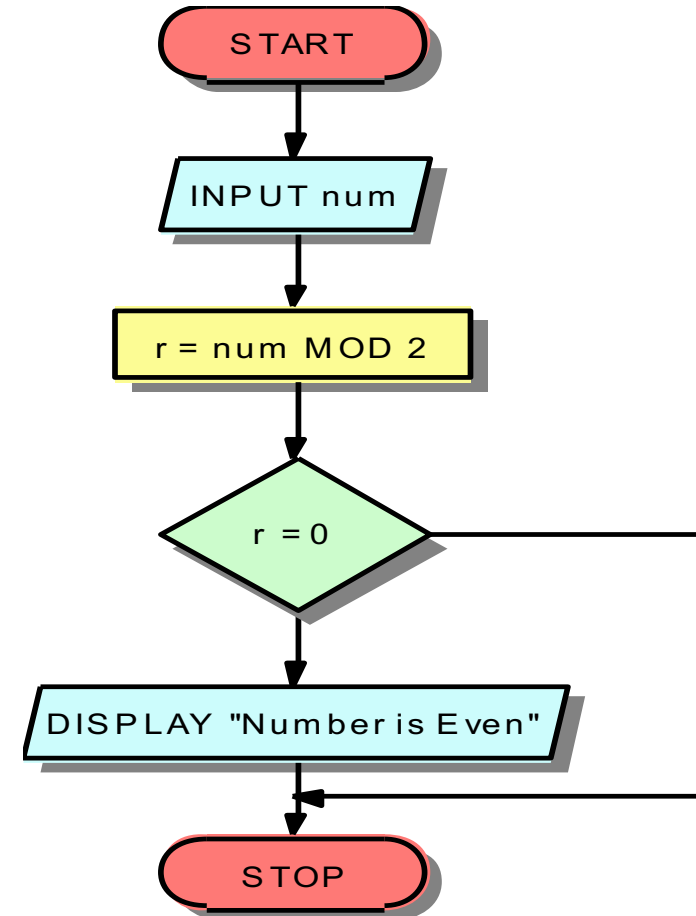


- Lựa chọn một công việc để thực hiện căn cứ vào một điều kiện nào đó.
- Có một số dạng cơ bản như sau:
  - Cấu trúc 1: **Nếu** < điều kiện > (đúng) **thì** thực hiện < công việc >
  - Cấu trúc 2: **Nếu** < điều kiện > (đúng) **thì** thực hiện < công việc 1 >, **ngược lại** (điều kiện sai) **thì** thực hiện < công việc 2 >
  - Cấu trúc 3: Trường hợp < i > thì thực hiện < công việc i >

# Ví dụ cấu trúc lựa chọn: Kiểm tra số chẵn



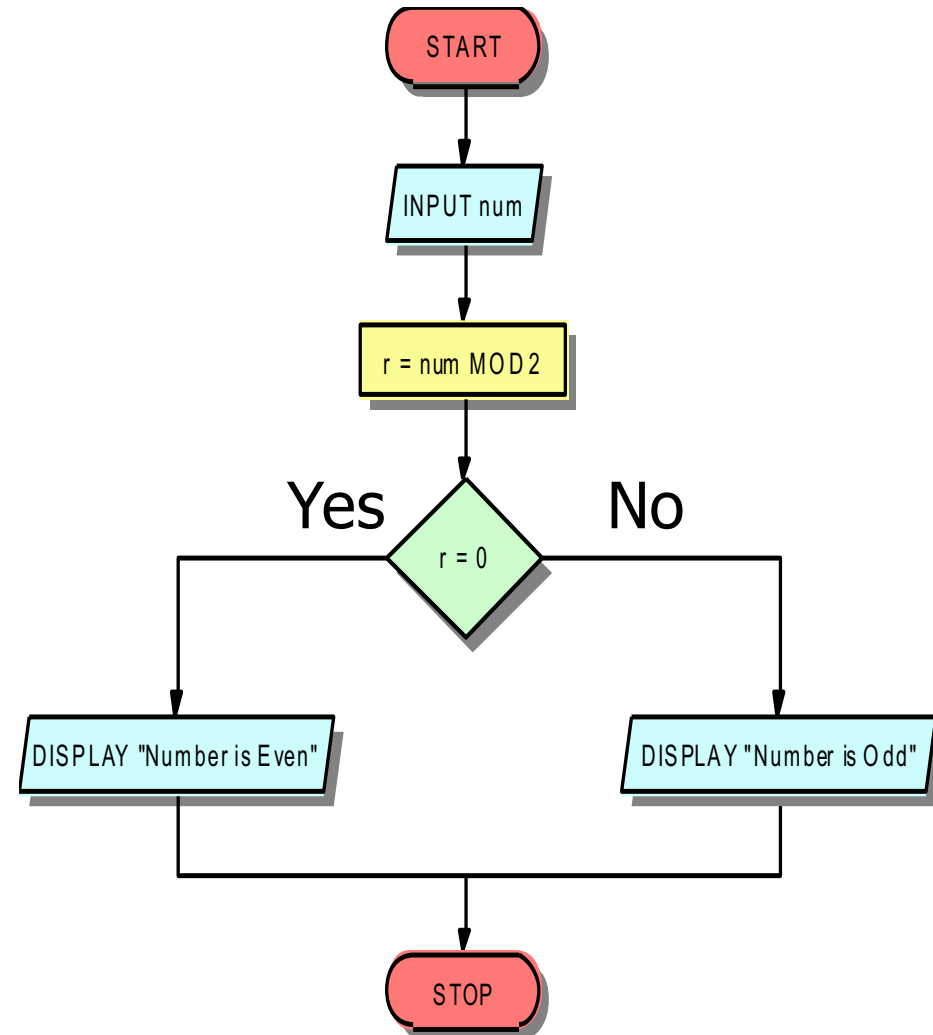
```
BEGIN  
INPUT num  
r = num MOD 2  
IF r=0  
    DISPLAY "Number is even"  
END IF  
END
```



