**DATA TYPES & CONDITIONALS**

Before we start digging deep, read through this overview of the most common data types in JavaScript: <https://javascript.info/types>

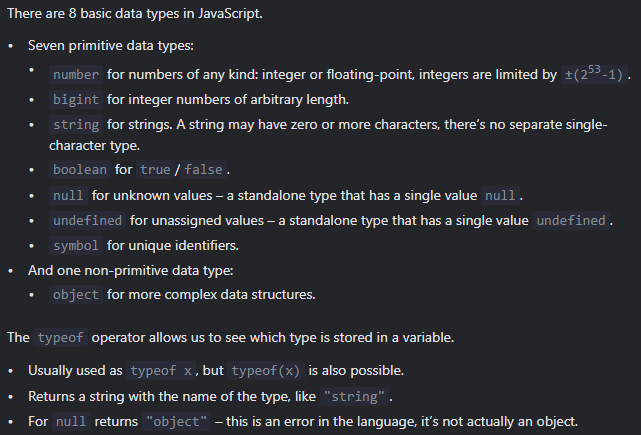
1. **Data Types**

* Null : represents “nothing”, “empty” or “value unknown”



* Undefined : The meaning of undefined is “value is not assigned”
* Object : to store collections of data and more complex entities
* Symbol : to create unique identifiers for objects

Summary:

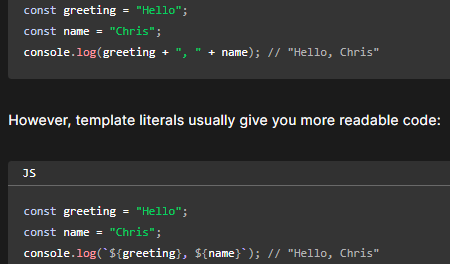


1. **String**

In JS, you can choose single quotes ('), double quotes ("), or backticks (`) to wrap your strings in. Strings declared using backticks are a special kind of string called a template literal. Inside a template literal, you can wrap JavaScript variables or expressions inside

‘ ${ } ‘, and the result will be included in the string:

* Concatenation using “+”

You can use ${} only with template literals, not normal strings. You can concatenate normal strings using the + operator:

* Template literals
* Including expressions in strings.
* Allow to write Multiline strings(without \n).
* Including quotes in strings. We also can use single quotes for wrapping strings that including (double) quotes. We also can use backslash to make sure the quote recognized as text. Ex: ‘I\'ve got no right to take my place…'

1. String Method

Source:

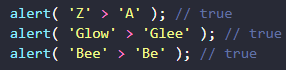
* <https://www.w3schools.com/js/js_string_methods.asp>
* <https://www.w3schools.com/jsref/jsref_obj_string.asp>

Docs: (You are not expected to memorize these but the documentation will be a very useful reference to revisit, so bookmark it!)

* <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String>

1. **Conditionals**
2. Comparison

* String comparison

To see whether a string is greater than another, JS uses unicode order.

- the longer string is greater.

- “a” is greater than “A”, Because the lowercase character has a greater index in the unicode.



- unicode comparison, first char "2" is greater than the first char "1".

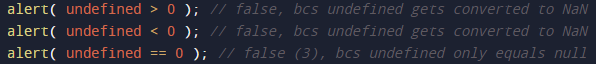
* Different types comparison

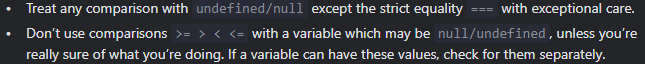
When comparing values of different types, JavaScript converts the values to numbers.

* Strict equality
* Null & Undefined comparison

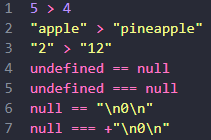
Strange result

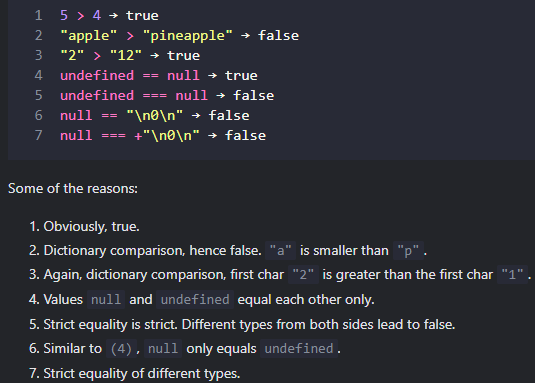
The reason is that an equality check ‘==’ and comparisons ‘> < >= <=’ work differently. Comparisons convert null to a number, treating it as 0. That’s why (3) null >= 0 is true and (1) null > 0 is false.

