

Bài 2 Biến, kiểu dữ liệu và toán tử

Module: BOOTCAMP PREPARATION



Kiểm tra bài trước

Hỏi và trao đổi về các khó khăn gặp phải trong bài "Nhập môn lập trình căn bản" Tóm tắt lại các phần đã học từ bài "Nhập môn lập trình căn bản"

Mục tiêu



- Trình bày được khái niệm biến
- Trình bày được cú pháp khai báo biến
- Trình bày được khái niệm kiểu dữ liệu
- Trình bày được các toán tử thông dụng
- Khai báo và sử dụng được biến
- Trình bày được các cách nhúng mã Javascript vào trong trang web
- Sử dụng được các kiểu dữ liệu
- Sử dụng được các toán tử cơ bản



Thảo luận

Biến và Hằng

Địn danh

Biến

Hằng

Quy ước đặt tên

Định vị toạ độ trên bản đồ

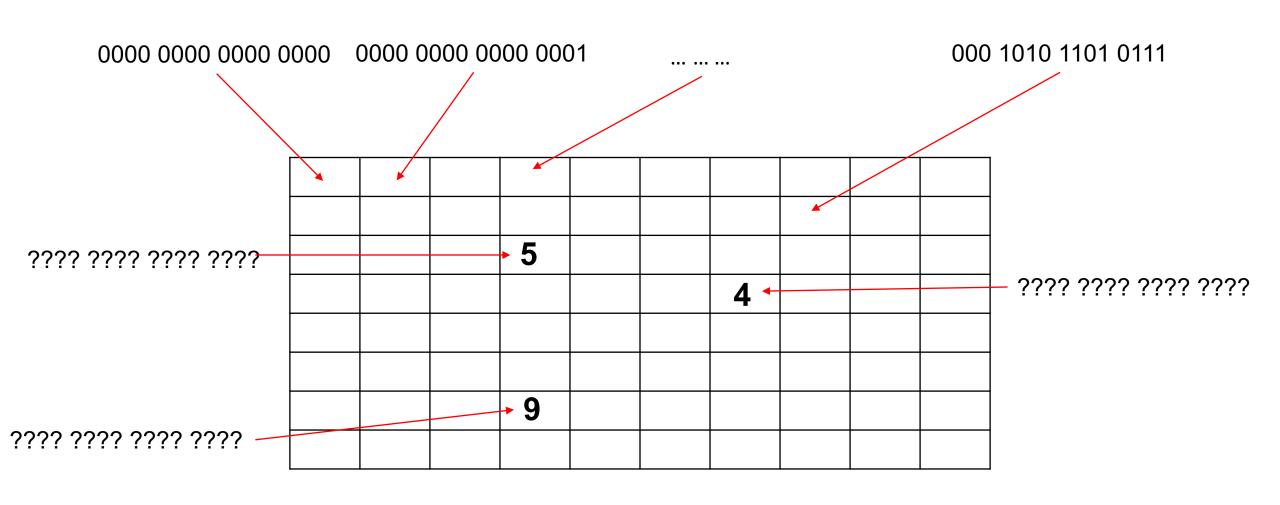


- Toạ độ 16°B và 112°Đ nằm ở đâu trên địa cầu?
- Có thể nhớ được toạ độ của các thủ đô của các nước không?
- Có dễ dàng hơn nếu muốn xác định vị trí của Quần đảo Hoàng sa?
- Có dễ dàng hơn nếu muốn xác định vị trí của Paris?



