FPT Education
FPT POLYTECHNIC

THỰC HỌC – THỰC NGHIỆP

LẬP TRÌNH JAVASCRIPT

BASIC JAVASCRIPT

FPT Education
FPT POLYTECHNIC

Hệ thống bài cũ

- Javascript là gì?
- Lịch sử Javascript
- Javascript có thể làm gì?
- Cài đặt môi trường
- Helloworld với Console trên browsers
- Javascript trên web
- Lệnh javascript
- Quy tắc cơ bản của javascript
- Built-in function

FPT Education
FPT POLYTECHNIC

MỤC TIÊU BÀI HỌC

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

FPT Education
FPT POLYTECHNIC

PHẦN I

Biến và kiểu dữ liệu

FPT Education
FPT POLYTECHNIC

Variables (Biến)

- **Variable (biến)** dùng để lưu trữ dữ liệu
- Cú pháp:

```
var/let/const <variable_name>;
```

- Cách đặt tên biến
  - Tên biến bao gồm chữ cái và số, nhưng không được bắt đầu bằng số
  - Tên biến không bao gồm dấu cách và dấu câu, ngoại trừ dấu gạch dưới (_)
- Có thể khai báo biến trên một dòng

```
var x,y,zeta;
```

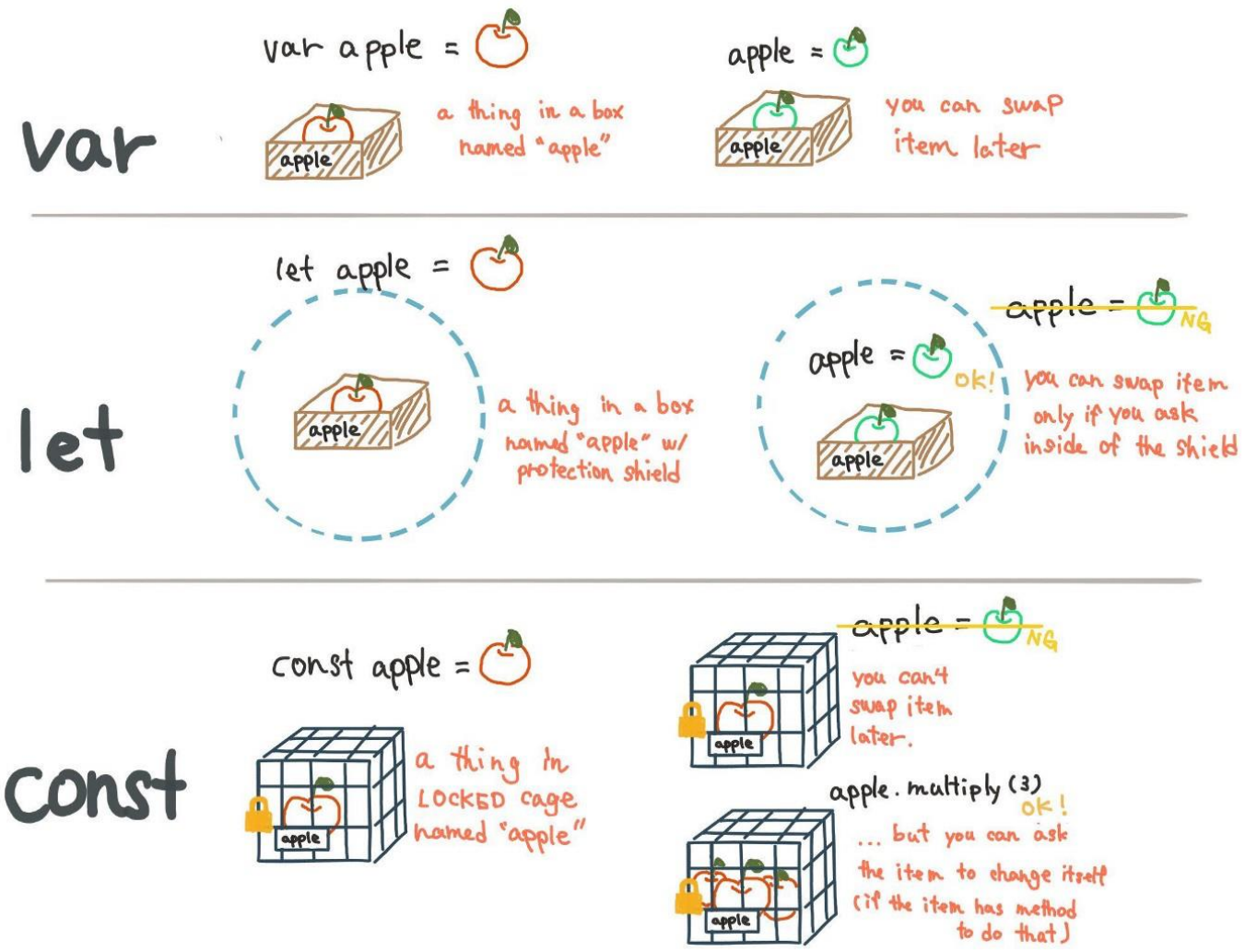
- Có thể vừa khai báo vừa khởi tạo giá trị cho biến

```
var x=1;
var x=1, y="hello";
```

