FPT Education
FPT POLYTECHNIC

THỰC HỌC – THỰC NGHIỆP

LẬP TRÌNH JAVASCRIPT

BASIC JAVASCRIPT

FPT Education
FPT POLYTECHNIC

Hệ thống bài cũ

- Biến và khai báo biến
- Các kiểu dữ liệu
- Ép kiểu
- Operators
- Công cụ gỡ lỗi

FPT Education
FPT POLYTECHNIC

MỤC TIÊU BÀI HỌC

- Cấu trúc điều khiển
- Hàm

FPT Education
FPT POLYTECHNIC

PHẦN I

Cấu trúc điều khiển

FPT Education
FPT POLYTECHNIC

Các lệnh logic

- if và if else
- else if
- Toán tử bậc 3 có điều kiện
- switch

FPT Education
FPT POLYTECHNIC

Logic Statements (Lệnh logic)

- Cú pháp

```
if(expression){
    //code here
}
```

- Ví dụ

```
var x = 3; var y = 4;
if (x == y) {
    //Thực hiện
}
```

FPT Education
FPT POLYTECHNIC

Logic Statements (Lệnh logic)

- Cú pháp

```
if(expression){
    //code here
} else {
    //code here
}
```

- Ví dụ

```
let rain = true;
if(rain){
    console.log("** Taking my umbrella when I need to go outside **");
} else {
    console.log("** I can leave my umbrella at home **");
}
```

FPT Education
FPT POLYTECHNIC

Logic Statements (Lệnh logic)

- Cú pháp

```
if (expression){
//code here
} else if (expression) {
//code here
} else if (expression){
//code here
} else {
//code here
}
```

- Ví dụ

```
if(age < 3){
    console.log("Access is free
under three.");
} else if(age < 12) {
    console.log("the fee is 5
dollars");
} else if(age < 65) {
    console.log("A regular ticket
costs 10 dollars.");
} else if(age >= 65) {
    console.log("A ticket is 7
dollars.");
}
```

FPT Education
FPT POLYTECHNIC

Conditional ternaty operators (Toán tử bậc ba có điều kiện)

- Cú pháp

```
operand1 ? operand2 : operand3;

expression ? statement for true : statement associated
with false;
```

- Ví dụ

```
let access = age < 18 ? "denied" : "allowed";
```

FPT Education
FPT POLYTECHNIC

Switch statement (lệnh switch)

- Cú pháp

```
switch(expression) {
   case value1:
      // code to be executed
      break;
   case value2:
      // code to be executed
      break;
   case value-n:
      // code to be executed
      break;
}
```

- Ví dụ

```
switch(activity) {
   case "Get up":
      console.log("It is 6:30AM");
      break;
   case "Breakfast":
      console.log("It is 7:00AM");
      break;
   case "Drive to work":
      console.log("It is 8:00AM");
      break;
}
```

FPT Education
FPT POLYTECHNIC

Switch statement (lệnh switch)

- Cú pháp

```
switch(expression) {
   case value1:
      // code to be executed
      break;
   case value2:
      // code to be executed
      break;
   case value-n:
      // code to be executed
      break;
   default:
      // when no cases match
      break;
}
```

- Ví dụ

```
switch(activity) {
   case "Get up":
      console.log("It is 6:30AM");
      break;
case "Breakfast":
      console.log("It is 7:00AM");
      break;
case "Drive to work":
      console.log("It is 8:00AM");
      break;
default:
      console.log("I cannot determine the
current time.");
      break;
}
```

FPT Education
FPT POLYTECHNIC

Loops (vòng lặp)

- Break, continue
- Lệnh lặp không biết trước số lần lặp
  - while
  - do while
- Lệnh lặp biết trước số lần lặp
  - For
  - For of
  - For in (học ở phần object)

While loops

- Cú pháp

```
while (condition) {
    // code that gets executed as
    //long as the condition is true
}
```

- Ví dụ

```
let i = 0;
while (i < 10) {
    console.log(i);
    i++;
}
```

Do while loops

- Cú pháp

```
do {
    // code to be executed if the condition is true
} while (condition);
```

- Ví dụ

```
do {
    number = prompt("Please enter a number between 0 and
100: ");
} while (!(number >= 0 && number < 100));
```

FPT Education
FPT POLYTECHNIC

For loops

- Cú pháp

```
for (initialize variable; condition; statement)
{
    // code to be executed
}
```

- Ví dụ

```
for (let i = 0; i < 10; i++)
{
    console.log(i);
}
```

Sơ đồ hoạt động của vòng lặp for

1. Cài đặt biến i=0
2. Kiểm tra điều kiện i<10
3. Nếu điều kiện là đúng, thực thi khối lệnh. Nếu điều kiện sai, vòng lặp kết thúc tại đây
4. Thực hiện câu lệnh i++
5. Trở lại bước 2