* An incomparable undefined

Avoid Problem:

Exercises: (true or false)





1. Logical Operator (CEK TASK paling bawah <https://javascript.info/logical-operators>)

Penjelasan detail mudah dipahami: <https://chatgpt.com/share/7952c32f-801f-4674-8743-be6a686bbc8c>

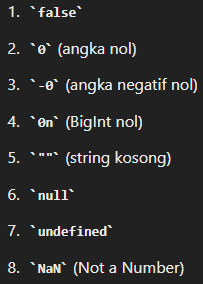
1. OR ( || )

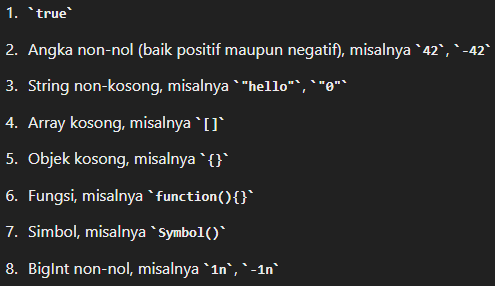
Operator OR mengevaluasi setiap operand dari kiri ke kanan dan mengembalikan nilai pertama yang "truthy". Jika semua operand "falsy", maka ia mengembalikan nilai yg terakhir.

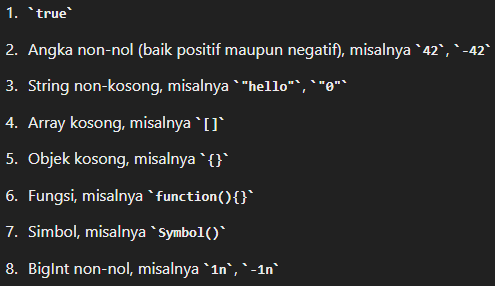
1. AND ( && )

Operator AND mengevaluasi setiap operand dari kiri ke kanan dan mengembalikan nilai pertama yang "falsy". Jika semua operand "truthy", maka ia mengembalikan nilai terakhir.

1. Nilai-nilai “falsy”

Nilai-nilai berikut akan dievaluasi sebagai false dalam konteks Boolean:

1. Nilai-nilai “Truthy”



1. Syntax

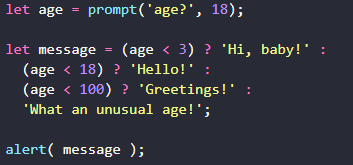
* Ternary operator

Syntax: condition ? <expression if true> : <expression if false>

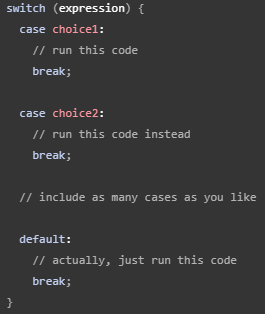
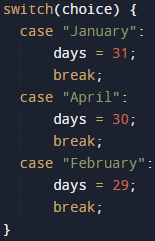
Example:

Penjelasan: if (harga < 10) then beli() else batal();

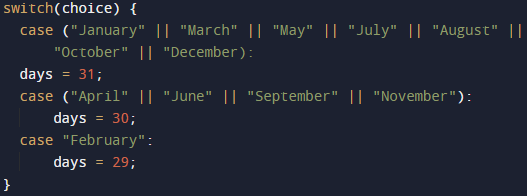
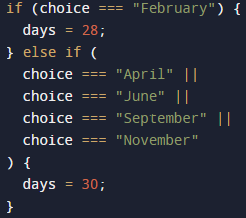
Other example:

Other example:

* Switch

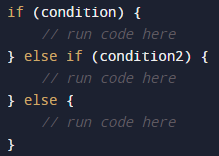
Syntax:

Ex:

Note: Tidak bisa seperti ini

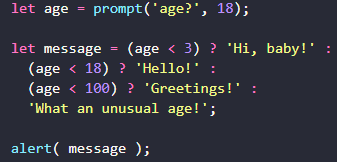
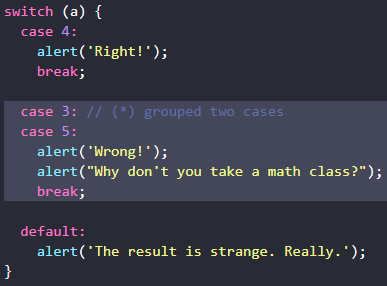
Solusinya:

* If-else

Syntax:

* Multiple ternary operator

The conditional similar with ‘else if’

* Multiple switch ‘case’

Note: Type Matters. Let’s emphasize that the equality check is always strict. The case values must be the same type. Example:

