

Bài: Bài tập về kiểu dữ liệu Number trong JavaScript

Xem bài học trên website để ủng hộ Kteam: [Bài tập về kiểu dữ liệu Number trong JavaScript](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

Trong bài này, Kteam và bạn sẽ thực hiện một số bài tập củng cố kiến thức về [Kiểu dữ liệu số trong JavaScript](#)

Để bài tập đạt hiệu quả tốt nhất bạn nên thực hiện theo các bước sau:

1. Đọc đề các câu hỏi và tự đưa ra đáp án/ lời giải của mình.
2. Tham khảo đáp án tại bài [Bài tập kiểu dữ liệu Chuỗi](#) và rà soát đáp án/ lời giải của bản thân
3. Xem video giải thích đáp án của bài này và note lại các ý chính để củng cố lại kiến thức.

Câu hỏi về kiểu dữ liệu số trong Javascript

Câu 1: Số khi đặt vào trong Javascript thì sẽ có kiểu dữ liệu là gì?

Câu 2: Infinity và -Infinity trong Javascript lần lượt bằng với các giá trị nào?

Câu 3: Giá trị nào dưới đây thuộc kiểu dữ liệu number ?

- a. NaN
- b. Undefined
- c. Null
- d. Float

Câu 4: Viết chương trình làm tròn số n, sau đó tính bình phương của số n, rồi in ra màn hình

Trong đó n là một số cho trước, và Kteam không yêu cầu bạn thực hiện việc nhập vào n, mà việc của các bạn là thực hiện yêu cầu như trên.

Câu 5: Nêu ra một vài hằng số và phương thức với Math trong Javascript.

Lấy một vài ví dụ.

Câu 6: Đặc điểm cơ bản nhất của bigint là gì ?

Câu 7: Ta có thể sử dụng **bigint** như là **number** thông thường hay không ? Nếu có, thì hãy cho ví dụ.

Câu 8: Chỉ ra một số **bitwise operator** và tác dụng của chúng.

Đáp án bài tập về biến trong JavaScrip

Bạn có thể xem chi tiết các câu hỏi ở [Bài tập về biến trong JavaScript](#)

Đáp án câu 1

Đáp án được hiển thị ra như sau:

JavaScript:

```
5 6
5 6
```

- **Giải thích:** Việc khởi tạo lại một biến bằng từ khóa **var** đơn giản chỉ là thay đổi giá trị của nó. Vì **var** giúp khởi tạo biến toàn cục, nên giá trị **t = 6** và **u = 5** sẽ được hiển thị bất kể nó được khởi tạo ở đâu.

Đáp án câu 2

	Biến cục bộ	Biến toàn cục	Hằng số
Phạm vi	Trong khối lệnh chứa nó	Toàn bộ chương trình đang được thực thi	
Độ linh động	Có thể thay đổi giá trị		Giá trị luôn phải được giữ nguyên trong suốt quá trình thực thi chương trình
Cách thức khởi tạo	Từ khóa let	Từ khóa var	Từ khóa const

Đáp án câu 3

Kết quả như bên dưới:

JavaScript:

```
1 2
kteam kter
```

Giải thích

- Ở 2 dòng đầu tiên, ta khởi tạo biến u và v có giá trị lần lượt là 'kteam' và 'kter'.
- Trong khối lệnh con (bọc trong cặp {}), ta lại khởi tạo 2 biến cục bộ u và v có giá trị là 1 và 2. Trong phạm vi khối lệnh này, u có giá trị là 1 và v có giá trị là 2. Do đó, cặp giá trị "1 2" được in ra đầu tiên.
- Sau đó, khi kết thúc khối lệnh con và trở về với khối lệnh chính, thì u và v lại "trở lại" với giá trị ban đầu ('kteam' và 'kter'). Vì vậy, giá trị in ra là 'kteam kter'.

Đáp án câu 4

Lựa chọn d đúng

Giải thích

- Ở đáp án a, vì 1 bé hơn 2 nên phép so sánh '1 lớn hơn hoặc bằng 2' trả về **false**.
- Ở đáp án b, cũng tương tự. 1 bé hơn 2.
- Ở đáp án c, vì 1 khác 2 nên trả về **false**.
- Ở đáp án d: 1 khác 2 à **true**.

Đáp án câu 5

Khác với các toán tử **!=** và **==** (khi mà **==** và **!=** chỉ so sánh sự giống và khác của 2 giá trị được so sánh), **===** và **!==** còn so sánh cả kiểu dữ liệu của chúng. Cụ thể, đối với toán tử **===**:

- Nếu 2 toán hạng khác kiểu dữ liệu, trả về **false**.

- Nếu 2 toán hạng là **object**, chỉ trả về true nếu cả 2 cùng trỏ đến một vùng bộ nhớ (gần tương tự như **Object.is()** – kiến thức mà Kteam sẽ đề cập đến trong tương lai)
- Nếu cả 2 toán hạng đều là **null** hoặc **undefined**, trả về true.
- Nếu một trong 2 toán hạng là **NaN**, trả về false.
- So sánh giá trị của 2 toán hạng. Nếu 2 toán hạng có cùng giá trị, trả về true và ngược lại.

Sự khác biệt đáng chú ý nhất, chính là việc **==** sẽ chuyển các giá trị về cùng kiểu trước khi so sánh.

Tất cả những trường hợp mà **===** trả về false, thì **!==** trả về true và ngược lại.

Trên đây đa phần là những kiến thức có vẻ “lạ”. Để biết rõ hơn về những kiến thức mà Kteam đang đề cập, mời các bạn tiếp tục theo dõi [khóa học JavaScript cơ bản](#) trên website howKteam.

Đáp án câu 6

Các toán tử so sánh trong Js được tóm tắt như trong bảng sau:

Kí hiệu trong JavaScript	Tác dụng
<	Trả về true nếu toán hạng thứ nhất bé hơn toán hạng thứ 2 và ngược lại
>	Trả về true nếu toán hạng thứ nhất lớn hơn toán hạng thứ 2 và ngược lại
>=	Trả về true nếu toán hạng thứ nhất lớn hơn hoặc bằng toán hạng thứ 2
<=	Trả về true nếu toán hạng thứ nhất bé hơn hoặc bằng toán hạng thứ 2
==	Trả về true nếu 2 toán hạng có giá trị bằng nhau và ngược lại.
!=	Trả về true nếu 2 toán hạng có giá trị khác nhau và ngược lại.
===	Trả về true nếu 2 toán hạng có giá trị bằng nhau và có cùng kiểu dữ liệu .
!==	Trả về true nếu 2 toán hạng khác giá trị hoặc khác kiểu dữ liệu .