FPT Education
FPT POLYTECHNIC

Vòng lặp và mảng

- Cú pháp

```
let arr = [some array];
for (initialize variable; variable smaller than arr.length; statement) {
     // code to be executed
}
```

- Ví dụ

```
let names = ["Chantal", "John", "Maxime", "Bobbi", "Jair"];
for (let i = 0; i < names.length; i ++){
   console.log(names[i]);
}
```

FPT Education
FPT POLYTECHNIC

Vòng lặp for of

- Cú pháp

```
let arr = [some array];
for (let variableName of arr) {
   // code to be executed
   // value of variableName gets updated every iteration
   // all values of the array will be variableName once
}
```

- Ví dụ

```
let names = ["Chantal", "John", "Maxime", "Bobbi", "Jair"];
for (let name of names){
    console.log(name);
}
```

FPT Education
FPT POLYTECHNIC

Lệnh break, continue

- Dùng để kiểm soát sơ đồ thực thi vòng lặp.
- `Break`: dừng vòng lặp và di chuyển sang mã bên dưới vòng lặp
- `Continue`: dừng vòng lặp và quay lại đầu vòng lặp, kiểm tra điều kiện

(continue trong for: thực hiện câu lệnh và sau đó kiểm tra điều kiện)

```
for (let i = 0; i < 10; i++) {
    console.log(i);
    if (i === 4) {
        break;
    }
}
```

```
let i = 1;
while (i < 50) {
    if (i % 2 === 0){
        continue;
    }
    console.log(i);
    i++;
}
```

PHẦN II

Hàm

FPT Education
FPT POLYTECHNIC

Hàm và xử lý sự kiện

- Hàm (function)
- Phạm vi biến (variable scope)
- Xử lý sự kiện (event)

FPT Education
FPT POLYTECHNIC

Hàm (function)

- Hàm: để thực hiện một chức năng cụ thể
- Cú pháp:
  - Hàm cơ bản
  - Hàm có tham số
  - Hàm trả về giá trị

FPT Education
FPT POLYTECHNIC

Hàm (function)

- Hàm cơ bản
  - Hàm xây dựng sẵn: `prompt(), console.log(), push(), sort(),….`
  - Hàm tự xây dựng

```
function nameOfTheFunction() {
    //content of the function
}
```

  - Gọi hàm `nameOfTheFunction();`
  - Ví dụ

```
function sayHello() {
    let you = prompt("What's your name? ");
    console.log("Hello", you + "!");
}
sayHello();
```

Hàm (function)

- Hàm có tham số
  - Cú pháp

```
function myFunc(param1, param2) {
    // code of the function;
}
```

  - Ví dụ

```
function addTwoNumbers(x, y) {
    console.log(x + y);
}
addTwoNumber(5,5);
```

Hàm (function)

- Hàm có tham số

```
function tester(para1, para2){
    console.log(para1 + " " + para2);
}
const arg1 = "argument 1";
const arg2 = "argument 2";
tester(arg1, arg2);
```

Hàm (function)

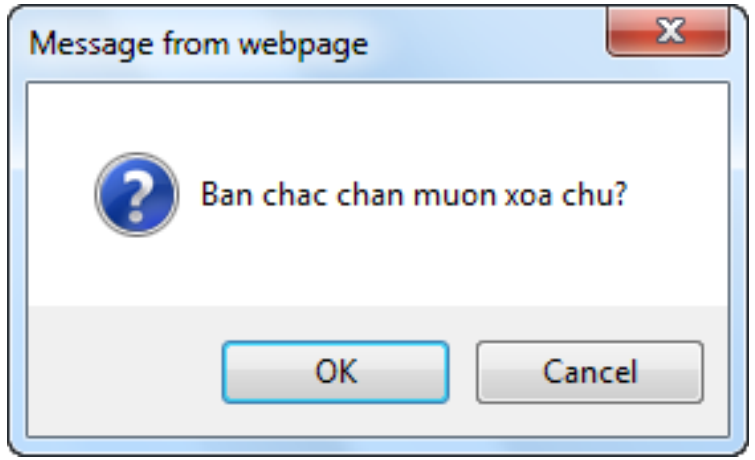
- Hàm trả về kết quả

```
function tester(para1, para2){
    return para1 + " " + para2;
}
const arg1 = "argument 1";
const arg2 = "argument 2";
let t = tester(arg1, arg2);
```

Hàm (function)