# Cấu trúc lựa chọn: Kiểm tra số chẵn/lẻ



```
BEGIN  
INPUT num  
r=num MOD 2  
IF r=0  
    DISPLAY “Even Number”  
ELSE  
    DISPLAY “Odd Number”  
END IF  
END
```





# Cấu trúc lặp

---

- Thực hiện lặp lại một công việc không hoặc nhiều lần căn cứ vào một điều kiện nào đó.
- Có hai dạng như sau:
  - Lặp xác định: Biết trước số lần lặp
  - Lặp không xác định: Không biết trước số lần lặp

# Cấu trúc lặp: Hiển thị 1000 từ Scooby



BEGIN

cnt=0

**WHILE** (cnt < 1000)

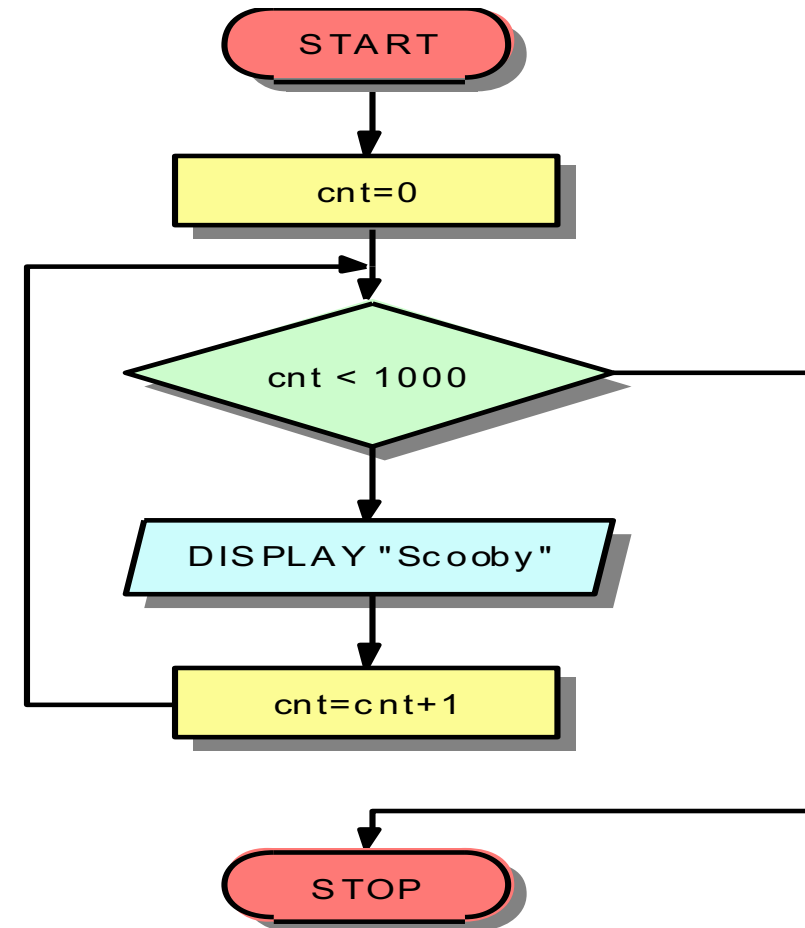
**DO**

    DISPLAY "Scooby"

    cnt=cnt+1

**END DO**

**END**



---

# Nghe video và thảo luận

IDE – Integrated Development Enviroment



# IDE – Integrated Development Environment

---



- Integrated Development Environment: Môi trường phát triển tích hợp là một loại phần mềm máy tính có công dụng giúp đỡ các lập trình viên trong việc phát triển phần mềm.
- IDE gồm:
  - Một trình soạn thảo (*source code editor*)
  - Trình biên dịch (*compiler*) và/hoặc trình thông dịch (*interpreter*).
  - Công cụ xây dựng tự động: khi sử dụng sẽ biên dịch (hoặc thông dịch) mã nguồn, thực hiện liên kết (*linking*), và có thể chạy chương trình một cách tự động.
  - Trình gỡ lỗi (*debugger*): hỗ trợ dò tìm lỗi.

# IDE phổ biến cho JavaScript

---



WebStorm



Visual Studio



Sublime Text



# WebStorm



- Tải webstorm tại: <https://www.jetbrains.com/webstorm/>



# Sử dụng WebStorm IDEA

---



- WebStorm là IDE dùng để soạn thảo mã nguồn
- Tìm bất kỳ lệnh/menu nào:
  - Help/Find Action...
  - $\wedge + \uparrow + A$
  - $\text{⌘} + \uparrow + A$
- Tìm kiếm trong tất cả các file:  $\uparrow + \uparrow$
- Reformat mã:
  - $\text{⌘} + \text{⌘} + L$
  - $\wedge + \text{⌘} + L$



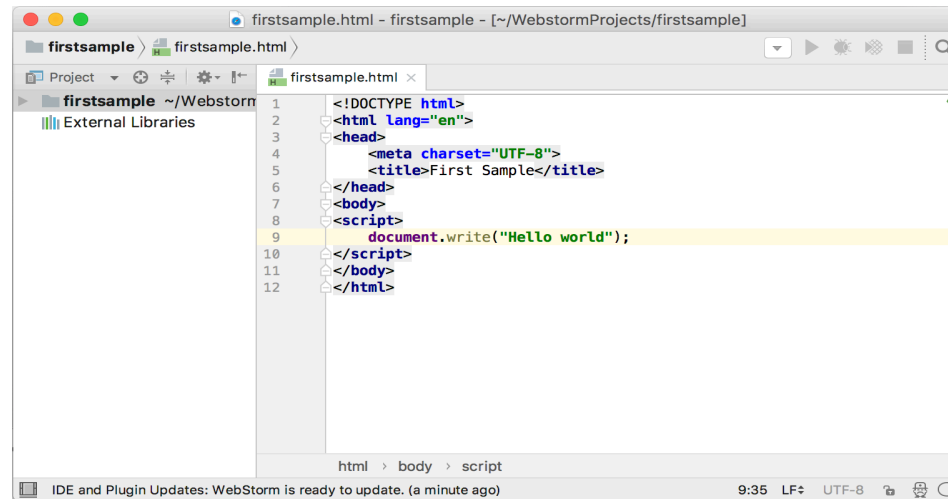
---

# Demo

Ứng dụng JavaScript đầu tiên

# Tạo ứng dụng Javascript đơn giản

- Bước 1: Tạo project mới trên webstorm đặt tên firstsample
- Bước 2: Tạo file firstsample.html với nội dung như hình



- Bước 3: Di chuột vào góc phải phía trên WebStorm thấy xuất hiện biểu tượng



Chọn vào trình duyệt đã cài trên máy để thực thi ứng dụng

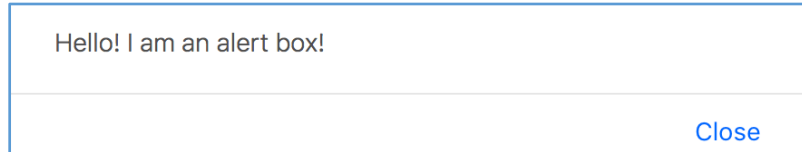
# Các hàm có sẵn: alert(), prompt(), confirm()



```
<!DOCTYPE html>
<html>
<body>

<script>
alert("Hello! I am an alert box!");
</script>

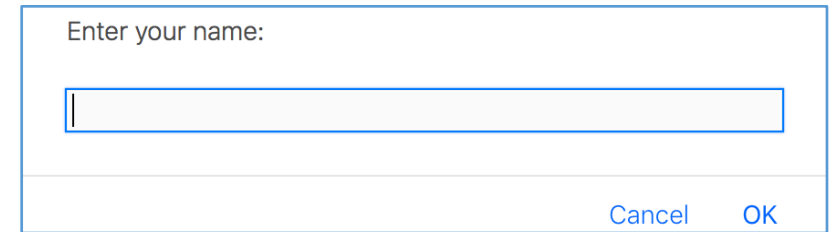
</body>
</html>
```



```
<!DOCTYPE html>
<html>
<body>

<script>
prompt("Enter your name: ");
</script>

</body>
</html>
```



