JavaScript Pow(x, n)

Challenge

Implement `pow(x, n)`, which calculates `x` raised to the power `n` (ex. `xⁿ`).

1st Example

```
Input: x = 2.00000, n = 10
Output: 1024.00000
```

2nd Example

```
Input: x = 2.10000, n = 3
Output: 9.26100
```

3rd Example

```
Input: x = 2.00000, n = -2
Output: 0.25000
Explanation: 2⁻² = 1/2² = 1/4 = 0.25
```

Constraints

- `-100.0 < x < 100.0`
- `-2³¹ <= n <= 2³¹-1`

- $-10^4 <= x^n <= 10^4$
- `n` is an integer.

Solution

```
const myPow = (x, n) => {
    if (n == 0) {
        return 1;
    }

    if (n % 2 == 0) {
        return myPow(x * x, n / 2);
    }

    if (n == 1) {
        return x;
    }

    if (n < 0) {
        return 1 / myPow(x, Math.abs(n));
    }

    return x * myPow(x, n - 1);
};
```

Explanation

I've defined a function called `myPow` that calculates the power of a given number. It takes two parameters, `x` and `n`, where `x` is the base number and `n` is the exponent.

The function starts by checking if `n` is equal to `0`. If it is, the function returns `1`, as any number raised to the power of `0` is always `1`.

If `n` is not `0`, the function checks if `n` is an even number by checking if `n` modulo `2` is equal to `0`. If it is, the function recursively calls itself with the base number `x` squared and the exponent `n` divided by `2`. This is possible because $x^n = (x^2)^{(n/2)}$ when `n` is even.

If `n` is not `0` and not even, the function checks if `n` is equal to `1`. If it is, the function returns the base number `x` itself, as any number raised to the power of `1` is itself.

If `n` is not `0`, not even, and not `1`, the function checks if `n` is less than `0`. If it is, the function recursively calls itself with the base number `x` and the absolute value of `n`. This is possible because $x^{(-n)} = 1/(x^n)$.

If none of the above conditions are met, the function returns the base number `x` multiplied by the function recursively called with the base number `x` and the exponent `n` decremented by `1`. This is possible because $x^n = x * x^{(n-1)}$.

In summary, the `myPow` function calculates the power of a given number by using recursion and applying different mathematical properties depending on the value of the exponent.

Author: Trevor Morin