- `Confirm()`: là hộp thoại nhận hồi đáp từ phía người dùng
  - Lời gọi hàm: `confirm(<thông điệp>);`
  - Hàm trả về hồi đáp của người dung
    - Trả về True nếu người dùng nhấn vào OK
    - Trả về False nếu người dùng nhấn vào Cancel

```
var ok = confirm("Ban chac chan muon xoa chu?");
if(ok == true){
   doSometing();
}
else{
   doAnythingElse();
}
```



Hàm (function)

- Ví dụ về: `Confirm()`

```
function xacNhan(traloi) {
    var ketQua = "";
    if (traloi) {
        ketQua = "Tuyet voi. Chuc ban chien thang!";
    } else {
        ketQua = "Hen gap lai ban nhe!";
    }
    return ketQua;
}
var traloi = confirm("Ban se choi game chu?");
var thongbao = xacNhan(traloi);
alert (thongbao);
```

FPT Education
FPT POLYTECHNIC

Phạm vi biến

- Biến cục bộ (local variable)
  - Biến được khai báo trong hàm
  - Chỉ được tham chiếu đến trong phạm vi khai báo
- Biến toàn cục (global variable)
  - Biến được khai báo ngoài hàm
  - Có thể tham chiếu đến từ bất cứ đâu

Phạm vi biến

- Biến cục bộ (local variable)

```
function testAvailability() {
    let y = "I'll return";
    console.log("Available here:", y);
    return y;
}
let z = testAvailability();
console.log("Outside the function:", z);
console.log("Not available here:", y);
```

Phạm vi biến

- Biến cục bộ (local variable) với `let` và `var`

```
function doingStuff() {
    if (true) {
        var x = "local";
    }
    console.log(x);
}
doingStuff();
//result: local
```

```
function doingStuff() {
    if (true) {
        let x = "local";
    }
    console.log(x);
}
doingStuff();
//result: ReferenceError: x is not
defined
```

Phạm vi biến

- Biến toàn cục (global variable)

```
let globalVar = "Accessible everywhere!";
console.log("Outside function:", globalVar);
function creatingNewScope(x) {
    console.log("Access to global vars inside function." , globalVar);
}
creatingNewScope("some parameter");
console.log("Still available:", globalVar);
```

```
//Outside function: Accessible everywhere!
//Access to global vars inside function. Accessible everywhere!
//Still available: Accessible everywhere!
```

FPT Education
FPT POLYTECHNIC

Xử lý sự kiện (events)

- Events là những thứ/những điều xảy ra trên một trang web
- Ví dụ: nhấp chuột (click) vào cái gì đó, di chuyển chuột qua (mouse over) một phần tử (element),…
- Tất cả các element trên trang web đều có một tập các sự kiện tương ứng.
- ***Một phần tử chỉ có 1 trình xử lý sự kiện (event handler) làm thuộc tính. Nếu phần tử có xử lý sự kiện onclick thì không thể có onmouseover***

Chú ý: Cùng một element, các browser khác nhau hỗ trợ các tập sự kiện khác nhau

Xử lý sự kiện (events)

- Một số sự kiện
  - `onClick`: được kích hoạt khi nhấn chuột vào một element
  - `onload` và `onUnload`: được kích hoạt khi người dùng vào hoặc thoát khỏi trang web
- Click vào bất kỳ đâu trên trang web

```
<!DOCTYPE html>
<html>
<head>
  <title>Page Title</title>
</head>
  <body onclick="alert('Hi Event!')"></body>
</html>
```

Sự kiện

Xử lý sự kiện

FPT Education
FPT POLYTECHNIC

Xử lý sự kiện (events)