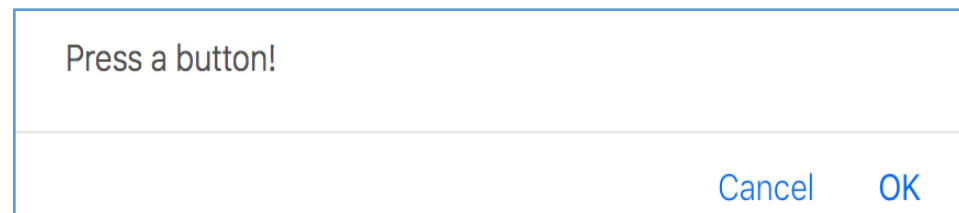
```
<!DOCTYPE html>
<html>
<body>

<script>

confirm("Press a button!");

</script>

</body>
</html>
```



# Javascript

---



- Javascript là một ngôn ngữ lập trình được sử dụng nhiều trên các Website
- Javascript chạy trên trình duyệt (nếu trình duyệt không hỗ trợ thì Javascript không chạy được)
- *Ngày nay, Javascript còn được sử dụng để tạo ra các ứng dụng ở phía back-end*
- Sử dụng Javascript trên trang web, chúng ta có thể:
  - Thay đổi nội dung trang web
  - Thay đổi giao diện trang web
  - Tăng tính tương tác của trang web



# Chương trình máy tính

---



- Chương trình máy tính là các chuỗi **hướng dẫn** (instruction) để máy tính **thực thi** (execute)
- Trong một ngôn ngữ lập trình, các chuỗi hướng dẫn được gọi là **câu lệnh** (statement)



# Câu lệnh (statement)

---

- Câu lệnh được tạo nên từ các thành phần:
  - Giá trị (value)
  - Toán tử (operator)
  - Biểu thức (expression)
  - Từ khoá (keyword)
  - Chú thích (comment)
- Ví dụ, đây là 4 câu lệnh trong JavaScript:

```
var x, y, z;  
x = 5;  
y = 6;  
z = x + y;
```

# Hiển thị kết quả thực thi

---



- Một số cách để hiển thị kết quả thực thi trong Javascript:
  - Sử dụng thuộc tính **innerHTML**
  - Sử dụng **document.write()**
  - Sử dụng hàm **alert()**
  - Sử dụng hàm **console.log()**



---

# Nghe video và thảo luận

SCMs – Source Code Management System

# Quản lý mã nguồn

---



- Lưu trữ mã nguồn tập trung
- Chia sẻ mã nguồn giữa các bên
- Cộng tác giữa các thành viên trong nhóm phát triển
- Khôi phục mã nguồn về các phiên bản khác nhau
- Dễ dàng chỉnh sửa mã nguồn
- Tránh trùng lặp, xung đột mã nguồn

# SCMs – Source Code Management System

---



- Source Code Management System– Hệ thống quản lý mã nguồn là một phần mềm hỗ trợ:
  - Phối hợp giữa các thành viên trong một nhóm phát triển phần mềm
  - Quản lý tập tin và kiểm soát phiên bản
  - Các nhà phát triển khả năng làm việc đồng thời trên các tập tin, hợp nhất với các thay đổi khác của nhà phát triển khác
  - Theo dõi và kiểm tra các thay đổi được yêu cầu và thực thi
  - Theo dõi tình trạng sửa lỗi và thực thi

# VCS – Version Control System

---



- Version Control System (VCS) – Hệ thống quản lý phiên bản mã nguồn là một phần mềm hỗ trợ:
  - Khôi phục lại phiên bản cũ của các file
  - Khôi phục lại phiên bản cũ của toàn bộ dự án
  - Xem lại các thay đổi đã được thực hiện theo thời gian
  - Xem ai là người thực hiện thay đổi cuối cùng có thể gây ra sự cố
  - Khôi phục lại các file vô tình xóa mất

# Các công cụ quản lý mã nguồn thông dụng

---

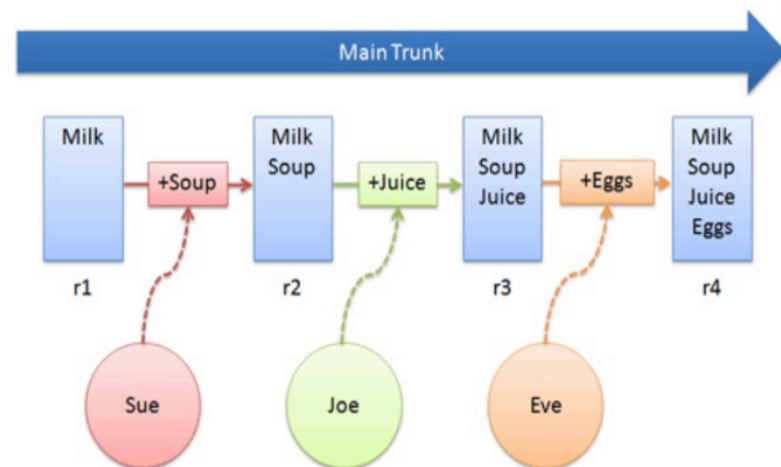


- Subversion
- Git
- Mercurial
- Bazaar
- CVS

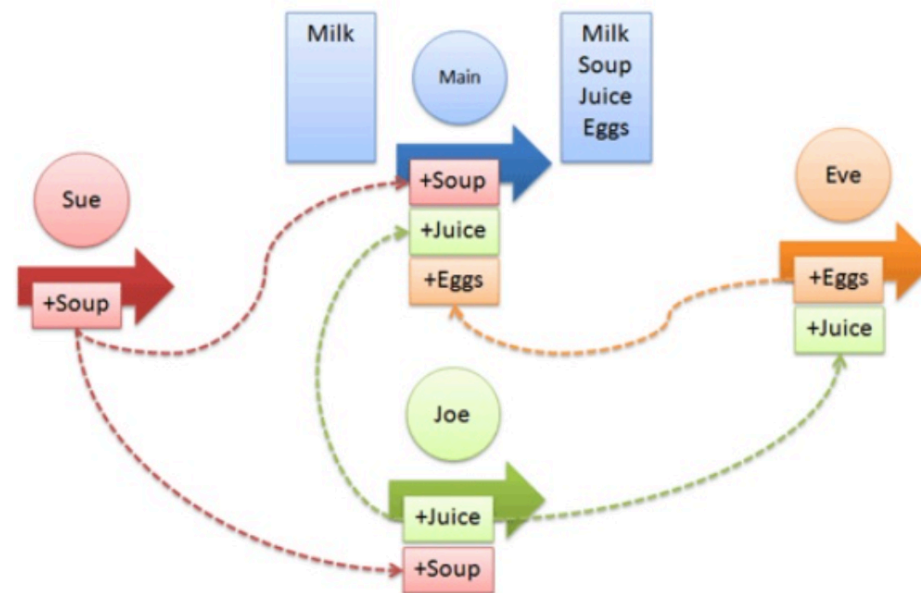




# Các kiểu của hệ thống quản lý phiên bản ( VCS)



Centralized – VCS



Distributed- VCS

# Git và GitHub

---



- Git là một hệ thống điều khiển phiên bản (version control system) theo hình thức phân tán
- Git được sử dụng để quản lý mã nguồn (source code) và ghi nhận các thay đổi
- GitHub là một dịch vụ Git được cung cấp miễn phí
- GitHub có phiên bản trả phí dành cho các doanh nghiệp





---

# Sử dụng Git

# Snapshot

---



- Snapshot là toàn bộ mã nguồn tại một thời điểm
- Các snapshot như là các lát cắt được tạo ra trong quá trình phát triển
- Lập trình viên quyết định lúc nào thì tạo một snapshot
- Có thể quay lại một snapshot bất kỳ

# Commit

---



- Commit là cách để tạo các snapshot
- Các commit thường được tạo ra khi có một thay đổi đáng kể đối với mã nguồn:
  - Tạo một tính năng mới
  - Sửa được một lỗi
  - Cải tiến mã nguồn
  - ...
- Một commit bao gồm các thông tin:
  - Thay đổi ở các file so với trước
  - Một tham chiếu đến commit trước nó (gọi là commit cha)
  - Một mã băm đại diện, thường có dạng như  
**87878747939740429190ca307289c494311e27fe**