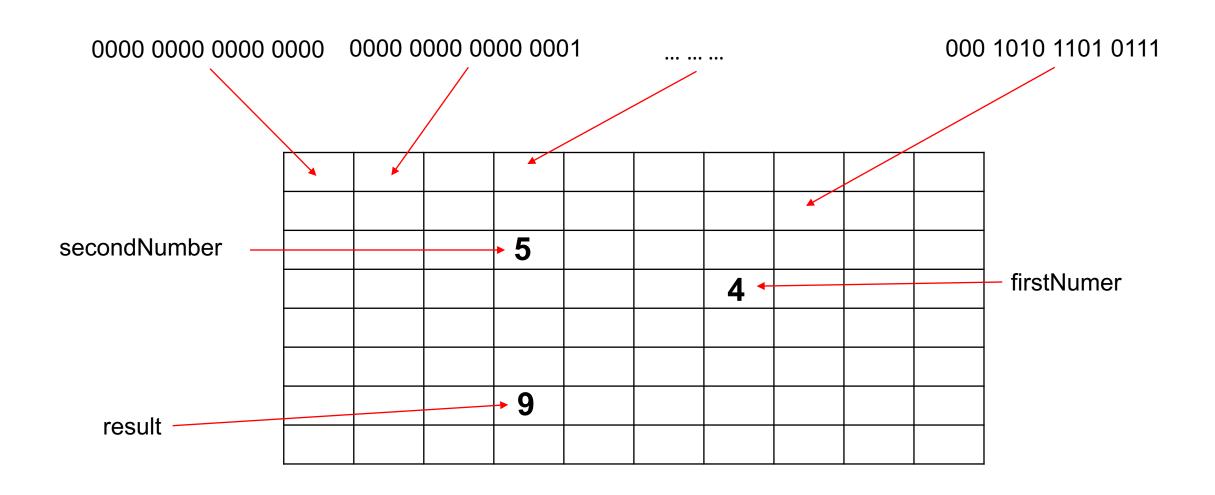
Lưu trữ dữ liệu trong bộ nhớ





Lưu trữ dữ liệu trong bộ nhớ





Biến (variable)



- Biến là một tên gọi được gắn cho một vùng nhớ chứa dữ liệu
- Dữ liệu được lưu trữ trong vùng nhớ của biến được gọi là giá trị (value)
- Có thể truy nhập, gán hay thay đổi giá trị của biến
- Khi gán một giá trị mới thì giá trị cũ sẽ bị ghi đè lên
- Cần phải khai báo biến trước khi sử dụng
- Chẳng hạn:

```
var x;
x = 6;
```

Ví dụ về biến



```
BEGIN

DISPLAY "Enter 2 numbers: "

INPUT A, B

C = A + B

DISPLAY C

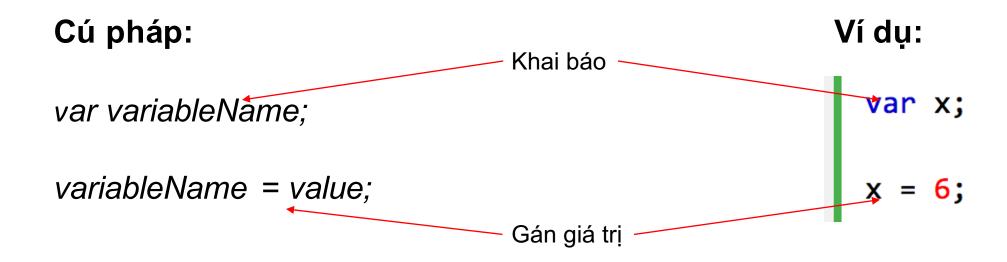
END
```

- A, B và C là các biến trong đoạn mã giả trên
- Tên biến giúp chúng ta truy cập vào bộ nhớ mà không cần dùng địa chỉ của chúng
- Hệ điều hành đảm nhiệm việc cấp bộ nhớ còn trống cho những biến này
- Để tham chiếu đến một giá trị cụ thể trong bộ nhớ, chúng ta chỉ cần dùng tên của biến

Khai báo và gán giá trị cho biến



- Từ khoá var được dùng để **khai báo** biến
- x là tên biến
- Dấu bằng (=) được dùng để **gán giá trị** cho biến



Đặt tên cho biến



- Tên biến phải bắt đầu bằng một ký tự alphabet (a-zA-z_)
- Theo sau ký tự đầu có thể là các ký tự chữ, số ...
- Nên tránh đặt tên biến trùng tên các từ khoá
- Tên biến nên mô tả được ý nghĩa của nó
- Tránh dùng các ký tự gây lầm lẫn
- Tên biến có phân biệt chữ hoa và chữ thường
- Nên áp dụng các quy ước đặt tên biến chuẩn khi lập trình

Đặt tên cho biến: Ví dụ



- Ví dụ đặt tên biến đúng
 - arena
 - s_count
 - marks40
 - class_one
- Ví dụ đặt tên biến sai
 - 1sttest
 - oh!god
 - start... end

Giá trị của biến



- Ví dụ
 - 5 số / giá trị số nguyên (integer)
 - 5.3 số / giá trị số thập phân (decimal)
 - "Black" **Giá trị chuỗi** (string)
 - 'C' **Giá trị ký tự** (character)
 - true và false là các **giá trị logic** (boolean)