FPT Education
FPT POLYTECHNIC

Variables (Biến)

- **`var`**: có phạm vi global scope
- **`let:`** có phạm vi block scope { }
- **`const:`** gán giá trị cho biến chỉ 1 lần, không được thay đổi giá trị



Source: @mpjme (https://twitter.com/mpjme)

FPT Education
FPT POLYTECHNIC

Kiểu dữ liệu

- Javascript hỗ trợ những kiểu dữ liệu - Primitive data sau:
  - String
  - Number
  - BigInt
  - Boolean
  - Symbol
  - Undefined
  - Null

Symbol (primitive data): kiểu dữ liệu mới từ ES6. Đề cập chi tiết sau.

FPT Education
FPT POLYTECHNIC

Kiểu dữ liệu - String

- Cách để khai báo string
  - Dấu nháy đôi " "
  - Dấu nháy đơn ' ‘
  - Backticks ` `
- Ví dụ

```
let funActivity = 'Let's learn JavaScript’;//Error

let language = "JavaScript";
let message = `Let's learn ${language}`;
console.log(message);
```

Kiểu dữ liệu - String

- Ký tự đặc biệt

Ký tự	Giải thích
\'	'
\"	"
\b	Dấu cách
\t	Dấu tab
\n	Xuống dòng

- Ví dụ

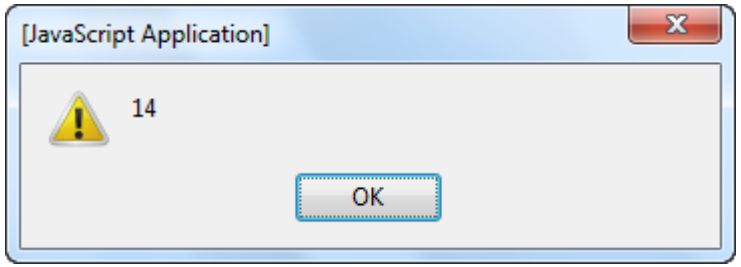
```
let str = "Hello, what's your name? Is it \"Mike\"?";
console.log(str);

let str3 = "New \nline.";
let str4 = "I'm containing a backslash: \\!";
console.log(str3);
console.log(str4);
```

Kiểu dữ liệu – String (thuộc tính & phương thức)

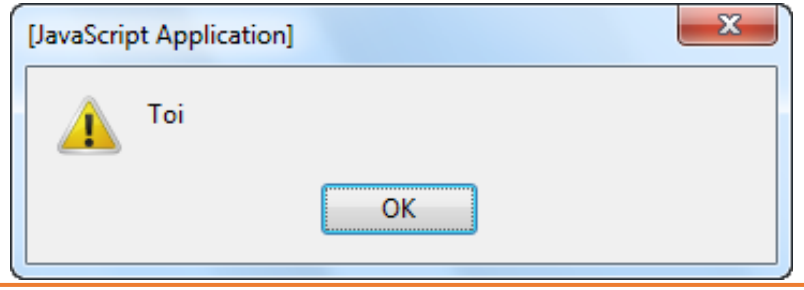
- Thuộc tính **length**

```
var x = "Toi la String."; alert(x.length);
```



- Phương thức **substring**

```
var x = "Toi la String."; alert(x.substring(0,3));
```



FPT Education
FPT POLYTECHNIC

Kiểu dữ liệu – String (thuộc tính & phương thức)

- Phương thức **`concat`**

```
let text1 = "Hello";
let text2 = "World";
let text3 = text1.concat(" ", text2);
```

- Phương thức **`toUpperCase`**

```
let text1 = "Hello World!";
let text2 = text1.toUpperCase();
```

- Phương thức **toLowerCase**

```
let text1 = "Hello World!"; // String
let text2 = text1.toLowerCase(); // text2 is text1 converted to
lower
```

FPT Education
FPT POLYTECHNIC

Kiểu dữ liệu – Number

- Javascript không chia ra kiểu Double, Integer…như các ngôn ngữ khác.
- Javascript gộp lại thành một kiểu duy nhất là Number
- Giới hạn của kiểu number là 2^{53}-1 và –(2^{53}-1)

```
let intNr = 1;
let decNr = 1.5;
let expNr = 1.4e15;
let octNr = 0o10; //decimal version would be 8
let hexNr = 0x3E8; //decimal version would be 1000
let binNr = 0b101; //decimal version would be 5
```

FPT Education
FPT POLYTECHNIC

Kiểu dữ liệu – Number (Phương thức)

- **`isNaN()`**
  - Xác định xem tham số truyền vào có phải là số hay không
  - Nếu là số, trả về False
  - Nếu không phải là số, trả về True

```
let x = 7;
console.log(isNaN(x));
```

- **`isInteger()`**

```
let x = 3;
let str = "integer";
console.log(Number.isInteger(x));
console.log(Number.isInteger(str));
```

FPT Education
FPT POLYTECHNIC

Kiểu dữ liệu – Number (Phương thức)

- **`toFixed()`:** chuyển number thành string, giữ lại số lượng số thập phân chỉ định
- **`toPrecision()`:** làm tròn số theo số lượng số thập phân chỉ định

```
var num = 5.56789;
var a = num.toFixed(); // 6
var b = num.toFixed(2); // 5.57
var c = num.toFixed(10); // 5.5678900000

var a1 = num.toPrecision(); // 5.56789
var b1 = num.toPrecision(2); // 5.6
var c1 = num.toPrecision(10); // 5.567890000
```

FPT Education
FPT POLYTECHNIC

Kiểu dữ liệu – BigInt

- BigInt
  - Được sử dụng trong trường hợp lớn hơn (hoặc nhỏ hơn) kiểu number
  - Sử dụng bằng cách thêm vào cuối ký tự n

```
let bigNr = 90071992547409920n;
```


Các cấu trúc điều khiển sẽ đề cập chi tiết ở phần sau

FPT Education
FPT POLYTECHNIC

Kiểu dữ liệu – Boolean

- Kiểu boolean có hai giá trị là **`true`** và **`false`**
- Các biểu thức Boolean thường được sử dụng trong các *cấu trúc điều khiển*

```
let bool1 = false;
let bool2 = true;


if (x > 18) {
   alert("Hi");
}
```

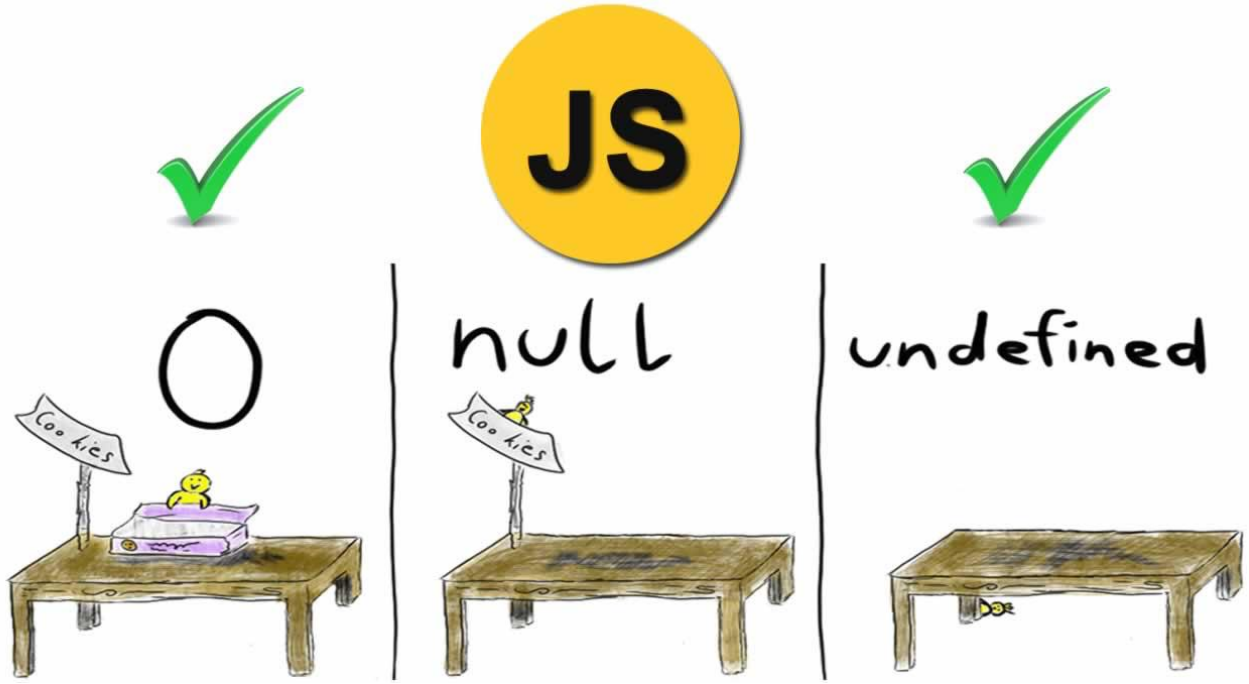

Các cấu trúc điều khiển sẽ đề cập chi tiết ở phần sau

FPT Education
FPT POLYTECHNIC

Kiểu dữ liệu – Undefined và null

- **`Null`** (giá trị rỗng) & **`undefined`**: không thuộc bất kỳ kiểu mô tả nào
- Là giá trị đặc biệt đại diện cho nothing (không có gì), empty(rỗng) hoặc value unknown (không các định)

```
let terribleThingToDo = undefined;
let lastName;
console.log("Same undefined:", lastName ===
terribleThingToDo);
let betterOption = null;
console.log("Same null:", lastName === betterOption);
```



Result:
Same undefined: true
Same null: false

FPT Education
FPT POLYTECHNIC

PHẦN II

Toán tử và biểu thức

FPT Education
FPT POLYTECHNIC

Javascript multiple values – Array (Mảng)

- **`Array`** (Mảng) : là kiểu dữ liệu dùng để lưu một tập các dữ liệu có kiểu giống nhau
- Tạo mảng

```
arr1 = new Array("hello", "world", "javascript", "array");
arr2 = ["hello", "world", "javascript", "array"];

console.log(arr1);
console.log(arr2);
```

- Truy xuất phần tử (element)

```
console.log(arr1[0]);
console.log(arr1[-1]);
```

FPT Education
FPT POLYTECHNIC

Javascript multiple values – Array (Mảng)

- Ghi đè phần tử (overwriting element)

```
arr1 = new Array("hello", "world", "javascript", "array");
arr1[0] = "Hi";
console.log(arr1[0]);


arr1[-1] = "overwriting";
console.log(arr1[3]);
console.log(arr1[-1]);
console.log(arr1);
```

FPT Education
FPT POLYTECHNIC

Javascript multiple values – Array (Mảng)

- Thuộc tính **length**

```
arr1 = new Array("hello", "world", "javascript", "array");
booleans = [true, false, false, true];
emptyArray = [];
console.log("Length of arr1:", arr1.length);
console.log("Length of booleans:", booleans.length);
console.log("Length of empty array:", emptyArray.length);

lastElement = arr1[arr1.length - 1];
```

FPT Education
FPT POLYTECHNIC

Javascript multiple values – Array (Mảng)

- Phương thức ***thêm và thay thế phần tử***
  - **Push():** thêm phần tử vào cuối mảng

```
let arrOfShapes = ["circle", "triangle", "rectangle", "pentagon"];
arrOfShapes.push("square");
console.log(arrOfShapes);
```

  - **splice():** Xoá y phần tử sau vị trí x, sau đó thêm phần tử vào mảng
  - Cú pháp: <arrayname>**.splice(x, y,** <e_1>,<e_2>,...,<e_x>**);**

```
let arrOfShapes = ["circle", "triangle", "rectangle", "pentagon"];
console.log(arrOfShapes);
arrOfShapes.splice(2, 0, "square", "trapezoid");
console.log(arrOfShapes);
```

FPT Education
FPT POLYTECHNIC

Javascript multiple values – Array (Mảng)