# Repository

---

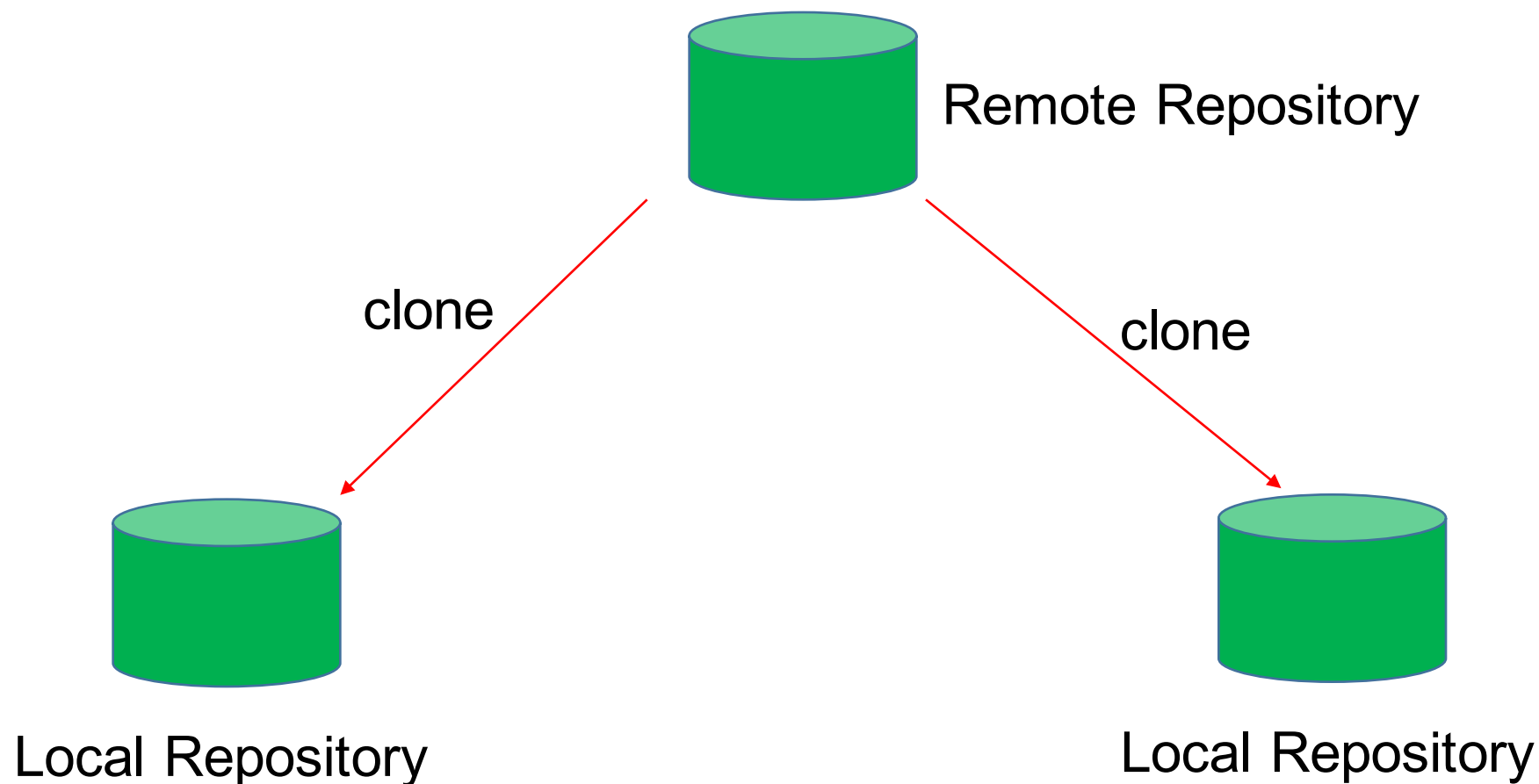


- Thường được gọi ngắn gọn là repo
- Repository là nơi chứa toàn bộ mã nguồn
- Repository bao gồm toàn bộ các file và lịch sử của các file đó
- Repository chứa tất cả các commit
- Có 2 loại repository:
  - Local Repository: Ở trên máy của lập trình viên
  - Remote Repository: Ở trên một máy chủ chia sẻ (chẳng hạn như GitHub)

# Clone



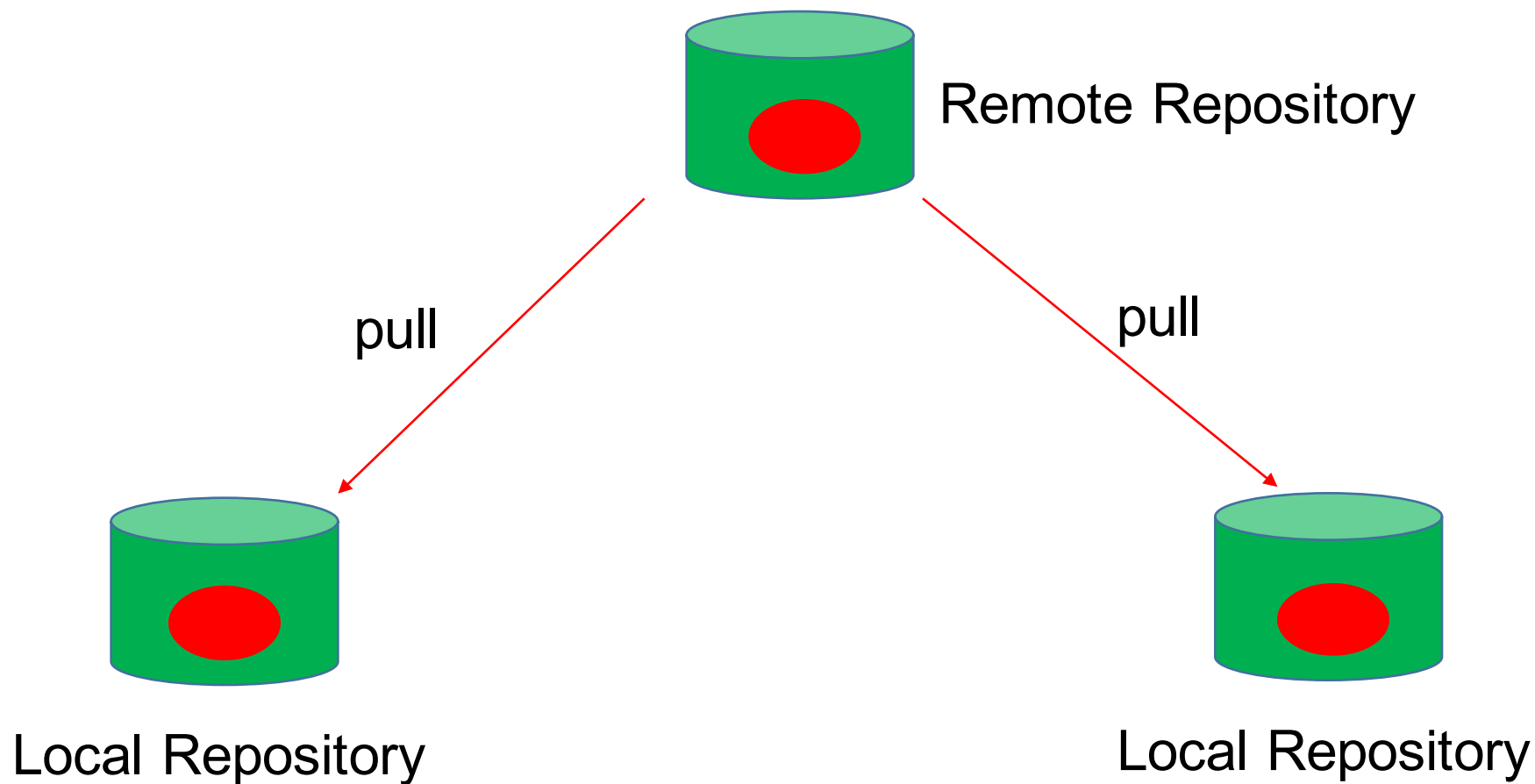
- Sao chép một Remote Repository về máy của lập trình viên



# Pull



- Cập nhật mã nguồn từ một Remote Repository về Local Repository

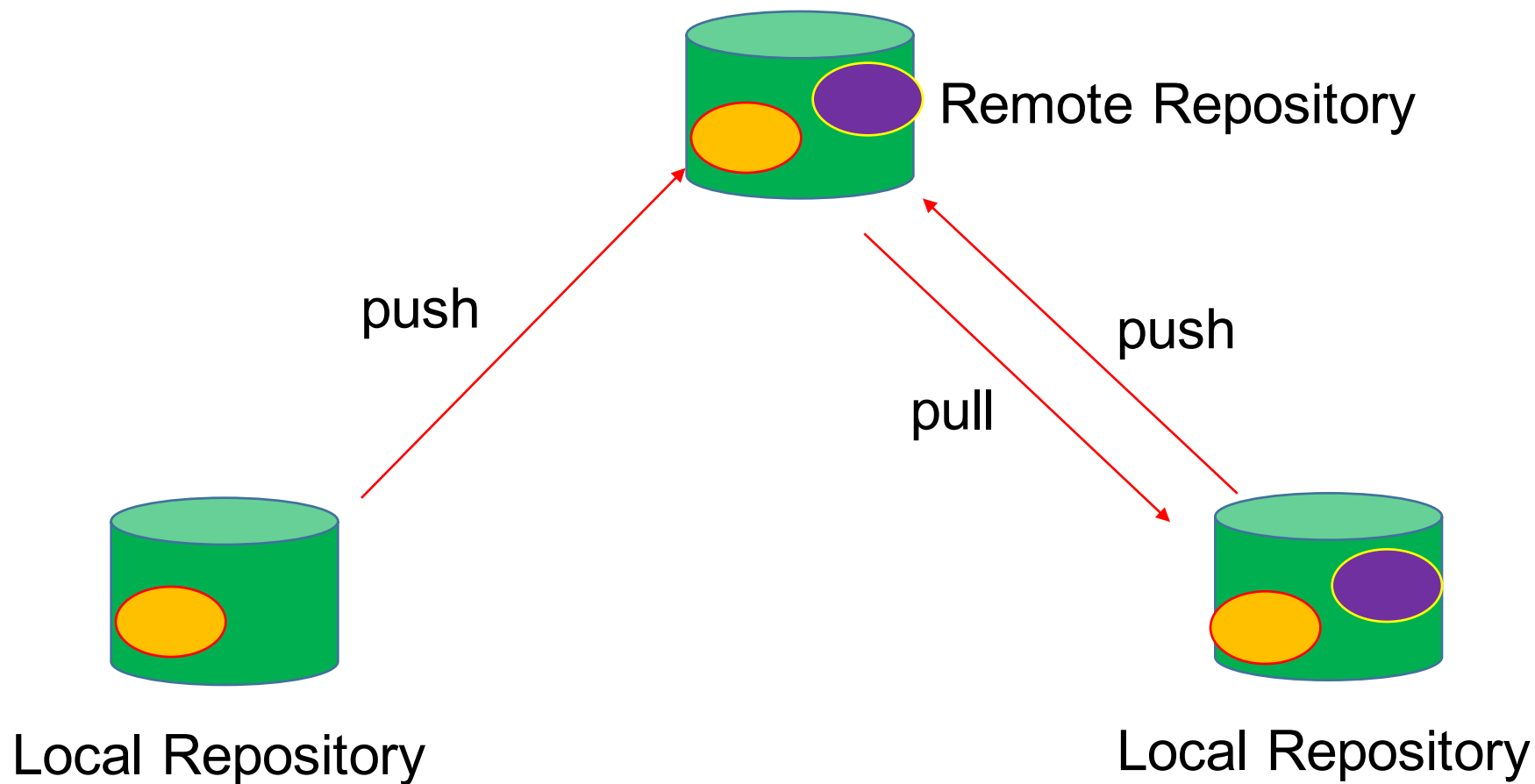




# Push



- Đẩy mã nguồn từ Local Repository lên Remote Repository





---

# Demo

- Sử dụng lệnh Git cơ bản



# Các câu lệnh cơ bản của Git

- `git clone`: Sao chép một repository
- `git init`: Khởi tạo một repository
- `git add`: Đưa các file vào trong vùng staged
- `git commit`: Ghi nhận các thay đổi
- `git push`: Đưa các thay đổi từ local repository lên remote repository



# Tóm tắt bài học

---



- Module Bootcamp Preparation giúp học viên sẵn sàng tham gia giai đoạn Bootcam tại CodeGym
- Có nhiều ngôn ngữ lập trình khác nhau để giúp giải quyết các vấn đề khác nhau
- Thuật toán là tập hợp các bước để giải quyết một vấn đề thực tế
- Có thể sử dụng các cách khác nhau để mô tả thuật toán, chẳng hạn như Mã giả, Lưu đồ
- Có nhiều IDE hỗ trợ việc viết mã, WebStorm là một trong số đó
- JavaScript có nguồn gốc là một ngôn ngữ chạy trên trình duyệt, ngày nay thì JavaScript còn được sử dụng để lập trình back-end và mobile
- Git là cơ chế quản lý mã nguồn phổ biến nhất hiện nay, GitHub là một dịch vụ miễn phí

---

# Hướng dẫn

- Hướng dẫn làm bài thực hành và bài tập
- Chuẩn bị bài tiếp: *Biến, toán tử và kiểu dữ liệu*