- Có thể thêm nhiều dòng lệnh

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body onclick="alert('Hi Event!');alert('Hello Event');"></body>
</html>
```

Sự kiện

Xử lý sự kiện

Trong trường hợp xử lý phức tạp cho sự kiện????

FPT Education
FPT POLYTECHNIC

Xử lý sự kiện (events)

- Sử dụng hàm để thực hiện các xử lý cho sự kiện

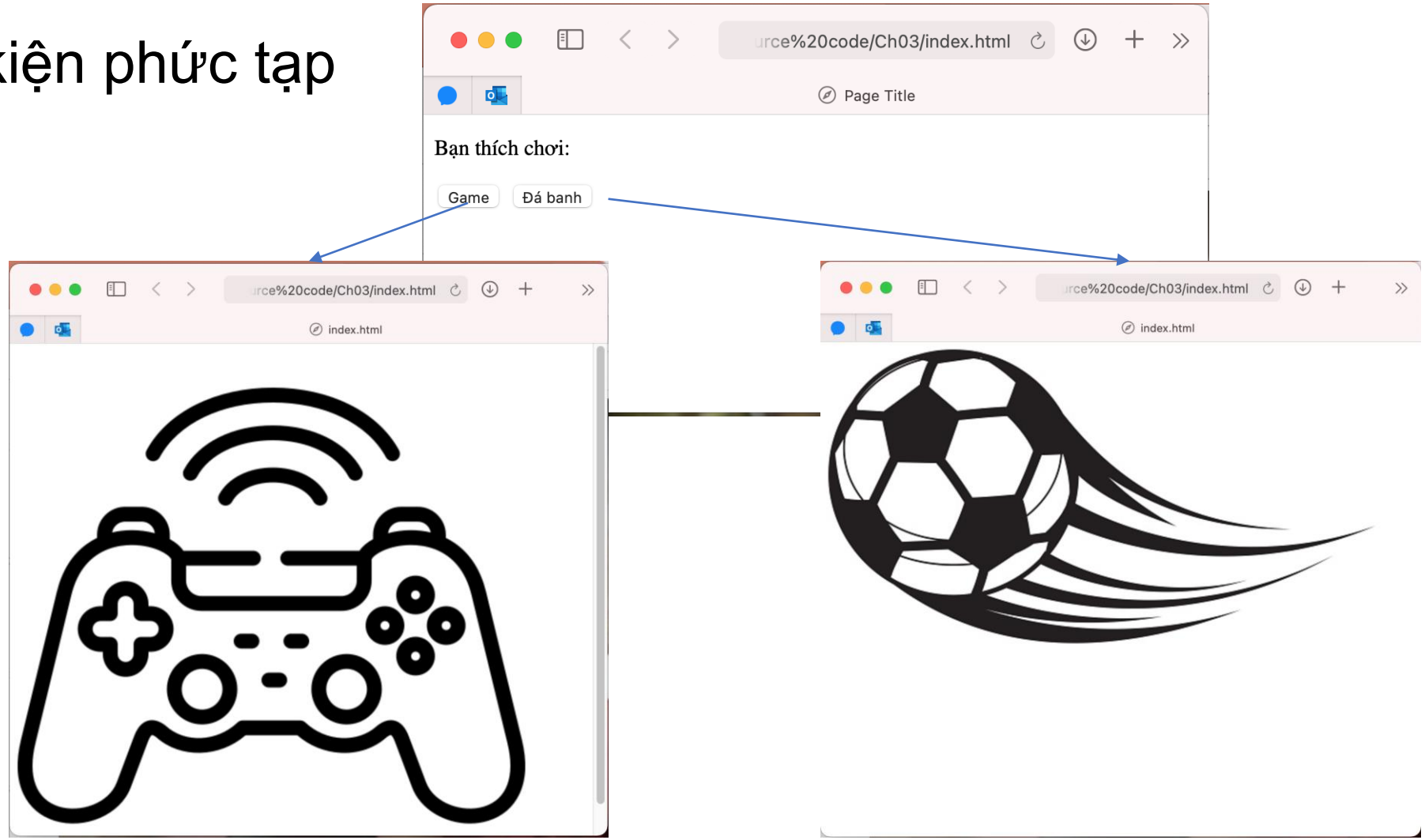
```
<!DOCTYPE html>
<html>
<head>
    <title>Page Title</title>
    <script src="chapter3.js"  type="text/javascript"></script>
    <script type="text/javascript">
        function sayHi(){
            alert("Hi Event");
            alert("Hello Event");
        }
    </script>
</head>
<body>

    <button onclick="sayHi()">Add a number</button>

</body>
</html>
```

Xử lý sự kiện (events)

- Sự kiện phức tạp



FPT Education
FPT POLYTECHNIC

Xử lý sự kiện (events)

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function hienThiAnh(dovat) {
	if (dovat == "game") {
		document.write("<img src = 'game.png'>");
	} else {
		document.write("<img src = 'football.jpg'>");
	}
}
</script>
</head>
<body>
    <p> Bạn thích chơi:</p>
    <input type="button" value="Game" onclick="hienThiAnh('game');"/>
    <input type="button" value="Đá banh" onclick="hienThiAnh('football');"/>
</body>
</html>
```

FPT Education
FPT POLYTECHNIC

TỔNG KẾT

- Lệnh logic: if, if else, if elseif
- Toán tử bậc 3 có điều kiện
- Lệnh switch
- Lặp: while, do while, for, for of, for in
- Hàm thực hiện một chức năng cụ thể. Hàm có thể trả về giá trị hoặc không trả về giá trị.
- Biến có phạm vi cục bộ và toàn cục

FPT Education
FPT POLYTECHNIC

TỔNG KẾT

- Hàm confirm là hàm được xây dựng sẵn dùng để lấy thông tin hồi đáp từ người dùng
- Javascript cung cấp sự kiện cho các element của trang web. Mỗi element có một tập các sự kiện khác nhau


Thank you!