Hằng (constant)



- Hằng là một tên gọi đại diện cho một giá trị cố định
- Giá trị của hằng không thể thay đổi
- Giá trị của hằng cần phải được gán tại thời điểm khai báo
- Ví dụ, sử dụng hằng PI thay cho giá trị 3.14159:

```
var area = radius * radius * 3.14159;
```

Được thay bằng:

```
const PI = 3.14159;
var area = radius * radius * PI;
```

Khai báo hằng



Cú pháp khai báo hằng:

const CONSTANTNAME = value;

Trong đó:

- const là từ khoá bắt buộc để khai báo hằng
- CONSTANTNAME là tên của hằng
- value là giá trị của hằng



Demo

Biến, kiểu dữ liệu Các kiểu dữ liệu nguyên thuỷ Chuỗi Giá trị mặc định

Kiểu dữ liệu (Data Type)



- Kiểu dữ liệu là một cách phân loại dữ liệu cho trình biên dịch hoặc thông dịch hiểu các lập trình viên muốn sử dụng dữ liệu.
- Kiểu dữ liệu mô tả loại dữ liệu sẽ được lưu trong biến
- Các kiểu dữ liệu khác nhau được lưu trữ trong biến là:
 - Số (numbers)
 - Số nguyên: 10 hay 83839
 - Số thực: 15.33 hay 23.6677
 - Số dương: 3, 4
 - Số âm: -6, -7
 - Chuỗi: "Hello"
 - Ký tự: 'A'
 - Logic: true, false

Kiểu dữ liệu (Data Type)



- Một kiểu dữ liệu cung cấp một bộ các giá trị mà từ đó một biểu thức (như biến, hàm ...) có thể lấy giá trị của nó
- Trong JavaScript khi khai báo biến và gán cho biến một giá trị đồng nghĩa xác định kiểu dữ liệu cho biến đó.
- Ví dụ:
 - var x = 10; // x mang giá trị 10, kiểu dữ liệu của x là số nguyên
 - var gender = true; // gender mang giá trị true, kiểu dữ liệu của gender là kiểu boolean

Chuỗi



- Chuỗi bao gồm các ký tự liên tiếp nhau
- Có thể khai báo chuỗi sử dụng dấu nháy đơn hoặc nháy kép
- Ví dụ:

```
var answer = "It's alright";
var answer = "He is called 'Johnny'";
var answer = 'He is called "Johnny"';
```

Số



- Có thể sử dụng số nguyên hoặc số thập phân
- Ví dụ:

```
var x1 = 34.00;
var x2 = 34;
```

Boolean



- Kiểu dữ liệu boolean chỉ có hai giá trị là true và false
- Tất cả mọi thứ có giá trị đều là true
- Tất cả mọi thứ không có giá trị đều là false

```
100
                                                                                  // returns false
            3.14
                                                                   var x = 0;
                                                                                  // returns false
                                                                   Boolean(x);
            -15
                                                    False
                                                                   var x = "";
True
                                                                   Boolean (x);
                                                                                    // returns false
            "Hello"
                                                                   var x;
            "false"
                                                                   Boolean(x);
                                                                                    // returns false
```



Thảo luận

Toán tử gán Toán tử số học Toán tử so sánh Toán tử logic

Toán tử (Operator)

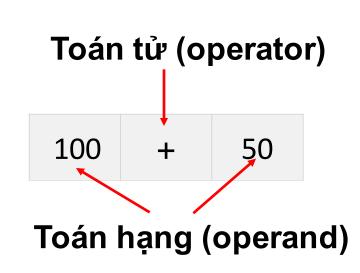


- Toán tử là các ký hiệu được sử dụng để thực hiện các thao tác trong các biểu thức và sinh ra kết quả cuối
- Có nhiều loại toán tử khác nhau:
 - Toán tử toán học
 - Toán tử gán
 - Toán tử cộng chuỗi
 - Toán tử so sánh
 - Toán tử logic
 - Toán tử typeof

Toán tử toán học (arithmetic)



- Toán tử toán học được sử dụng trong các biểu thức toán học
- Toán tử toán học được sử dụng trên các giá trị số (hoặc là các biến kiểu số)
- Toán tử toán học thông thường có 2 toán hạng



Toán tử	Mô tả
+	Cộng
-	Trừ
*	Nhân
1	Chia
%	Chia lấy phần dư (Modulus)
++	Tăng 1 giá trị
	Giảm 1 giá trị

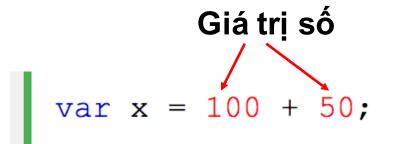
Toán tử toán học: Ví dụ



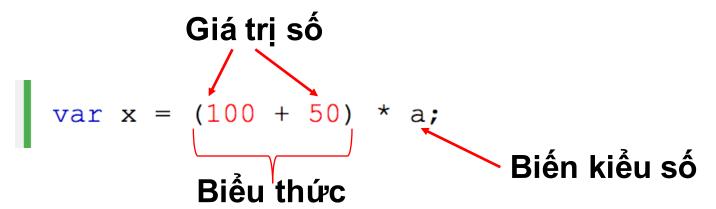
• Sử dụng với các giá trị:

• Sử dụng với các biến:

• Sử dụng với biểu thức:



Biến kiểu số
$$var x = a + b;$$



Toán tử cộng: Ví dụ



```
var x = 5;
var y = 2;
var z = x + y;
```

Toán tử trừ: Ví dụ



```
var x = 5;
var y = 2;
var z = x - y;
```

Toán tử nhân: Ví dụ



```
var x = 5;
var y = 2;
var z = x * y;
10
```

Toán tử chia: Ví dụ



```
var x = 5;
var y = 2;
var z = x / y;

25
```

Toán tử chia lấy số dư: Ví dụ



```
var x = 5;
var y = 2;
var z = x % y;
```

Toán tử tăng: Ví dụ



```
var x = 5;
x++;
var z = x;
```

Toán tử giảm: Ví dụ



```
var x = 5;
x--;
var z = x;
```

Toán tử gán (assignment)



- Toán tử gán được sử dụng để gán giá trị cho một biến
- Toán tử gán có thể sử dụng với tất cả các kiểu dữ liệu
- Ví dụ:

var x = 10;

Operator	Example	Same As
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y

Toán tử Cộng bằng: Ví dụ



```
var x = 10;
x += 5; //Tương đương với x = x + 5
```

Toán tử Trừ bằng: Ví dụ



```
var x = 10;
x -= 5; //Tương đương với x = x - 5
```

Toán tử Nhân bằng: Ví dụ



```
var x = 10;
x *= 5; //Tương đương với x = x * 5
```

Toán tử Chia bằng: Ví dụ



```
var x = 10;
x /= 5; //Tương đương với x = x / 5
```

Toán tử Phần trăm bằng: Ví dụ

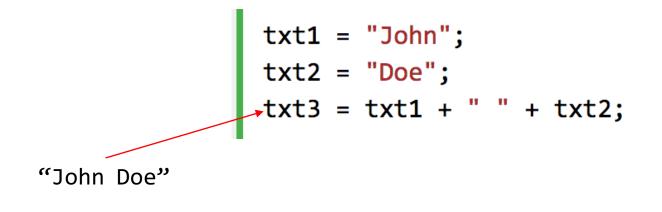


```
var x = 10;
x %= 5; //Tương đương với x = x % 5
0
```

Toán tử cộng chuỗi (string concatenate)



- Toán tử cộng chuỗi được sử dụng để nối hai chuỗi
- Có thể nối chuỗi với số



```
10 x = 5 + 5;

"55" y = "5" + 5;

z = "Hello" + 5;
```

Toán tử so sánh (comparision)



 Toán tử so sánh được dùng để đánh giá mức độ tương quan giữa các giá trị

Operator	Description
==	equal to (bằng)
===	equal value and equal type (bằng giá trị đồng thời cùng kiểu dữ liệu)
!=	not equal (khác)
!==	not equal value or not equal type (không bằng giá trị hoặc không cùng kiểu dữ liệu)
>	greater than (lớn hơn)
<	less than (nhỏ hơn)
>=	greater than or equal to (lớn hơn hoặc bằng)
<=	less than or equal to (nhỏ hơn hoặc bằng)

Toán tử so sánh bằng (==): Ví dụ



```
var x = 5;
var y = x == 5;
var/z = x == "5";
true true
```

Toán tử so sánh bằng (===): Ví dụ



```
var x = 5;
var y = x === 5;
var z = x === "5";
true false
```

Toán tử so sánh khác: Ví dụ



```
var x = 5;
var y = x != 8;
var z = x != 5;
true false
```

Toán tử so sánh khác: Ví dụ (2)



```
var x = 5;
var y = x !== 5;
var z = x !== "5";
false
```

Toán tử logic (logical)



- Toán tử logic được dùng trong các biểu thức logic (true/false)
- && là toán tử "và"
- || là toán tử "hoặc"
- •! Là toán tử "phủ định"

Operator	Description
&&	logical and
П	logical or
!	logical not

Toán tử &&



Giá trị biến a	Giá trị biến b	Kết quả (a && b)
true	true	true
true	false	false
false	true	false
false	false	false

Toán tử ||



Giá trị biến a	Giá trị biến b	Kết quả (a b)
true	true	true
true	false	true
false	true	true
false	false	false

Toán tử!



Giá trị biến a	Kết quả !a
true	false
false	true

Toán tử typeof



 Toán tử typeof được dùng để lấy về kiểu dữ liệu của một biến hoặc giá trị

Độ ưu tiên của các toán tử



- Trong một biểu thức có nhiều phép toán thì chúng sẽ lần lượt được đánh giá dựa vào đô ưu tiên
- Có thể sử dụng dấu ngoặc "()" để thay đổi độ ưu tiên của các toán tử
- Các toán tử có cùng độ ưu tiên thì sẽ thực hiện từ trái sang phải

Operators	Precedence
postfix	expr++ expr
unary	++exprexpr +expr -expr ~!
multiplicative	* / %
additive	+ -
shift	<<>>>>>
relational	<><=>= instanceof
equality	== !=
bitwise AND	&
bitwise exclusive OR	^
bitwise inclusive OR	
logical AND	&&
logical OR	
ternary	?:
assignment	= += -= *= /= %= &= ^= = <<= >>>=

Độ ưu tiên của các toán tử: Ví dụ



```
var x = 5;
var y = 10;
         (++x * y) < 5 * 10 && 6 > 3;
          (6 \quad * y) < 5 \quad * 10 \&\& 6 > 3;
            60 ) < 50 && 6 > 3;
                   false && true;
                               false
```



Demo

Toán tử gán Toán tử số học Toán tử so sánh Toán tử logic



Demo

Các cách nhúng JavaScript

Nhúng Javascript vào trang web



- Có một số cách khác nhau để đưa mã Javascript vào trong trang web:
 - Viết mã Javascript bên trong thẻ <script></script>
 - Sử dụng file .js và nhúng vào trang web
 - Sử dụng file .js từ bên ngoài

The <script>



• Có thể đặt trong thẻ <head> hoặc thẻ <body>

```
<!DOCTYPE html>
<html>
<body>
<h1>A Web Page</h1>
A Paragraph
<button type="button" onclick="myFunction()">Try it</button>
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
</script>
</body>
</html>
```

File .js



 Nhúng file .js vào tài liệu HTML sử dụng thuộc tính src của thẻ <script>

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

<script src="myScript.js"></script>

</body>
</html>
```

Sử dụng file .js từ bên ngoài



• Tái sử dụng các thư viện có sẵn

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>

Câu hỏi mở?



- Khi nào dùng cách đưa mã JavaScript vào trong trang web dạng:
 - Viết mã Javascript bên trong thẻ <script></script>
 - Sử dụng file .js và nhúng vào trang web
 - Sử dụng file .js từ bên ngoài

Chú thích (comment)



- Chú thích được dùng để giải thích ý nghĩa của một đoạn mã nào đó
- Chú thích không được thực thi (execute)
- Cách viết chú thích trên một dòng sử dụng / /
- Cách viết chú thích trên nhiều dòng sử dụng /* . . . */

 Lưu ý: Thêm chú thích vào trong mã nguồn để giải thích cho mã nguồn dễ hiểu hơn là một việc tốt. Tuy nhiên, sẽ tốt hơn nếu mã nguồn của chúng ta không cần có chú thích mà vẫn dễ hiểu.

Tóm tắt bài học



- Biến
- Toán tử
- Kiểu dữ liệu



Hướng dẫn

Hướng dẫn làm bài thực hành và bài tập Chuẩn bị bài tiếp theo: *Cấu trúc điều kiện*