- Phương thức ***thêm và thay thế phần tử***
  - **`concat()`:**

```
let arr5 = [1, 2, 3];
let arr6 = [4, 5, 6];
let arr7 = arr5.concat(arr6);
console.log(arr7);
```

FPT Education
FPT POLYTECHNIC

Javascript multiple values – Array (Mảng)

- Phương thức ***xoá phần tử***
  - **Pop():** xoá phần tử cuối trong mảng

```
let arr5 = [1, 2, 3];
let arr6 = [4, 5, 6];
let arr7 = arr5.concat(arr6);
console.log(arr7);

let arr8 = arr7.concat(7, 8, 9);
console.log(arr8);
arr8.pop();
```

  - **splice():** Xoá y phần tử sau vị trí x

```
arr8 = [ 2, 3, 4, 5, 6, 7, 8 ];
arr8.splice(1, 3);
console.log(arr8);
```

FPT Education
FPT POLYTECHNIC

Javascript multiple values – Array (Mảng)

- Phương thức ***tìm phần tử***
  - **`index()`***:* trả về vị trí của phần tử x

```
arr8 = [ 2, 6, 7, 8 ];
let findIndex = arr8.indexOf(6);
let findIndex2 = arr8.indexOf(10);
console.log(findIndex, findIndex2);
```

  - **`lastIndexOf()`**: trả về vị trí cuối cùng của phần tử x

```
let animals = ["dog", "horse", "cat", "platypus", "dog"];
let lastDog = animals.lastIndexOf("dog");
```

Ép kiểu

- Ép kiểu ngầm định
  - Trình thông dịch tự động chuyển kiểu

```
var x = 100;
alert("Hello" + x);
```

Ép kiểu

- Ép kiểu tường minh
  - Ép kiểu số thành chuỗi

```
var x = String(100);
alert(typeof(x));
```

  - Ép kiểu chuỗi thành số

```
var x = "100";
var y = Number(x);
alert(typeof(y));
```

Ép kiểu

- String to Array: **`split()`**

```
let result = "Hello JavaScript";
let arr_result = result.split(" ");
console.log(arr_result);
```

- Array to String: **`join()`**

```
let letters = ["a", "b", "c"];
let x = letters.join();
console.log(x);
```

FPT Education
FPT POLYTECHNIC

Operators (Toán tử)

- Toán tử số học

Toán tử	Giải thích
+	Cộng
-	Trừ
*	Nhân
/	Chia
%	Chia lấy dư

- Toán tử logic

Toán tử	Giải thích
&	Và
|	Hoặc
^	XOR
!	NOT

Operators (Toán tử)

- Toán tử quan hệ và toán tử bằng

Toán tử	Giải thích
>	Lớn hơn
<	Bé hơn
>=	Lớn hơn hoặc bằng
<=	Bé hơn hoặc bằng
==	Bằng
!=	Khác

FPT Education
FPT POLYTECHNIC

Operators (Toán tử)

- Toán tử một ngôi

Toán tử	Giải thích
+	Chuyển toán hạng sang số dương
-	Chuyển toán hạng sang số âm
++	Tăng
--	Trừ

- Toán tử gom nhóm

Toán tử	Giải thích
(...)	(x+y)

Operators (Toán tử)

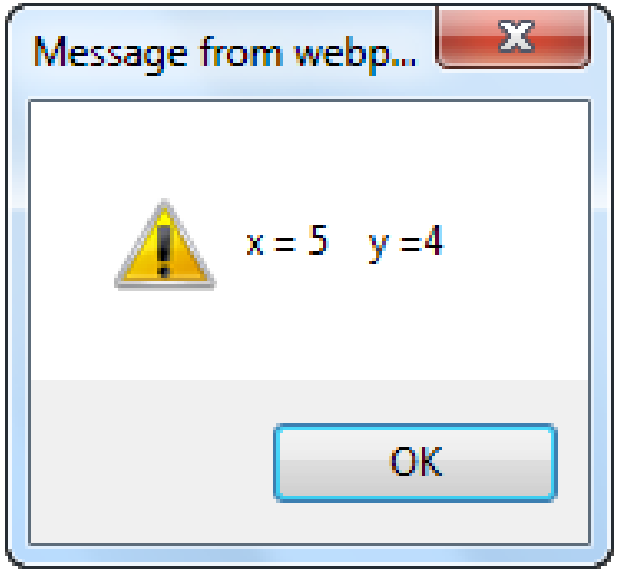
```
var x = 4;
x++;
alert (x);
```

Operators (Toán tử)

- Sự khác nhau giữa ++x và x++

```
var x = 4;
var y = ++x;
alert ("x = "+x+"y = "+y);
```


```
var x = 4;
var y = x++;
alert ("x = "+x+"y = "+y);
```



Math methods

- Math có nhiều methods cho phép thực hiện các phép tính và phép toán trên nhiều số.

- Tìm số lớn nhất và nhỏ nhất

```
let highest = Math.max(2, 56, 12, 1, 233, 4);
console.log(highest);

let highestOfWords = Math.max("hi", 3, "bye");
console.log(highestOfWords);
```

Math methods

- Căn bậc 2 và luỹ thừa

```
let result = Math.sqrt(64);
console.log(result);

let result2 = Math.pow(5, 3); //pow(base, exponent)
console.log(result2);
```

- Chuyển số thập phân thành số nguyên: `math.round()`, `math.ceil()`, `math.floor()`, `math.trunc()`

```
let x = 6.78;
let y = 5.34;
console.log("X:", x, "becomes", Math.round(x));
console.log("Y:", y, "becomes", Math.round(y));
```

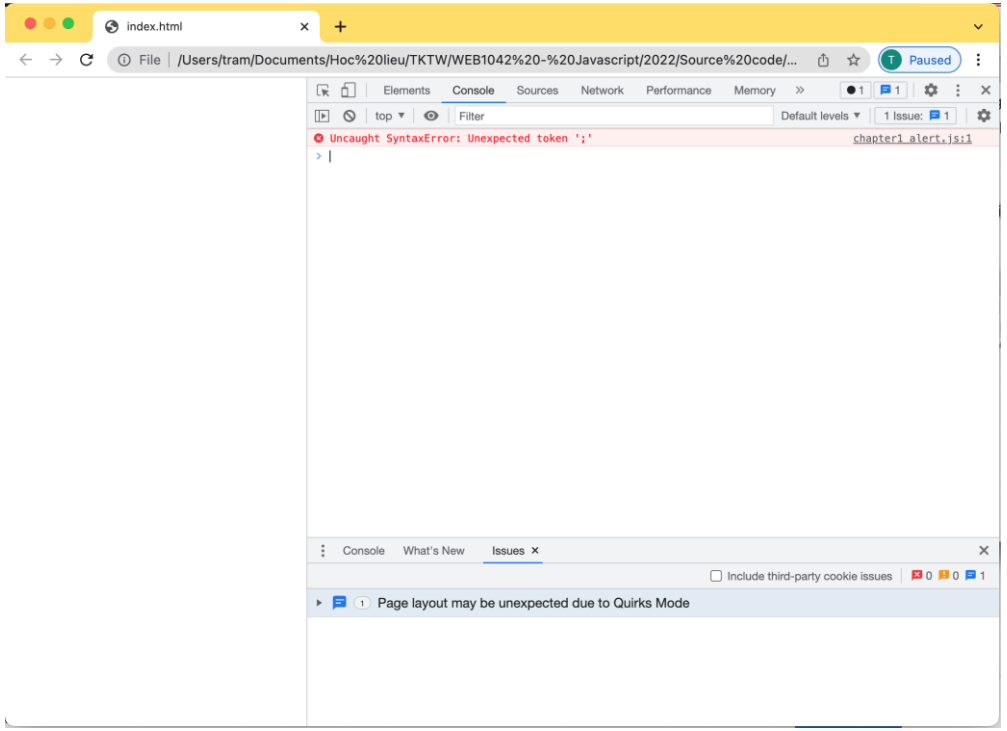
Math methods

```
let x = 6.78;
let y = 5.34;
console.log("X:", x, "becomes",
Math.round(x));
console.log("Y:", y, "becomes",
Math.round(y));
```

FPT Education
FPT POLYTECHNIC

Công cụ gỡ lỗi

- Devtools trên Chrome
  - Mở tập tin html (có sử dụng javascript trực tiếp hoặc liên kết)
  - Mở devtools bằng cách nhấn Command+Option+I (Mac) hoặc Control+Shift+I (Windows, Linux).



Ảnh chụp **Console** tab

FPT Education
FPT POLYTECHNIC

TỔNG KẾT

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


Thank
you!